Redžúr 2021

PROCEEDINGS

15th International Workshop on Multimedia Information and Communication Technologies

Edited by:
Ivan Minárik
Gregor Rozinaj
Radoslav Vargic

STU
FEI

ZDRUŽENIE
POUŽÍVATEĽOV
TELEKOMUNIKÁCIÍ
SLOVENSKA

ISBN 978-80-227-5113-1

SPEKTRUM
STU

PROCEEDINGS

Redžúr 2021

15th International Workshop on Multimedia Information
and Communication Technologies

4th June 2021, Bratislava, Slovakia

EDITED BY:
Ivan Minárik
Gregor Rozinaj
Radoslav Vargic

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering and Information Technology
Institute of Multimedia Information and Communication Technologies
Ilkovičova 3
812 19 Bratislava
Slovak Republic

Published by:

Vydavateľstvo SPEKTRUM STU
of the Slovak University of Technology in Bratislava

in cooperation with

Združenie používateľov telekomunikácií Slovenska

ISBN 978-80-227-5113-1

STU
FEI

ZDRUŽENIE
POUŽÍVATEĽOV
TELEKOMUNIKÁCIÍ
SLOVENSKA

General Chair

Gregor Rozinaj

Slovak University of Technology, Bratislava, Slovak Republic

Program Committee

Aura Conci

Fluminense Federal University, Brasil

Mislav Grgic

University of Zagreb, Croatia

Gerhard Gruhler

Heilbronn University, Germany

Jaromír Hrad

Czech Technical University, Czech Republic

Juraj Kačur

Slovak University of Technology, Slovakia

Jarmila Pavlovičová

Slovak University of Technology, Slovakia

Pavol Podhradský

Slovak University of Technology, Slovakia

Markus Rupp

Vienna University of Technology, Austria

Yevgeniya Sulema

National Technical University of Ukraine, Kyiv Polytechnic Institute, Ukraine

Radoslav Vargic

Slovak University of Technology, Slovakia

Tomáš Zeman

Czech Technical University, Czech Republic

Branka Zovko-Cihlar

University of Zagreb, Croatia

Review Committee

Abreu, Raphael

Aguilera, Cristhian A.

Araújo, José Denes

Aung, Zeyar

Bergamasco, Leila

Bergo, Felipe

Bezerra, Eduardo

Bozek, Jelena

Bravenec, Tomas

Bujok, Petr

Burget, Radim

Casaca, Wallace

Ciarelli, Patrick Marques

Copetti, Alessandro

Costa, Tales Fernandes

Čepko, Jozef

Davídková Antošová, Marcela

Devamane, Shridhar

Galić, Irena

Gonçalves, Vagner Mendonça

Grgic, Sonja

Habijan, Marija

Haddad, Diego Barreto

Henriques, Felipe

Hocenski, Zeljko

Hrad, Jaromír

Jakóbczak, Dariusz Jacek

Juhár, Jozef

Kačur, Juraj

Karwowski, Damian

Kominkova Oplatkova, Zuzana

Körting, Thales Sehn

Kos, Marko

Kultan, Matej

Laguna, Juana Martinez

Latkoski, Pero

Lima, Alan

Londák, Juraj

Lopes, Bruno

Lopes, Guilherme Wachs

Lourenço, Vítor

Malajner, Marko

Mandic, Lidija

Marana, Aparecido Nilceu

Marchevský, Stanislav

Marinova, Galia

Markovska, Marija

Matos, Caio

Medvecký, Martin

Minárik, Ivan

Mocanu, Stefan

Mustra, Mario

Nyarko, Emmanuel Karlo

Paiva, Anselmo

Papa, Joao Paulo

Podhradský, Pavol

Polak, Ladislav

Prinosil, Jiri

Rakús, Martin

Rodriguez, Denis Delisle

Rozinaj, Gregor

Rybárová, Renata

Silva, Aristófanes

Silvestre, Santiago

Slanina, Martin

Sousa De Almeida, Joao Dallyson

Sousa, Azael Melo E

Stasinski, Ryszard

Tcheou, Michel

Toledo, Yanexis Pupo

Trúchly, Peter

Turan, Jan

Vargic, Radoslav

Veras, Rodrigo

Vitas, Dijana

Vlaj, Damjan

Vukovic, Josip

Wajda, Krzysztof

Zamuda, Ales

Zeman, Tomas

Organizing Committee

Juraj Londák

Slovak Republic

Ivan Minárik

Slovak Republic

Šimon Tibenský

Slovak Republic

Marek Vančo

Slovak Republic

Table of Contents

This page is intentionally left blank.

Preface

Redžúr is an International Workshop on Multimedia Information and Communication Technologies with fruitful history organized at Slovak University of Technology since 2007. The main idea of this workshop was to give to young researchers a first opportunity to publish their first scientific paper. However, within the years Redžúr has become a well-established event for young and more experienced professionals in the field, as well. Throughout its history, Redžúr has been organized on various places, namely Bratislava, Vienna, Smolenice and Dubrovnik. It has evolved from a narrow-focused workshop on speech and signal processing into an international meetup of mainly young researchers in wide spread of fields covering signal and multimedia creation, processing and transmission across various media and underlying infrastructure.

Focus of the workshop is on young researchers, preferably university students, where they can present usually their first scientific results. The 15th International Workshop on Multimedia Information and Communication Technologies, Redžúr 2021 has been held in Bratislava, the capital of the Slovak Republic, on 4th June 2021 as a collocated event of International Conference on Systems, Signals and Image Processing, IWSSIP 2021, and hosted by the Slovak University of Technology, Faculty of Electrical Engineering and Information Technology in Bratislava. Due to the specific pandemic situation, Redžúr and IWSSIP have been organised as on-line events with the great support of underline.io.

Dear participants, thank you for your interest in Redžúr 2021.

Bratislava, 4th June 2021

Gregor Rozinaj

Chairman of Redžúr

This page is intentionally left blank.

Improving Deep Learning Convergence using Atlas Registration as a Preprocessing Step to Predict Final Stroke Lesion from Multimodal MRI Images

Pierrick Ullius[1], Noëlie Debs[1][0000−0002−9958−8806], David Rousseau[2][0000−0002−7935−1609], and Carole Frindel[1][0000−0003−4570−0994]

[1] Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1206, F-69XXX, LYON, France
[2] LARIS, UMR INRAE IRHS, Université d'Angers, Angers, 49000, France
carole.frindel@creatis.insa-lyon.fr

Abstract. In this paper, we propose a method to improve deep learning convergence in the context of final stroke lesion prediction. This method is based on a deep convolutional neural networks (CNN) using acute multimodal magnetic resonance images as input. Firstly, we registered input images on an anatomical atlas as a preprocessing step. This preprocessing step allowed to normalize input images beyond the pixel intensity ranges but also on the brain tissues morphology. Then, a CNN was trained with these atlas-registered images to predict the final stroke lesion. These experiments, performed on a large-scale dataset (92 patients), proved that registration of raw data on anatomical atlas accelerate the convergence of the CNN training phase by a factor of 38%. Therefore, atlas registration could save substantial time for fine-tuning the CNN.

Keywords: Deep learning · Convolutional neural network · Preprocessing · Registration · Anatomical atlas · Convergence · Magnetic resonance imaging · Stroke · Prediction.

1 Introduction

Stroke is the leading cause of long-term disability and mortality world-wise. Acute neuroimaging is crucial to choose the best therapeutic option and is currently focused on the prediction of the lesion. Currently, both computed-tomography (CT) and magnetic resonance imaging (MRI) entail threshold-based methods to delineate the still salvageable brain. Specifically in MRI, criteria for the infarct core and ischemic penumbra are apparent diffusion coefficient (ADC, extracted from diffusion-weighted imaging) and time to maximum of the residue function (T_{max}, extracted from perfusion-weighted imaging) [10]. Still, developing automated methods to predict the extent of the final stroke lesion from MRI scans remains an open challenge [13]. Machine learning approaches have been successfully proposed in past years [9, 8] and deep learning more recently [14, 12, 17, 4].

Preprocessing (registration, normalization, denoising, ...) is a very common approach in image processing. Preprocessing is expected to improve the performance of the final processing by reducing the non meaningful variability. This is specially useful for hand crafted image processing models based on few parameters [1, 2]. Deep learning offers models with much higher expressivity than hand crafted models which can perform even with few preprocessing on raw data. Common preprocessing steps consist in rescaling image intensities within a given value range to prevent early saturation of non-linear activation function and in resizing images to a unified dimension that matches the dimension of the training samples. Preprocessing can also cause some artefact. This is specially the case for image registration which requires some interpolation step [7]. For deep learning approaches it is therefore not straightforward to know what will be the consequence of a pre-registration step.

Generally, an automated scheme involves two main steps that produce features along with classification using a machine learning mechanism. Therefore, the success of this scheme depends on identifying the most significant features to solve the classification problem and choosing which learning algorithm to use. Recently, deep learning methods have attracted considerable attention because of their remarkable performance enhancement [5]. Using a deep architecture to mimic the natural multi-layer neural network, these methods can automatically and adaptively learn a hierarchical representation of patterns from low to high-level features for a given task. Compared to the conventional machine learning methods, a point of difference in deep learning is that feature extraction is automatic and goes through a large number of parameters (number of neurons and layers encoded in the underlying network). A limitation to the practical use of deep learning is the tuning of hyper-parameters which is crucial to accommodate the complexity of the machine learning task and the underlying data. This optimization is usually done by a grid search technique where each parameter represents a dimension of the grid and is tested within a predefined range. Furthermore, in this process, CNN training requires a large number of epochs to achieve satisfactory performance and results in considerable computational cost. Therefore, we propose in this work to register the input brain MRI data on non-linear ICBM 152 Atlas [6] upstream to the CNN training in order to reduce the complexity of the data with regard to the targeted task and therefore accelerate the convergence time. The CNN training convergence and its quality are quantified via quantitative measurements on the loss (stop epoch in the early stopping, distance between the training and validation loss curves and oscillations in the training loss curve) and is done by looking jointly at task performance (Dice score on the prediction of the final stoke lesion).

2 Methods

2.1 Input data

Patients were included from the HIBISCUS-STROKE cohort. HIBISCUS-STROKE is an ongoing monocentric observational cohort enrolling patients with a large intracranial artery occlusion treated by thrombectomy, following a baseline diffusion-perfusion MRI. All patients underwent on admission diffusion-weighted-imaging (DWI) and dynamic susceptibility-contrast perfusion imaging (DSC-PWI). A follow-up FLAIR was performed 6 days after admission which describes the final lesion and was used as ground truth for the prediction task.

MRI acquisition parameters were as follows : DWI (repetition time 6000 ms, field of view 24 cm, matrix 192x192, slice thickness 5 mm), FLAIR (repetition time, 8690 ms; echo time, 109 ms; inversion time 2500 ms; flip angle, 150°; field of view, 21 cm; matrix, 224 × 256; 24 sections; section thickness, 5 mm) and DSC-PWI (echo time 40 ms, repetition time 1500 ms, field of view 24 cm, matrix 128 × 128, 18 slices, slice thickness 5 mm; gadolinium contrast at 0.1 mmol/kg injected with a power injector). All patients gave their informed consent and the imaging protocol was approved by the regional ethics committee.

2.2 Image post-processing

Parametric maps were extracted from the DSC-PWI by circular singular value decomposition [16] of the tissue concentration curves (Olea Sphere, Olea Medical, La Ciotat, France): cerebral blood flow (CBF), cerebral blood volume (CBV), mean transit time (MTT), time to maximum (T_{max}) and time to peak (TTP).

Lesions on the baseline DWI and final FLAIR were segmented by an expert with a semi-automated method (3D Slicer, https://www.slicer.org/). Specifically, a region-of-interest-controlled thresholding was used with manual corrections when required. Images were co-registered within subjects to the baseline DWI MRI using non-linear registration with Ants [3]. The skull from all patients was removed using FSL [15]. Finally, images were normalized between 0 and 1 to ensure inter-patient standardization.

2.3 Atlas registration

The anatomical atlas chosen for our study is the non-linear Atlas ICBM 152 [6]. This atlas exists in different anatomical MRI contrasts T1, T2 and PD. The T2-weighted template was chosen to develop the registration procedure because it is similar to the contrast of our $b = 0$ diffusion data (i.e. without diffusion coefficient).

As the inter-patient anatomical differences can be significant in terms of the general brain shape and volume (see Fig. 1, first column contours in blue) but also in more specific regions, such as the ventricles (see Fig. 1, first column contours in magenta), we decided to use non-linear registration using [3]. In order to find the right set of registration parameters for each patient, we have optimized 4 parameters which have been identified as having the most significant impact on the quality of the final registration (preliminary tests not detailed here) :

- *shrinking factor*: registration is carried out gradually at different resolutions, called levels. The idea is to start low resolution then go to the next step, with a higher resolution version, and so on. The shrinking factor is

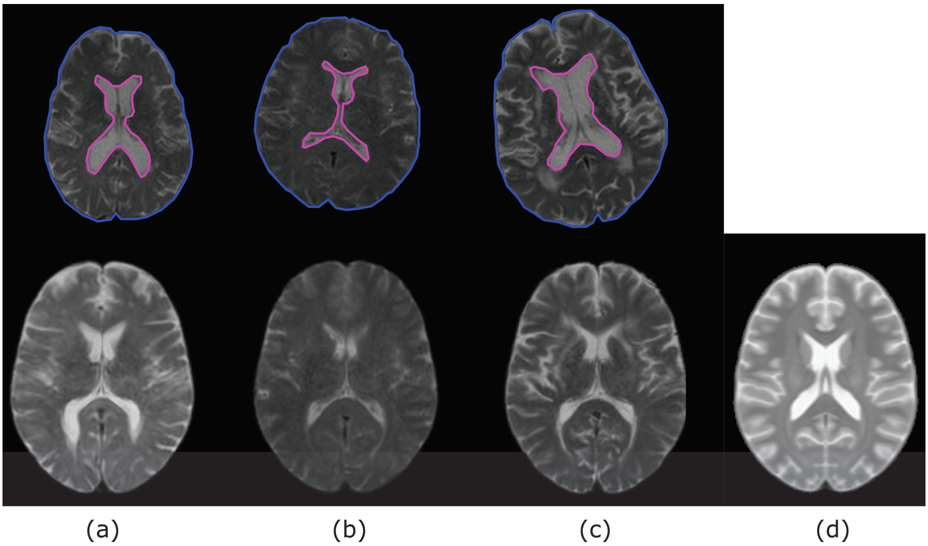


Fig. 1: Examples of raw (first line) and registered (second line) MRI images used in this study. Each column corresponds to a different patient and the last to the altas used in this study.

the down-sampling factor applied to each hierarchical level (6 levels in our case); 6 different values were tested (30x20x15x10x5x4, 20x15x10x5x4x2, 15x10x8x6x4x2, 10x8x6x5x3x1, 10x6x4x2x1x1).

- *update field variance*: determines how much to smooth the gradient field between updates and increasing this value increases smoothness in the velocity field ; 5 different values have been tested (1, 2, 4, 6, 10).
- *mutual information percentage*: mutual information measures how similar two images look. It uses the histograms of the two images. To speed up this comparison, it is proposed to evaluate the mutual information only on a percentage of voxels sampled regularly among the image ; 7 different values have been tested (0.10, 0.20, 0.40, 0.50, 0.75, 0.85, 0.95).
- *convergence threshold*: the threshold value tells the algorithm to stop if the improvement in mutual information has not changed more than the specified value in the last chosen number of iterations (10 iterations in our case) ; 4 different values have been tested (1×10^{-4}, 1×10^{-6}, 1×10^{-8}, 1×10^{-10}).

To quantify the quality of the registration, we extracted the white and gray brain tissue masks from the registered data and the atlas using FSL [15] and then calculated their area of overlap via the the Dice similarity score. The same transformations were then applied to full DWI and DSC-PWI data, and lesion masks. The registration for the 92 patients data took approximately 82 minutes on a work station with an NVIDIA GeForce GTX 1080 GPU with 128 GB memory.

2.4 CNN Architecture and training

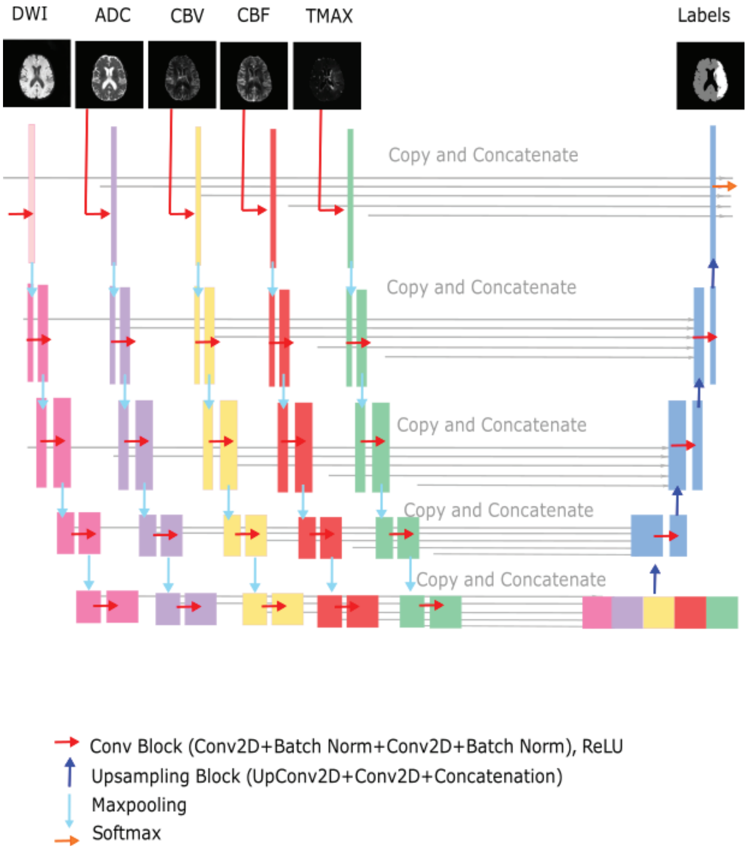


Fig. 2: Overview of the late fusion deep learning architecture. Top left: The network takes five MRI images (2D slices from DWI, ADC, CBV, CBF, T_{max} images) as input. Below: Each input image is processed independently on 5 separate branches. Pink, purple, yellow, red and green feature maps result from 2D-convolutions and maxpooling. The output of the 5 branches are then concatenated, and upsampled through 2D-deconvolution layers. The network produces an output map with 3 classes (lesion, healthy tissue and background). Top Right : The predicted lesion has to be compared to the true lesion from the final FLAIR.

We used a U-Net architecture that has already shown its potential for infarct prediction tasks [14]. More precisely, we used the architecture of one of our previous study [4], in order to study the impact of registration as a preprocessing step. Five inputs were used: raw DWI and ADC for diffusion MRI, as well as T_{max}, CBF and CBV for perfusion MRI. Late fusion was chosen for its potential to better integrate each MRI input [4]. The five inputs were fed into our late fusion network by 5 distinct convolution branches (see Figure 2). The associated encoding layers are detailed in Table 1. Each input consisted of whole 2D images. No patches were applied in order to benefit from the global spatial context for lesion prediction. The network produced probability maps respectively associated with three classes: background, healthy tissue and stroke lesion. The probability map associated with the lesion was thresholded at 0.5 to define the final infarct.

Table 1: Encoding layers of the proposed late fusion U-net. The encoder was composed of 5 convolution blocks (Conv Block), maxpooling operations (2D MaxPooling) and dropout. The Conv Block was made of: 2D convolution (3*3)+ batch normalization + 2D convolution (3*3)+ batch normalization.

Layer (type)	Output shape
Conv Block 1	192*192*8
2D MaxPooling	96*96*8
Conv Block 2	96*96*16
2D MaxPooling	48*48*16
Conv Block 3	48*48*32
2D MaxPooling	24*24*32
Conv Block 4	24*24*64
Dropout + 2D Maxpooling	12*12*64
Conv Block 5 + Dropout	12*12*128
Concatenation	12*12*640

To reduce the class imbalance problem between background, healthy tissue and stroke lesion, we used a multi-class Dice function for the loss function [11], where the lesion class was assigned a weight 8 times higher than those of healthy tissue and background classes. For updating weights in the network, we used the Adam optimizer with a learning rate of 1×10^{-4}, decay of 5×10^{-4}) and a batch size of 12. To prevent overfitting, we applied dropout (set to 0.5), used a L2 regularizer at each convolution layer ($reg = 2 \times 10^{-4}$) and the number of epochs (set to 500) was regulated by early stopping (training was stopped once the best validation loss did not increase more than 0.005 on 100 epochs). The evaluation of each model was performed using a 5-fold cross-validation. We used Keras 2.1.3 library with Python 3.6.3 interface. The training phase took approximately 1 hour on a work station with an NVIDIA GeForce GTX 1080 GPU with 128 GB memory.

2.5 Evaluation metrics

CNN convergence was assessed by 1/ the number of epochs (EPOCHS) before reaching stable performance on the validation dataset, 2/ the amplitude of the oscillations (OSC) on the training and validation loss curves, 3/ the distance between the training and the validation loss curves (DIST) and 4/ the Dice similarity coefficient (DSC) between the prediction and the ground truth. These metrics were calculated for the 5 folds of our cross-validation and given as a mean and a standard deviation.

3 Results

Table 2 shows the convergence and the performance of our CNN trained on the raw data (without registration). These results are associated with a set of hyper-parameters identified as optimal (these values are recalled in Table 3). The number of features corresponds to the number of units in the first CNN layer.

Table 2: CNN convergence and performance on raw data (without registration)

EPOCHS	265 ± 115
OSC	0.07
DIST	0.4
DSC	0.52 ± 0.05

This setting allows to achieve a stable convergence (few OSC oscillations and a reduced distance between the train and validation curves DIST) in 265 epochs on average, with a mean DSC of 0.52 on the validation set. As a reference, it is interesting to note that the best models so far in the stroke prediction ISLES 2017 challenge [14] had an average DSC of 0.38 (±0.22). Our performance is increased with regard to this previous work, although an absolute comparison is not strictly possible because the dataset used for ISLES 2017 is different from ours (in terms of resolution and balance between patients).

Table 3: Optimal CNN hyper-parameters on raw data (without registration)

CNN parameters	# features	dropout	regularizer
Raw data	8	0.5	0.0002

Table 4 shows the convergence and the performance of our CNN trained on registered data. Results depicted in it has to be compared with results from Table 2. In Table 4, each hyper-parameter was tuned independently while maintaining the other hyper-parameters fixed to their default values (those found for the raw data).

It is interesting to note from Table 4 that even if input data differs from Table 3 (especially in terms of variability), the optimal set of parameters remains the same. On the other hand, the convergence time (EPOCHS) is considerably reduced (165 epochs in average for the optimal hyper-parameters with registered data compared to 265 epochs without registration), which represents a gain in time of 38%, especially for fine-tuning phase. Considering all the possible hyperparameter combinations presented in Table 4 for the CNN fine-tuning step, there are a total of 64 hyperparameter combinations to be tested. As specified in Section D, fine-tuning on a given hyper-parameter set takes approximately 1 hour. A gain of 38% in training time corresponds to about twenty hours in comparison to only 82 minutes of registration necessary for the 92 patients.

Beyond the time saving, it should also be noted from Table 4 that the training is more stable (DIST and OSC reduced). Oscillating performance is known to be caused by weights that diverge and may drive the model to a suboptimal solution. In our case, the fact that the model is learned from data with reduced variability smoothes the landscape linked to the optimization of the weights and helps the model to converge towards a global optimum.

As for the detection performance, it is almost intact (very slightly reduced) which can be explained by the fact that the atlas registration slightly blurs the data and erases certain details, as illustrated in Fig. 1. However, this will have minimal impact in the context of our application, since what is aimed at is more a faithful assessment of the final lesion volume than a precise contouring.

4 Discussion

The objectives of data preprocessing in classical machine learning includes size reduction of the input space and smoother relationships. This preprocessing step can provide a better modeling and avoid numerical problems [1, 2].

In this paper, we proposed to register CNN input data on an anatomical atlas as a preprocessing step to normalize the input data beyond the pixel intensity ranges but also on the tissue morphology. These results were illustrated in the context of a database of 92 patients as part of the prediction of the final stroke injury.

According to our results, the pre-processing stage did not reduce the number of hyperparameters required for the network. This result is very application-dependent and difficult to generalize. In our case, we have optimized the network so that the loss focuses on the detection of the ischemic stroke lesion. Setting all patients in a common atlas space did not lead to better prediction performance, but did speed up the optimum search process : our model built from registered

Table 4: CNN convergence and performance on registered data and influence of three CNN hyper-parameters in the training convergence. In bold are displayed the parameters identified as optimal with regard to the metrics represented in this table.

Panel A: Influence of the number of features

# features	2	4	**8**	16
EPOCHS	330 ±127	275 ±93	185 ±43	120 ±13
OSC	0.2	0.02	0.02	0.01
DIST	0.35	0.3	0.3	0.56
DSC	0.42 ±0.10	0.48 ±0.06	0.50 ±0.07	0.5 ±0.06

Panel B: Influence of the drop out parameter

dropout	**0.5**	0.6	0.7	0.8
EPOCHS	175 ±33	180 ±42	190 ±50	195 ±45
OSC	0.02	0.03	0.14	0.17
DIST	0.35	0.30	0.31	0.3
DSC	0.50 ±0.06	0.48 ±0.07	0.48 ±0.07	0.48 ±0.07

Panel C: Influence of the regularization parameter

regularizer	0.0001	0.0002	**0.0005**	0.001
EPOCHS	200 ±27	180 ±34	165 ±23	200 ±19
OSC	0.03	0.02	0.02	0.01
DIST	0.35	0.31	0.3	0.3
DSC	0.50 ±0.05	0.50 ±0.06	0.51 ±0.05	0.51 ±0.06

images converges faster during the training phase and its chances of falling into a local optimal were reduced.

5 Conclusion

On the basis of several convergence indicators and with regard to a performance indicator linked to the prediction, it has been shown that the registration on a anatomical atlas makes it possible to accelerate the convergence by 38% while maintaining the same set of optimal parameters and an unchanged performance level. This represents an important result for the real world application of CNNs where fine-tuning is costly in computation time, but also crucial to accommodate the complexity of the machine learning task and the underlying data.

References

1. Almuhaideb, S., Menai, M.E.B.: Impact of preprocessing on medical data classification. Frontiers of Computer Science **10**(6), 1082–1102 (2016)
2. Alshdaifat, E., Alshdaifat, D., Alsarhan, A., Hussein, F., El-Salhi, S.M.F.S., et al.: The effect of preprocessing techniques, applied to numeric features, on classification algorithms' performance. Data **6**(2), 11 (2021)
3. Avants, B.B., Tustison, N., Song, G.: Advanced normalization tools (ANTS). Insight journal **2**(365), 1–35 (2009)
4. Debs, N., Cho, T.H., Rousseau, D., Berthezène, Y., Buisson, M., Eker, O., Mechtouff, L., Nighoghossian, N., Ovize, M., Frindel, C.: Impact of the reperfusion status for predicting the final stroke infarct using deep learning. NeuroImage: Clinical **29**, 102548 (2021)
5. Feng, R., Badgeley, M., Mocco, J., Oermann, E.K.: Deep learning guided stroke management: a review of clinical applications. Journal of neurointerventional surgery **10**(4), 358–362 (2018)
6. Fonov, V.S., Evans, A.C., McKinstry, R.C., Almli, C., Collins, D.: Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. NeuroImage **47**, S102 (2009)
7. Fu, Y., Lei, Y., Wang, T., Curran, W.J., Liu, T., Yang, X.: Deep learning in medical image registration: a review. Physics in Medicine & Biology **65**(20), 20TR01 (2020)
8. Giacalone, M., Rasti, P., Debs, N., Frindel, C., Cho, T.H., Grenier, E., Rousseau, D.: Local spatio-temporal encoding of raw perfusion MRI for the prediction of final lesion in stroke. Medical image analysis **50**, 117–126 (2018)
9. Kamal, H., Lopez, V., Sheth, S.A.: Machine learning in acute ischemic stroke neuroimaging. Frontiers in neurology **9**, 945 (2018)
10. Kidwell, C.S., Wintermark, M., De Silva, D.A., Schaewe, T.J., Jahan, R., Starkman, S., Jovin, T., Hom, J., Jumaa, M., Schreier, J., et al.: Multiparametric MRI and CT models of infarct core and favorable penumbral imaging patterns in acute ischemic stroke. Stroke **44**(1), 73–79 (2013)
11. Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). pp. 565–571. IEEE (2016)

12. Nielsen, A., Hansen, M.B., Tietze, A., Mouridsen, K.: Prediction of tissue outcome and assessment of treatment effect in acute ischemic stroke using deep learning. Stroke **49**(6), 1394–1401 (2018)
13. Rekik, I., Allassonnière, S., Carpenter, T.K., Wardlaw, J.M.: Medical image analysis methods in MR/CT-imaged acute-subacute ischemic stroke lesion: Segmentation, prediction and insights into dynamic evolution simulation models. a critical appraisal. NeuroImage: Clinical **1**(1), 164–178 (2012)
14. Winzeck, S., Hakim, A., McKinley, R., Pinto, J.A., Alves, V., Silva, C., Pisov, M., Krivov, E., Belyaev, M., Monteiro, M., et al.: ISLES 2016 and 2017-benchmarking ischemic stroke lesion outcome prediction based on multispectral MRI. Frontiers in neurology **9**, 679 (2018)
15. Woolrich, M.W., Jbabdi, S., Patenaude, B., Chappell, M., Makni, S., Behrens, T., Beckmann, C., Jenkinson, M., Smith, S.M.: Bayesian analysis of neuroimaging data in FSL. Neuroimage **45**(1), S173–S186 (2009)
16. Wu, O., Østergaard, L., Weisskoff, R.M., Benner, T., Rosen, B.R., Sorensen, A.G.: Tracer arrival timing-insensitive technique for estimating flow in mr perfusion-weighted imaging using singular value decomposition with a block-circulant deconvolution matrix. Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine **50**(1), 164–174 (2003)
17. Yu, Y., Xie, Y., Thamm, T., Gong, E., Ouyang, J., Huang, C., Christensen, S., Marks, M.P., Lansberg, M.G., Albers, G.W., et al.: Use of deep learning to predict final ischemic stroke lesions from initial magnetic resonance imaging. JAMA Network Open **3**(3), e200772–e200772 (2020)

This page is intentionally left blank.

A Unified Approach to the Standardization of AI-Centered Audio-Visual Data Processing

Leonardo Chiariglione[1†], Andrea Basso[1], Marina Bosi[2][0000-0001-9851-2570], Sergio Canazza[3], Miran Choi[4], Gérard Chollet[5], Michelangelo Guarise[6], Roberto Iacoviello[7], Niccolò Pretto[3], Paolo Ribeca[8][0000-0001-5599-3933], Mark Seligman[9]

[1] Moving Picture, Audio and Data Coding by Artificial Intelligence
[2] Center for Computer Research in Music and Acoustics, Stanford University, Stanford, USA
[3] Centro di Sonologia Computazionale (CSC) - DEI, University of Padova, Padua, Italy
[4] Electronics and Telecommunication Research Institute, Daejeon, Republic of Korea
[5] Institut Polytechnique de Paris (IMT-TSP), Evry, France
[6] Volumio SRL, Florence, Italy
[7] Rai, Radiotelevisione Italiana, Turin, Italy
[8] Biomathematics and Statistics Scotland, Edinburgh, United Kingdom
[9] Speech Morphing Inc, San Jose, USA
[†] leonardo@chiariglione.org

Abstract. Media compression standards helped the digital media boom. Artificial Intelligence (AI) promises to be the next revolution, but there are no data coding standards for AI tools yet. The mission of the Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI) organization [7,8] is to develop AI-centered standards that will have the same positive effect on industry and consumers as media standards had. MPAI standards rely on the AI Framework (MPAI-AIF). It defines computational Modules (AIMs) that can use AI, Machine Learning (ML), and data processing technologies implemented in hardware, software, and mixed hardware/software. One can then connect AIMs to compose arbitrary workflows. MPAI standards define the interfaces of AIMs but not their internals, making them replaceable, re-usable and upgradable without changing the logic of the application. Through this mechanism, MPAI standards promote horizontal markets of AIMs and innovation, because AIMs can continuously improve by incorporating more efficient technologies. More standards are in preparation supporting various technologies, ranging from audio-video coding to industrial data, server-based gaming, and integrative sensor-genomics analysis. This paper will focus on the standards for which development is more mature.

Keywords: Standardization, Artificial Intelligence, Audio-visual data.

1 Introduction

The amount of data created in 2021 is expected to be 74 ZB and to more than double by 2024, yielding 149 ZB [3]. However, this copious data is used only minimally, because moving and processing it is costly. Data processing-based video and audio coding enabled the television industry and other businesses to thrive; likewise, AI-centered data coding standards can foster the development of all industries – including media-based ones – that produce vast amounts of data requiring a digital representation. Despite a growing interest in the field [2], what has been proposed so far seldom goes beyond providing interoperable formats for the representation of neural networks [16].

The MPAI Manifesto [8] establishes the principles needed to achieve the goal of standardizing AI-centric data coding. In a matter of months, MPAI has developed a work plan [10] covering audio, video, human-machine conversation, financial data, online gaming, and integrated genomic/sensor analysis. One standard is under development and three Calls for Technologies (CfTs) have been published. This paper provides an outline of the work in process, mainly according to its maturity.

2 MPAI methodology

The MPAI standardization process follows a bottom-up approach with seven phases:

1. Use cases are collected and harmonized leading to specific topics
2. Individual use cases are extended and formalized
3. Functional requirements for the technologies are developed to enable use cases
4. The framework license is developed, to be used by whoever holds patents on technologies that will become part of the standard
5. A CfT is developed and published
6. The standard is developed using the technologies proposed
7. The standard is approved and published.

The first three phases are open to the public; the next three are open to MPAI members; and the last phase is the prerogative of MPAI Principal members [9].

At this time, the most advanced project is MPAI-AIF (phase 6); MPAI-CAE and MPAI-MMC are in phase 5. These projects are described in the following sections. Other projects in earlier phases are also briefly introduced.

3 The AI Framework (MPAI-AIF)

The AI Framework (AIF) can create, compose, execute, and update AIM workflows; it constitutes the cornerstone of the MPAI standards, allowing the interconnection of multi-vendor AIMs that are trained for specific tasks, operate within the AIF, and exchange data in standard formats. MPAI is well aware that in this transitional phase, many technologies that AI promises to replace are still used to provide products and services. Therefore, AIF enables the coexistence, interoperation, and mutual replacement of AIMs based on AI, ML, and traditional algorithms.

The MPAI-AIF adopts the component-based development (CBD) philosophy, enabling independent components (AIMs) to be reused within systems. In the AIF, all of the data and functions inside each AIM are semantically related. AIMs communicate with each other via standardized interfaces; these interfaces specify the services that

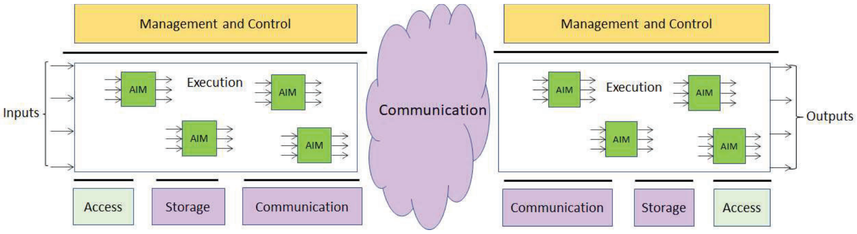


Fig. 1. Conceptual schema of the AI Framework (AIF)

other components can use, and how they can do so. Thus, an AIF client exploiting computational services does not need to know the details of the implementation of an AIM in order to use it. The AIF is based on the requirements described in [11] (see **Fig. 1**).

Interface abstraction is to be achieved through the extensive use of metadata. Profiles will be defined to provide hardware-software (HW-SW) interoperability and a general framework. For instance, in a signaling scenario, the number of SW signals handled can easily be very large, while in HW it will be constrained by the physical wiring; and other constraints will be imposed by the need for SW modules to simulate persistent inter-module connections.

The MPAI-AIF Development Committee (DC) is currently evaluating various execution models with additional hierarchical workflow levels, inheriting concepts from the Message Passing Interfaces (MPI, [14]) standard and the Common Workflow Language (CWL, [1]). The goal is to provide management and control of combinations of AI modules, but also to make possible the interconnection and execution of AIMs in resource-constrained scenarios (MCUs). Depending on the profiles, MPAI-AIF will support event-based as well as signal-based execution and resource management, both at the AIM and workflow levels. According to the execution model chosen, the AIF will provide shared storage and communication mechanisms.

One of the key characteristics of MPAI-AIF is its support for specific ML functionalities, in particular training, retraining, and the dynamic updating of ML components. Registration of AIMs and their associated metadata will be secure and credential-based.

4 Context-based Audio Enhancement (MPAI-CAE)

In the last few years, AI has had a strong impact on audio research [6,15,18,19,21].

MPAI has identified four use cases collected in a single standard project named MPAI-CAE [12]. *Emotion-enhanced speech* concerns an emotion-less synthetic or natural speech segments which is to be enhanced with a particular emotion (e.g., anger) with a specified intensity [20]. In *Audio recording preservation*, sound from an old audio tape is improved and a master file is produced for preservation using a video camera that points to the magnetic head reproducing the tape [17]. The *Audio-on-the-go* experience preserves any external sounds considered relevant, while minimizing any interference with the audio event itself.

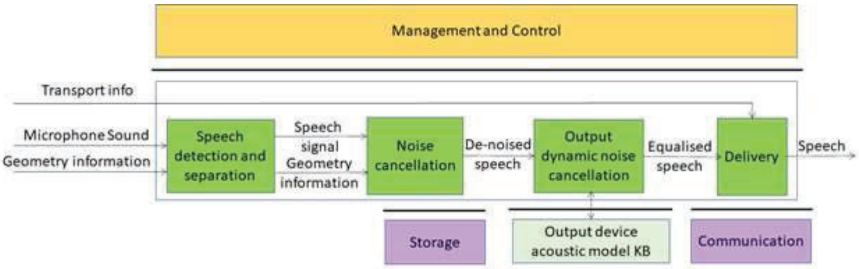


Fig. 2. MPAI-CAE: The Enhanced audioconference experience (EAE) workflow

Enhanced audioconference experience (EAE) aims to improve audio quality and user experience in a video/audio conference environment. The main problem tackled by EAE concerns background noise and undesired sounds that can distract the participants and hinder them from following the ongoing discussion. AI-based noise-cancellation and sound enhancement, along with awareness of microphone placement, can virtually eliminate these problems. The workflow represented in **Fig.** shows how an EAE structure can be implemented [12]. An EAE system receives microphone sound, along with information concerning the microphone array geometry that describes the number of microphones employed, their position, and their configuration. Additional microphone information (e.g., concerning frequency response) can be easily added, though it is not shown in the figure. Based on this information, a *Speech detection and separation* module makes it possible to isolate the material relevant to audioconference from spurious signals. The resulting *Speech signal* is then processed to eliminate any distortion, and further equalized based on the characteristics of the output device. These are derived from an *Output device acoustic model Knowledge Base (KB)*, which describes the features of the specified device (e.g., its frequency response). In this fashion, the relevant speech can be equalized, removing any coloration caused by the output device and yielding an optimally delivered sound experience [5].

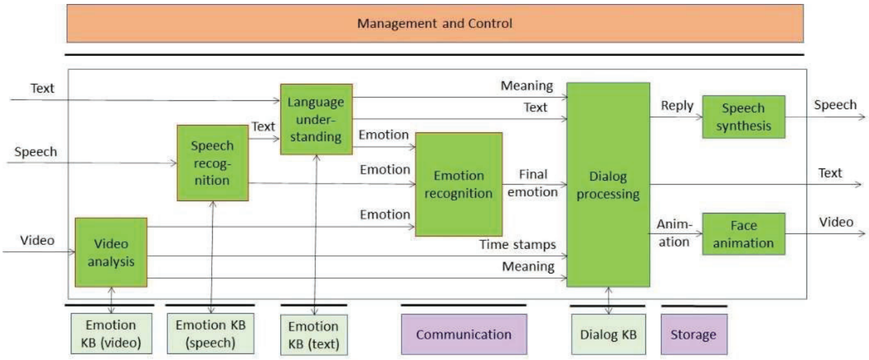


Fig. 3. MPAI-MMC: The Conversation with emotion (CWE) workflow

5 Multi-modal Conversation (MPAI-MMC)

Currently, the MPAI-MMC [13] standard includes three use cases in which a human carries on an audio-visual conversation with a machine, thus emulating human-to-human conversation. In *Multimodal question answering*, a human requests information about a displayed object, and the machine responds with synthesized speech. In *Personalized automatic speech translation*, a human-uttered sentence is translated by a machine using a synthesized voice that mimics the original speaker's speech features.

In *Conversation with emotion* (CWE; see **Fig.**), the human side of the dialogue includes speech, video, and possibly text, while the machine responds with a synthesized voice and an animated face. The computer's replies to the human are informed by automatic recognition of emotion in the user's speech and/or text and video cues from the human's face [4]. First, a set of emotion-related cues are extracted from text, voice, and video by appropriate modules. The outputs of these modules are fused to yield the *Final emotion*, which in turn is transferred to *Dialog processing*. A synthetic reply can then be generated as text or concept [22] based upon the *Final emotion* and *Meaning* derived from this text and video analysis.

6 Other standards being developed

Compression and Understanding of Industrial Data (MPAI-CUI) enables AI-based filtering and extraction of key information from the flow of data produced by companies, thus helping them to assess their risks.

Server-based Predictive Multiplayer Gaming (MPAI-SPG) aims to minimize the discontinuities in visual gameplay caused by network disruptions during an online, cloud-based, real-time game. If information from a client is missing, the gaps are filled out by an AI-based system predicting the moves of the client.

AI promises to further reduce compression rates beyond those achieved so far. AI-Enhanced Video Coding (MPAI-EVC) replaces with AI-based tools the existing tool in the Essential Video Coding (EVC) standard.

Integrative Genomic/Sensor Analysis (MPAI-GSA) uses AI to understand and compress the results of high-throughput experiments combining genomic/proteomic and other data, e.g., from video, motion, location, weather, and medical sensors. Identified use cases range from the integrative analysis of ‘omics datasets to smart farming.

7 Conclusions

According to its mandate, MPAI is developing standards for the coding of data types used by AI-centric technologies. In the first few months of its life, MPAI has produced an approach to standardization, a development process, and a work plan covering the areas where most immediate needs have been identified.

MPAI standards can promote a virtuous circle whereby: (1) technology providers develop and offer conforming AIMs to an open market; (2) application developers can find the AIMs they need on the open market; (3) consumers have a wider choice of better AI applications; and (4) innovation is fueled by the demand for novel and higher-performing AIMs.

MPAI standards offer additional advantages. Today's typical AI-based applications are monolithic and opaque; their inner workings and implications are hard to fathom. By contrast, MPAI standards partition applications into smaller AI modules to enable a market of components. That allows the performance of sub-modules to be assessed separately, and results in a complete application that is significantly more explainable.

References

1. CWL Community website, https://www.commonwl.org/, last accessed 2021/03/07.
2. https://news.itu.int/international-standards-for-an-ai-enabled-future/, accessed 2021/03/19.
3. https://www.statista.com/statistics/871513/worldwide-data-created/, accessed 2021/03/05.
4. Maréchal C., Mikolajewski D., Tyburek K., Prokopowicz P., Bougueroua L., et al.: Survey on AI-based multimodal methods for emotion detection. High-Performance Modelling and Simulation for Big Data Applications, Springer, 307-324 (2019).
5. Martínez Ramírez, M.A., Reiss, J.D.: Modeling nonlinear audio effects with end-to-end deep neural networks. In: IEEE ICASSP, 171–175 (2019).
6. Moffat, D., Sandler, M.B.: Approaches in intelligent music production. Arts 8(4) MDPI (2019).
7. MPAI Community, https://mpai.community/, last accessed 2021/03/06.
8. MPAI Manifesto, https://mpai.community/about/manifesto/, last accessed 2021/03/06.
9. MPAI Statutes, https://mpai.community/about/statutes/, last accessed 2021/03/07.
10. MPAI Work Plan, https://mpai.community/standards/work-plan/, last accessed 2021/03/05.
11. MPAI-AIF, https://mpai.community/standards/mpai-aif/, last accessed 2021/03/07.
12. MPAI-CAE, https://mpai.community/standards/mpai-cae/, last accessed 2021/03/06.
13. MPAI-MMC, https://mpai.community/standards/mpai-mmc/, last accessed 2021/03/06.
14. MPI Forum website and documents, https://www.mpi-forum.org/, last accessed 2021/03/07.
15. Nicholls, S., Cunningham, S., Picking, R.: Collaborative artificial intelligence in music production. In: Proc. Audio Mostly 2018. ACM, New York, USA (2018).
16. ONNX Community website: https://onnx.ai/index.html, last accessed 2021/03/19.
17. Pretto, N., Fantozzi, C., Micheloni, E., Burini, V., Canazza, S.: Computing methodologies supporting the preservation of electroacoustic music from analog magnetic tape. CMJ 42(4), 59–74 (2019).
18. Schedl, M., Bauer, C., Reisinger, W., Kowald, D., Lex, E.: Listener modeling and context-aware music recommendation based on country archetypes. Front Artif Intell 3, 108 (2021).
19. Schubert, E., Canazza, S., De Poli, G., Rodà, A.: Algorithms can mimic human piano performance: The deep blues of music. J. New Music Research 46(2), 175–186 (2017).
20. Tits, N.: A methodology for controlling the emotional expressiveness in synthetic speech - a deep learning approach. In: Proc. ACIIW, 1–5 (2019).
21. Widmer, G.: Getting closer to the essence of music: The "Con Espressione" manifesto. ACM Trans. Intell. Syst. Technol. 8(2) (2016).
22. Young, S.J., Fallside, F.: Speech synthesis from concept: A method for speech output from information systems. JASA 66(3), 685–695 (1979).

An Eye Tracking Based Solution for Reading Related Disorders Detection

Martin Janiga[1] and RadoslavVargic[1]

[1]FEI STU in Bratislava, Ilkovičova 3, 841 04 Bratislava, Slovakia

martinjaniga451@gmail.com

Abstract. In this contribution, we present a system for reading related disorder detection. The system uses known dataset with low- and high-risk subjects. The proposed method is evaluated along with other approaches. The results show similar performance of the proposed method as the used reference methods.

Keywords: Eye-tracking, Data classification, Disorder detection.

1 Introduction

Eye-sight is one of the main human senses, that allow us to recognize various text, shape of different objects, colors, light and orientation in space. The term health disability can cover wide range of various physical or mental disorders - reflects detailed deviations from the norm. One of them is dyslexia [6], which affects 5-10% of whole population. The term was used by Rudolf Berlin for the first time in 1887.It comes from the Greek words: lexis-verbal expression, speech, language and a prefix „dys''-presents something broken. Till today there is no precise definition of this cognitive disorder. It can be defined as chronical - neurological developing reading disorder of people having unbroken intelligence. It can be reflected in the early school age and indicates the level of reading and writing, which is in conflict with the detected level of intellectual abilities. Statistically, boys are affected 3 times more than girls.

Usage of eye tracking is one of the most common diagnostic tools in this specific area. Topic eye movement when reading is very alive. By recording eye movement while reading text we can detect various cognitive disorders. In the process of reading the text, readers move their eyes from one word to another by rotating fixations (points in which the eyes are not moving but focused on the concrete word) and saccades (points in which eyes are moving between words) Saccades, that moves a reader forward in the texts are called progressive. On the other hand saccades, which force a reader to return in the text to the previous points are called regressive saccades. We confirm that eye movements of dyslectic readers are different from the typical readers without dyslexia.

An individual having dyslexia has different eye movement – more regressive saccades, fixations and frontal saccades.

In our contribution we will work on correct classification of dyslectic and non-dyslectic people and divide them into separate groups based on the eye movement monitoring while reading a text and based on subsequent data processing. We describe here in more detail the selected dataset and research of the authors of the dataset. Afterwards we describe our experiment using the dataset.

2 Related work

Several researches and evaluations dealing with difficulties and abnormalities in reading, using eye tracking were conducted.

A novel fast-screening method for reading difficulties with special focus on dyslexia under name s presented in [2]. It is a new, fast, non-invasive method called Rapid Assessment of Difficulties and Abnormalities in Reading (RADAR) - Quick assessment of reading difficulties and abnormalities that screens for features associated with the aberrant visual scanning of reading text seen in dyslexia Measurement of parameters of vision monitoring, which are stable when re-testing and have a high discriminant analysis as is indicated by their curves ROC (receiver operating characteristic) were obtained during text reading. These parameters were combined to derive a total reading score (TRS) that can reliably separate readers with dyslexia from typical readers. We tested TRS in a group of school-age children from 8.5 to 12.5 years. TRS achieved 94.2% correct classification of children tested. Specifically, 35 out of 37 control (specificity 94.6%) and 30 out of 32 readers with dyslexia (sensitivity 93.8%) were classified correctly using RADAR, under a circular validation condition. 9 participants (3 non-dyslexics and 6 dyslexics) were rejected due to unreliable eye movement recording or insufficient cooperation with people issuing experiments. Each child in our experiment had to have an IQ greater than 90% and also eye acuity (the ones who taken correction or uncorrected) and also great hearing sense.

In [4] was conducted research related to eye movements of dyslexic children when reading in a regular orthography. Participants were German dyslexic readers (13-year-olds) who compared to English dyslexic readers-suffer mainly from slow laborious reading and less from reading errors. The eye movements of eleven dyslexic boys and age-matched controls were recorded during reading of text passages and pseudoword lists. For both text and pseudoword reading, the dyslexic readers exhibited more and much longer fixations, but relatively few regressions. Increased length of words and pseudowords led to a greater increase in number of fixations for dyslexic than normal readers. Comparisons across studies suggest that the present German dyslexic eye movement findings differ from English-based findings by a lower frequency of regressions (presumably due to the higher regularity of German) and from Italian findings by longer fixation duration (presumably due to the greater syllabic complexity of German).

The research related to Predictive Model for Dyslexia from Fixations and Saccadic Eye Movement Events was described in [1]. In this research a small set of eye movement features have been proposed that contribute more to distinguish between dyslexics and non-dyslexics by machine learning models. Features related to eye movement events such as fixations and saccades are detected using statistical measures, dispersion threshold identification (I-DT) and velocity threshold identification (I-VT) algorithms. These features were further analyzed using various machine learning algorithms such as Particle Swarm Optimization (PSO) based SVM Hybrid Kernel (Hybrid SVM – PSO), Support Vector Machine (SVM), Random Forest classifier (RF), Logistic Regression (LR) and K-Nearest Neighbor (KNN) for classification of dyslexics and non-dyslexics. The accuracy achieved using the Hybrid SVM –PSO model is 95.6 %. The best set of features that gave high accuracy are average number of fixations, average fixation gaze duration, average saccadic movement duration, total number of saccadic movements. It is observed that eye movement features detected using velocity-based algorithms performed better than those detected by dispersion-based algorithms and statistical measures.

3 Selected dataset

In our experiment, we used a dataset provided by Swedish research, [5] which contains the movements of the right and left eyes in X and Y coordinates and information of time. Eye movement was issued while reading s short text. A text was read by 185 individuals, who were attending 3rd class in age of 9-10 years. They were divided into 2 groups. In the first group there were 97 high-risk individuals who were diagnosed with dyslexia – called HR group – 76 boys and 21 girls. The second group of LR consisted of 88 low-risk individuals without dyslexia- 69 boys and 19 girls. The text that individuals read consisted of 10 sentences divided into 8 lines with the average word length 4,6 letters and was in Swedish. An individuals were wearing a pair of light 80g individually adjustable glasses mounted on the head , having 4 sets of infra-red transmitters and detectors organized to squares around each eye. To prevent a head movement and obtain a stability of 45 cm viewing distance they set chin rest and forehead rest. Calibration was performed manually before the recording by adjusting the signal of each axis separately for each eye. The subject has recorded an eye movements while reading a text. Their approach stands on the analysis of eye movements from measured data using the algorithm dynamic dispersion based on which they determined fixations, saccadic movements and other characteristics. They used these data to train a classification model. Saccades were divided into two types: forward and reverse. Subsequently the following parameters were defined for fixations and saccades :

- duration
- distance
- average eye position
- standard deviation of variable position
- maximum range between 2 positions
- Accumulated distance in all following positions

With this approach they acquired 168 features capturing various quantitative parameters of the eyes when text reading such as duration, amplitude, direction, stability, asymmetry. To classify an individualwe used Support vector machine method – 10 multiple cross validation provided an assessment of predictive performance. This process was repeated 100 times. To reduce abundant symptoms, which did not contribute to final classification we used RFE method. In this method we have selected the best features (48 in total) and the results succeed as 95,6% ± 4,5%, TPR 95,5% ± 4,6% a TNR 95,5% ± 4,5%. In other method where the signs were chosen randomly the success was reduced to 91,1% ± 6,1. When selected 126 random features in each training folder , they achieved the best classification as 95,3 % ± 4,6%, TPR 95,2% ± 4,7%, TNR 95,5% ± 4,5%. The results showed that eye movements while reading can be highly predictive for ability of individual reading and that eye tracking can be an effective instrument for identification children threatened of long term reading problems.

4 Experiment

Our aim is to schedule selected subjects into the LR and GR groups. The data acquired from the Swedish site contained data of 185 subjects. Each of them was written in a text file, which contained an information of time and position of both eyes on axes X and Y. We have selected 20 individuals for our experiment– 5 boys and 5 girls from LR group and also 5 boys, 5 girls from HR group. Additionally, we were work with data about time and position of the eye view on the axe X. For simplicity the acquired data were averaged by using a formula

$$X = \frac{Lx + Rx}{2}$$

Furthermore, we have used a program Matlab, in which we cut the data on an active process from the beginning of the first line to the end of the 7th line including moving a view to the beginning of the 8th line. In the Fig. 1 is shown how the data were cut. In the next step we issued k-means clustering in the program Matlab.

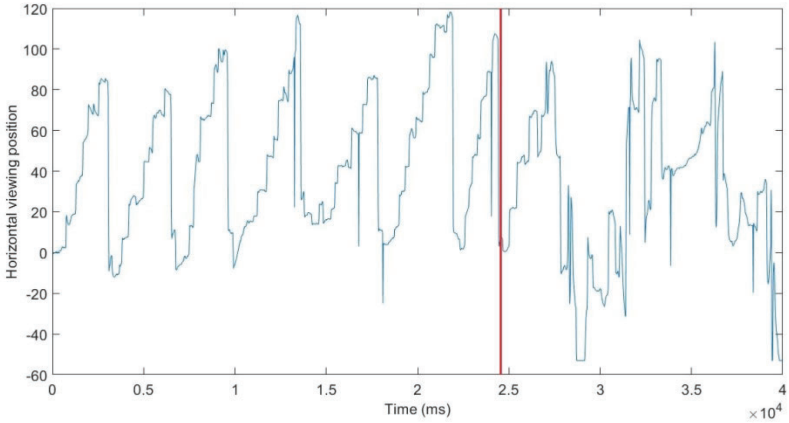


Fig. 1. Example of input gaze position data and cut od the data when reaching the bottom of the text (red line)

Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different. In other words, we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application-specific.

K-means[3] belongs to the group of training without a teacher. This is because we do not have a base for comparing the output. We just want to try to examine a structure of data by grouping data into various subgroups. Due to its simplicity is considered as one of the most used clustering algorithms.It is iterative algorithm, which try to divide a set of points into groups (clusters), where each data point belongs to one group. It assigns cluster points the way that the summary of distances amplified a square power between the data points and the cluster centroid (the arithmetic average of all data points, which belongs to that cluster) is minimal.

The way k-means works as below:

We enter the number of cluster k. To the point space the centroids are added on the random position to which the nearest points are assigned. Centroids are then moved to the center of gravity of the point cluster and reassigned to the nearest point.This process is repeated till the position of the centroids get stabilized. Subsequently after this process, the centroids are removed and we get a new marked points – clusters. In out method we verified a number of clusters from 5 to 100 and by usingsilhouette metrics we have chosen optimal k. The silhouette value for each point is degree of similarity of this point to other points in the same cluster compared to points in other clusters. Fig. 2 shows the assignment of points to clusters using k-means.

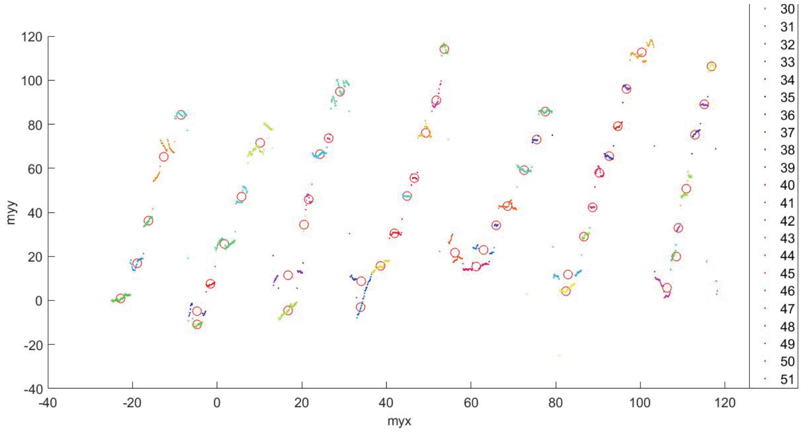


Fig. 2. Example of assigning clusters using k-means

5 Results

Method 1

After application of k-means algorithms to all our selected subjects we put down the information of number of clusters and time in a tab and displayed them by using Matlab in a graph in Fig. 3. An axis X shows a time used by our readers for reading a text and an axis Y presents a number of acquired clusters using algorithm k-means. It can be reflected from the graph that dyslectics read a text much longer than non-dyslectics, but classification using a number of clusters is quite inaccurate and we cannot use it to classify subjects to the correct group.

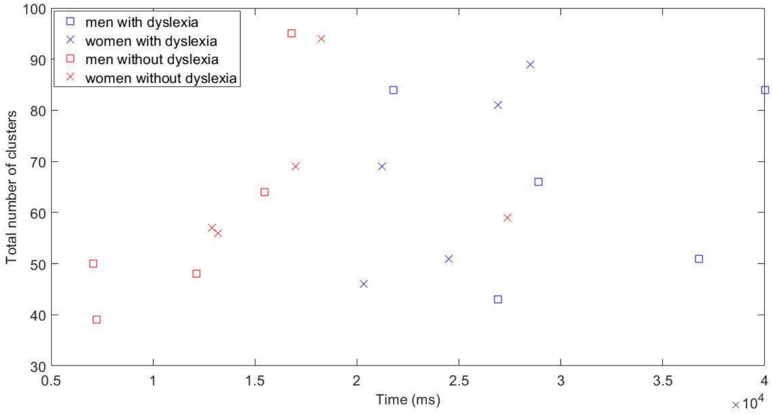


Fig. 3. Time and total cluster count

Method 2

In the next step we tried to better arrange the point position on Y-axis using the distance of eye movement in a horizontal position, which we calculated as a summary of value differences of adjacent points on the X-axis.

Dyslexics record more regressive saccades compared to non-dyslectic readers. Based on this we point out an assumption that a total distance in a horizontal eye movement is greater for dyslectics. Fig. 4. shows the selected subjects, where the X-axis indicates the time and the Y-axis indicates the total distance of eye movement.

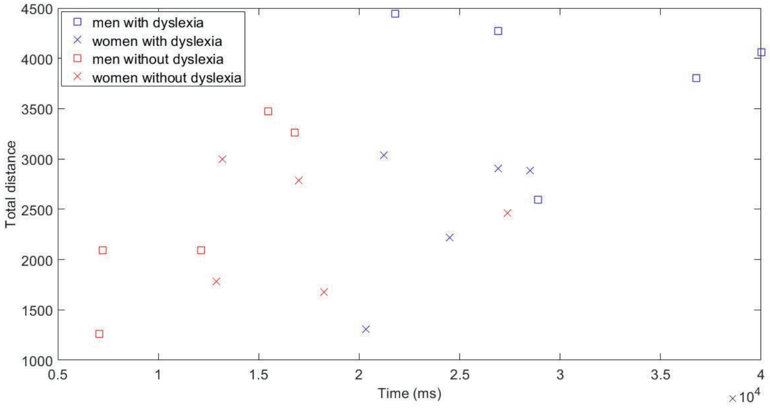


Fig. 4. Time and total distance

It has been shown that an average path in dyslectics was 3153.04 and non-dyslectics 2388.85, but this is not enough for sufficient accuracy of classification of subjects.

Method 3

Based on the assumption that people with dyslexia need a longer time to decode a specific words, we believe that clusters of these subjects will contain more points than clusters of health individuals. We justify the fact that eye movement of people with dyslexia while reading record more frequent and longer fixations on certain words. As we affect Y- axis, it should be reflected at the vertical position of the points and of imaginary separation of set points in the graph. After analyzing the data we found out that clusters of health individuals in comparison to individuals with dyslexia contain on average a much lower number of points. In our graph we included only clusters with the number of points higher than 15. Health individuals we forced closer to 0 while the ones with dyslexia were influenced less are showed in the upper corner of the graph. In the Fig. 5. we present distribution of subject by using of oblique line in which axis X presents a time and axis Y indicates the number of clusters, where the volume of points in cluster is bigger than 15 in %. Within this specific research method was a succeed rate 95% , only 1 LR individual was classified to group HR.

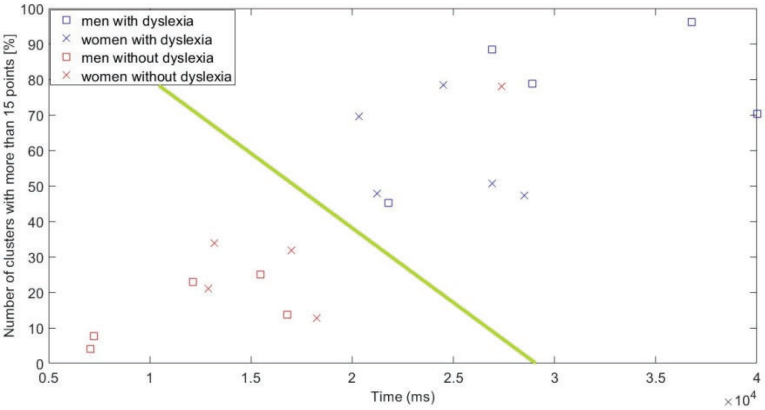


Fig. 5. Time and cluster count

6 Conclusion

The aim of this experiment was to correctly include subjects with dyslexia and without dyslexia in HR and LR groups using eye tracking. In our experiment, we used a dataset from Swedish research that contains data of the positions of the subjects' eyes over time. We selected 20 subjects, 10 from HR group and 10 LR from the mentioned dataset. We created features such as the overall distance of eye movement, the number of clusters, and the number of clusters with a specific content of points. We achieved the best results in the third attempt, in which we only counted the clusters with the number of points higher than 15. Our success rate was 95%.

Acknowledgments

The research in this paper has been supported by VEGA 1/0440/19, KEGA 015STU-4/2021 and the Excellent creative team project VirTel.

References

1. A J. Prabha, & R. Bhargavi. (2020). Predictive Model for Dyslexia from Fixations and Saccadic Eye Movement Events, Computer Methods and Programs in Biomedicine. Dostupné na Internete: ScienceDirect: https://www.sciencedirect.com/science/article/pii/S0169260720300377

2. Crete, T. E., Optotech Ltd., Emmetropia Eye Institute, Medotics Ltd, Athens Medical School, Jules Gonin Eye Hospital, & Department of Neurology. (11. August 2017). NBCI. (X. Weng, Editor) Dostupné na Internete: RADAR: A novel fast-screening method for reading difficulties with special focus on dyslexia: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5553666/

3. Dabbura, I. (17. September 2018). K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks. Dostupné na Internete: towards data science: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

4. F. Hutzler, & H. Wimmer. (2004). Eye movements of dyslexic children when reading in a regular orthography. In Brain and Language. Dostupné na Internete: ScienceDirect: https://www.sciencedirect.com/science/article/pii/S0093934X03004012

5. M. N. Benfatto, G. O. Seimyr, J. Ygge, T. Pansell, A. Rydberg, & C. Jacobsen. (9. December 2016). Screening for Dyslexia Using Eye Tracking during Reading. Dostupné na Internete: PLOS ONE: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0165508

6. Sally E. Shaywitz, M. (29. January 1998). The NEW ENGLAND JOURNAL of MEDICINE. Dostupné na Internete: Dyslexia: https://www.nejm.org/doi/full/10.1056/nejm199801293380507

This page is intentionally left blank.

Eyetracking using Device Camera for Distance Learning Videoconferencing Solutions.

Dominik Cisár[1] and Radoslav Vargic[1]

[1] FEI STU in Bratislava, Ilkovičova 3, 841 04 Bratislava, Slovakia
xcisar@stuba.sk

Abstract. In this contribution we present a system for eye-tracking using device camera. The system covers common devices with camera such as smartphones, notebooks, System is designed for videoconferencing and distance learning support, where it is often important to know if the participants are watching the screen and which part of it. We present preliminary evaluation results and detailed description of the whole system.

Keywords: Eye-tracking, Videoconferencing, Distance learning.

1 Introduction

Eye tracking and more specifically gaze tracking systems are subset of a broader area of computer vision. Computer vision systems capture images of the real world and process them in some way [1]. The captured data can be in a form of various sensor data, camera images or even sound. Gaze tracking specifically focuses on estimating the precise point of gaze of a user by processing image data. Applications of gaze tracking include using a gaze tracker as an input device to control a computer, by simulating a mouse or a keyboard, or to assist users unable to use a computer in other ways. Another application of gaze tracking is recording and assessing gaze data to evaluated students' learning habits to provide the appropriate support to student with reading difficulties [2]. While there are many methods and approaches to gaze tracking, many available systems and implementations are inaccessible to an average user, being either too costly or too cumbersome to use, which may not be practical for the problem at hand. Therefore, we propose a gaze tracking system that only uses a simple web camera as an input device, while also using only free to use software resources.

2 Related work

Gaze tracking systems can be categorized by the type and number of cameras being used to capture data. Either a single camera is used [3]–[7], [11]–[13], or there was a need for a more complex setup, making use of infrared light sources and infrared cameras. In the case of single-camera systems, either regular, or special light sensitive cameras were used, although studies focusing on regular cameras were more numerous.

Approaches that made use of light sensitive cameras had to focus less system resources on image processing to compensate for light variation. Although, the precision was not desirable. The authors of [6] and [12] observed a 5-degree precision. System where a single regular camera was used were overall more portable. The achieved results varied greatly because many different methods were used. Precision ranged between 2.27 and 7 degrees in [3]-[5],[7] and [11]. A specific category was infrared light-based systems. Those approaches had good results in general because corneal reflections were used [8]. The precision varied greatly based on the number of infrared lights and the type of camera being used. In [8], and [9] precision of less than 1 degree was observed and in [10], only 0.6-degree deviation was achieved, although there was no head movement expected. In setups with multiple infrared cameras and/or light sources, reflections, sun light and variation in equipment placement had to be considered so application was limited.

3 Chosen methods

3.1 Face detection

The first step of the processing pipeline is detection of the face in a video frame. The area of the face needs to be detected before facial feature detection can take place. The methods being applied include histogram of oriented gradients a linear classifier, image pyramid and sliding window detection. Once the general area of facial features is determined, a different detector is applied, using a cascade approach [14].

3.2 Eye detection

Once the locations of facial features are acquired, important features of the eyes can be isolated. Those features are namely the inner and outer corners of the eyes and the centers of the pupils. Locations of the corners of the eyes are already known, from the facial feature detector, but the centers of the pupils need to be estimated. This can be achieved by applying a set of morphological operators to the image. [15]

3.3 Gaze vector and calibration

To translate the eye features to a point on the screen, the locations of the corners of the eyes and centers of the pupils are reduced to a set of two vectors. Then a mapping function is estimated, using a set of calibration points on the screen. To acquire the function, the least squares method is used, and the computed function is quadratic [16]. The mapping function is calculated for both eyes separately, for both the x and y axis.

4 Implementation

The implementation of the proposed system could be split into multiple distinct parts. Gaze point estimation, mapping, calibration, eye and facial feature detection and video stream processing. Each part succeeds the next part in the pipeline. The programming language chosen for the implementation was C++, namely because of the plethora of available libraries. The two main libraries required for image processing were OpenCV and Dlib, Function of the Dlib library were used for face detection and OpenCV was used for the rest. The system needs to work in real time and since the time required to process each frame may be different from the interval in which the web camera captures images. Thus, a synchronization method needed to be implemented. A separated thread tasked with capturing new frames had to be created to reduce the time the main thread waits for a new camera frame. This thread continuously saves the newest camera frame to a memory area shared with the main thread, and a synchronization process ensures the data does not get overwritten while it is being read. A significant part of the time required to process a frame takes up copying the frame data, since it is a rather large data structure. Therefore, there was a need to reduce the number of times the frame needs to be copied to a minimum.

4.1 Face detection

Both the detection of the face and facial feature detection are done using pre-trained models, designated to be used with the Dlib library. Regression trees are the basis for the detection, as previously stated. Since the rest of the program uses mainly functions of the OpenCV library, and different image formats are used in each case, conversion is used, OpenCV image instances are encapsulated in Dlib headers, and no data have to be copied. Results of both the face detection and facial feature detection can be seen in Fig. 1. To stabilize facial features detected in subsequent frames, optical flow stabilization and a Kalman filter are applied.

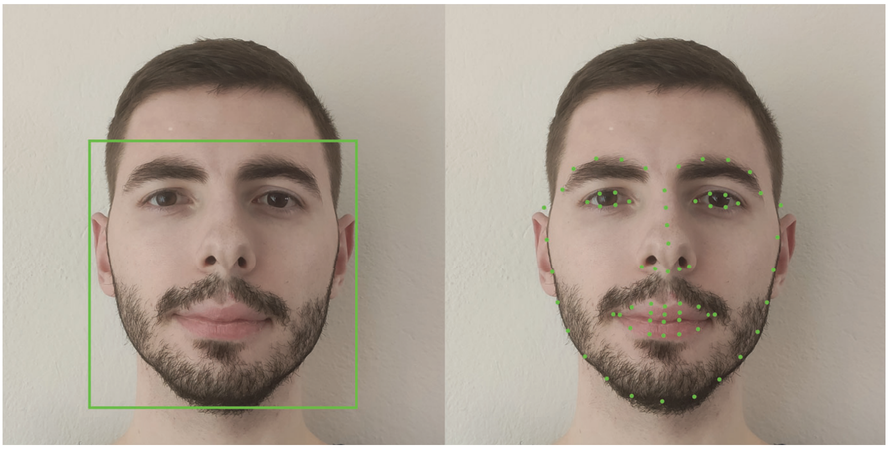


Fig. 1. Results of face detection and facial feature detection.

4.2 Pupil center detection

Position of eye corners are acquired from the facial feature detector, but center of the pupil have to be calculated separately. As previously stated, several morphology operations are applied, all provided by the OpenCV library. The general area of each of pupils is first isolated using the facial features from the previous step. The resulting images are then converted to grayscale, and using the six points surrounding the eyes, the resulting rectangle is further reduced to a hexagonal area, encapsulating the sclera, the iris, and the pupil. To reduce the inherent image noise, a bilateral filer is applied. While it is slower than other blur filters, it gives good results when it is important to preserve edges in the image. Next an erosion filter is applied to remove artefacts. The structural element is a square with a side of three pixels. To compensate for varying lighting condition an adequate binarization threshold is estimated for each of the eyes. The images are binarized with 19 different thresholds and the appropriate one is chosen based on the ratio of white to black pixels in the resulting binary images. The results of different threshold being applied can be seen in Fig. 2 and Fig. 3.

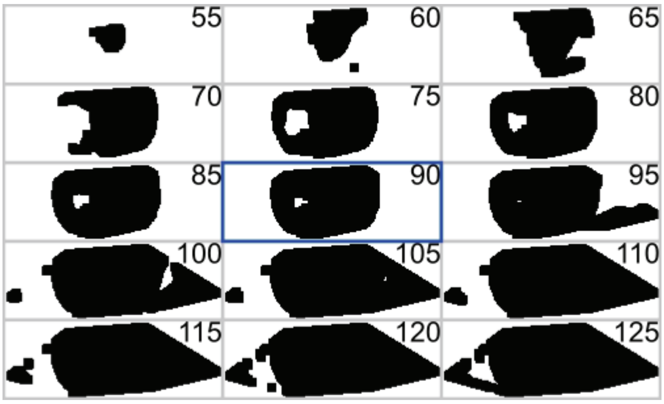


Fig. 2. Threshold chosen for the left eye

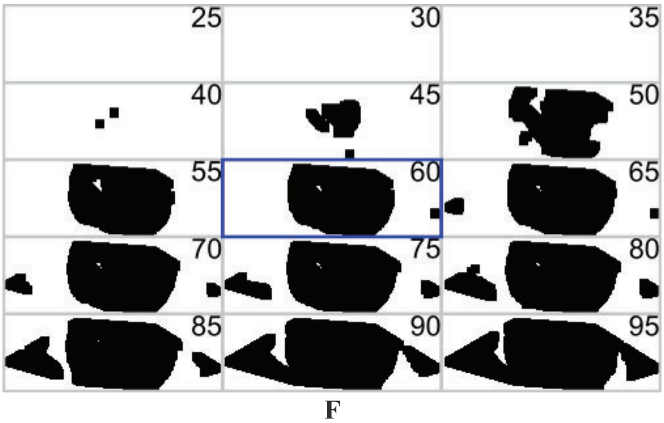


Fig. 3. Threshold chosen for the right eye

The result is a single black and white image for each of the eyes, The content of the image should be a single blob of black pixels containing the iris, surrounded by white pixels. To find the center of the pupil, contours of the blobs are acquired and in the case of there being multiple groups of black pixels, the largest one is chosen, since the smaller groups are likely to be artefacts. Center of the pupil is calculated using the centroid of the largest group. Flowchart of the whole process can be seen in Fig. 4.

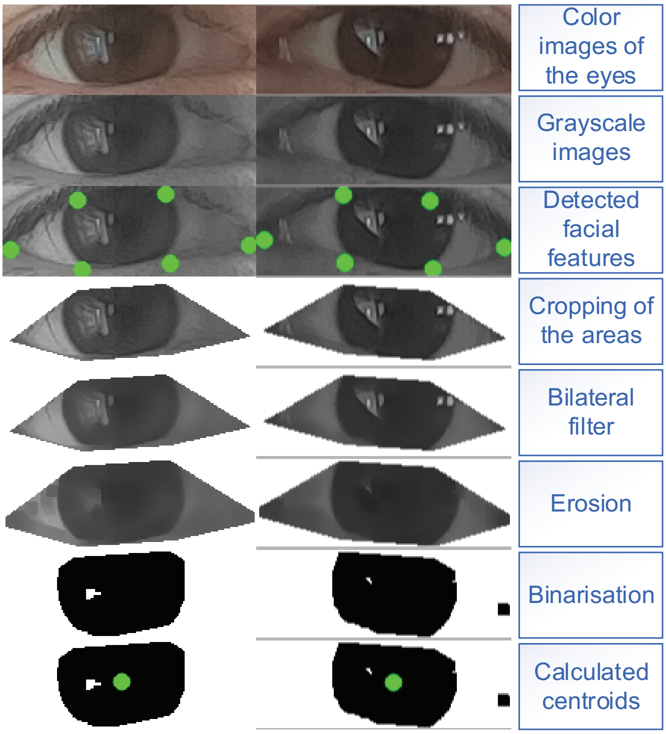


Fig. 4. Eye image processing

4.3 Calibration

Corners of the eyes and the center of the pupil are used to calculate a vector that can be later mapped to a location on the screen using calibration. The calibration procedure starts by creating a set of 25 points on the screen. A user is meant to click at each of the points while looking at them. This way all the data required for calibration is collected. Four mapping functions, one for each of the x and y axes for both eyes are calculated. As previously stated, the least squares method is used. A screenshot of the application with the 25 calibration points being displayed can be seen in Fig. 5. The processed feed from the webcam is also displayed to provide feedback.

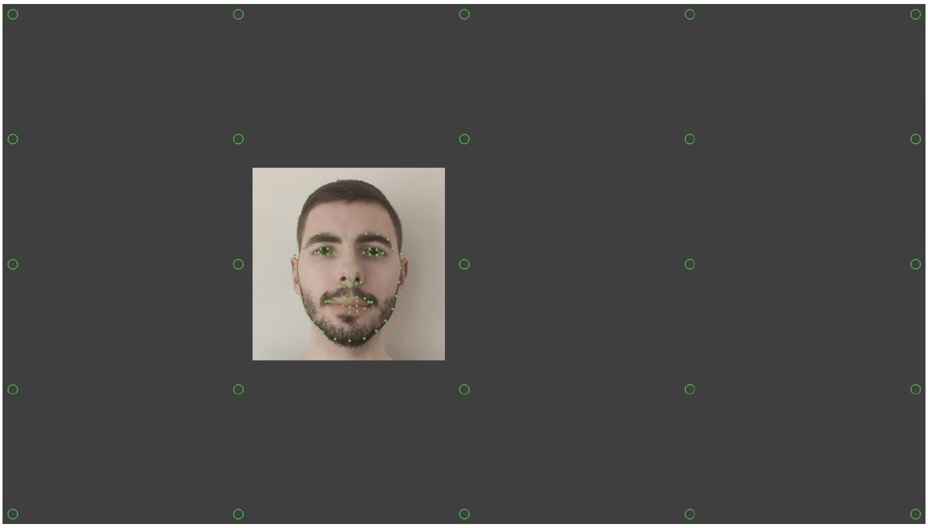


Fig. 5. Calibration points being disaplayed

Gaze tracking

After the gaze tracker is calibrated, an estimated point of gaze starts being drawn. A different one is being calculated for each of the eyes and a third, final one is calculated as the average of the two. In Fig. 6, the white circle is the average, and the red and blue circles are the ones specific to each of the eyes.

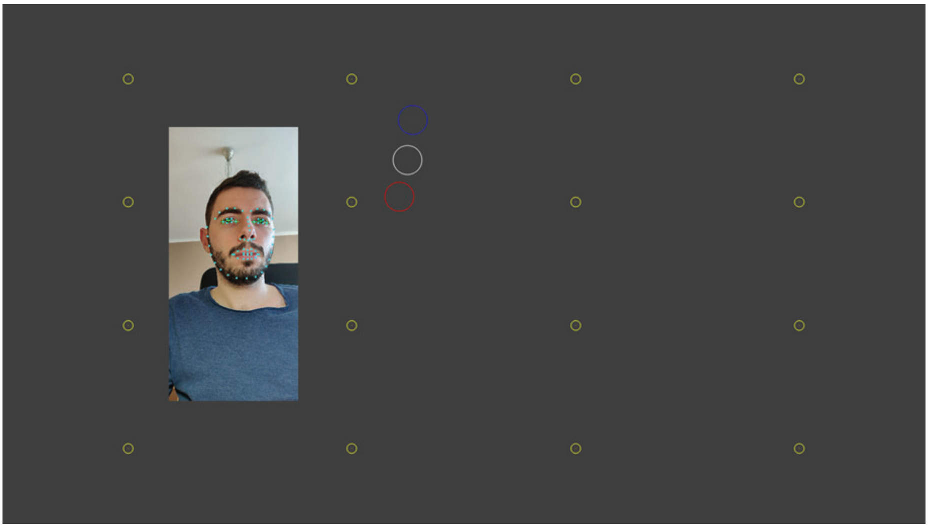


Fig. 6. Gaze point estimation

5 Evaluation

To assess the performance of the system, tests were done with the help of 3 test subject. Each of were placed 50 cm from and screen and performed two calibration and evaluation sessions. Since the system does not compensate for movement of the head, the subjects were asked to keep their heads still if possible. However, a small movement could not be avoided. After the systems was calibrated a set of new 16 points was displayed, and the subjects were asked to click on the new points as well. After each click, a set of 8 coordinates a few frames apart was captured. Estimated points of gaze were compared with the actual coordinates of the 16 points on the screen. The difference between the actual points on the screen and the estimated ones can be seen in Fig. 7 and Fig. 8 the results from the 16 points are combined to display the spread by which the estimated coordinates deviated.

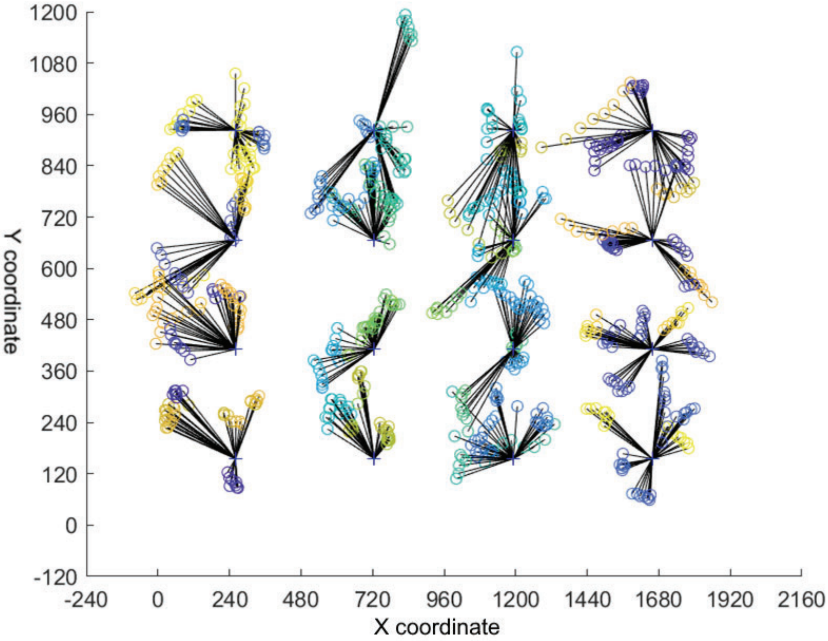


Fig. 7. Combined test results

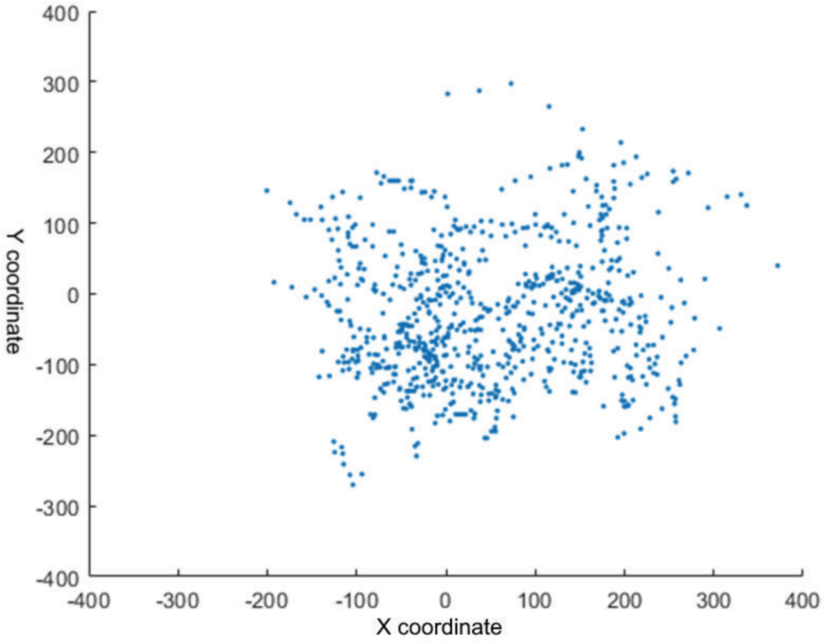


Fig. 8. Combined spread of the estimated points

From the test results we can conclude the system does work, although it important to point out that the test subjects were asked to keep their heads still, so the results are only comparable to systems with similar constraints. The results were consistent with similar systems described in the literature. In Fig. 8. a slight bias to the right can be observed, which could be cause by the camera placement. A great source of error was the facial feature detector, even though stabilization was applied, the locations of eye corners was not precise enough, which could be solved by applying a better detector. Numerical result from the tests can be seen in Table 1. Precision is in both degrees and the number of pixels can is shown.

Subject	Test	Presicion [px]		Presicion [°]	
		x axis	y axis	x axis	y axis
A	1	61.48	91.89	1.22	1.82
A	2	60.16	63.81	1.19	1.27
B	1	49.48	108.02	0.98	2.14
B	2	95.11	91.52	1.89	1.82
C	1	126.46	54.71	2.51	1.09
C	2	180.68	92.29	3.59	1.83
	minimum:	49.48	54.71	0.98	1.09

Table 1. Table captions should be placed above the tables

6 Conclusion

We described and implemented a functional gaze tracking system with the limitation of requiring little to no head movement to function with desirable precision. Observed median precision was 78.3 pixels horizontally and 91.71 pixels vertically, corresponding to 1.55 and 1.82 degrees, respectively. This is consistent with similar systems described in the literature. A further future expansion of the system would include a type of head movement compensation, which might require a more robust mapping function. Application of a more complex facial feature detector meant for processing video, rather than single frames would also give better results, since the calibration data would be more precise.

Acknowledgements

The research in this paper has been supported by VEGA 1/0440/19, KEGA 015STU-4/2021, 2020-1-CZ01- KA226-VET-094346 DiT4LL project and the Excellent creative team project VirTel.

References

1. A. Kadambi, A. Bhandari, and R. Raskar, Computer Vision and Machine Learning with RGB-D Sensors. 2014, pp. 3–26, isbn: 978-3-319-08650-7. doi: 10.1007/978-3-319-08651-4.

2. Lexplore AB, About Us, 2019.

3. Y. M. Cheung and Q. Peng, "Eye Gaze Tracking with a Web Camera in a Desktop Environment", IEEE Transactions on Human-Machine Systems, vol. 45, no. 4, pp. 419–430, Aug. 2015, issn: 21682291. doi: 10.1109/THMS.2015.2400442.

6. K. Wang and Q. Ji, "Real Time Eye Gaze Tracking with 3D Deformable Eye-Face Model", in Proceedings of the IEEE International Conference on Computer Vision, vol. 2017-Octob, IEEE, Oct. 2017, pp. 1003–1011, isbn: 9781538610329. doi: 10 . 1109 / ICCV . 2017 . 114.

5. J. Chen and Q. Ji, "3D gaze estimation with a single camera without IR illumination", in 2008 19th International Conference on Pattern Recognition, IEEE, Dec. 2009, pp. 1–4, isbn: 978-1-4244-2174-9. doi: 10 .1109 / icpr. 2008 . 4761343.

6. F. Vicente, Z. Huang, X. Xiong, F. De La Torre, W. Zhang, and D. Levi, "Driver Gaze Tracking and Eyes off the Road Detection System", IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 2014–2027, Aug. 2015, issn: 15249050. doi: 10.1109/TITS. 2015.2396031.

7. X. Xiong, Q. Cai, Z. Liu, and Z. Zhang, "Eye gaze tracking using an RGBD camera", in Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous

Computing Adjunct Publication - UbiComp ’14 Adjunct, New York, New York, USA: ACM Press, 2014, pp. 1113–1121, isbn: 9781450330473. doi: 10.1145/2638728.2641694.

8. D. Beymer and M. Flickner, "Eye gaze tracking using an active stereo head", 2003, pp. II–451–8, isbn: 0-7695-1900-8. doi: 10.1109/cvpr.2003.1211502.

9. J. Chen, Y. Tong, W. Gray, and Q. Ji, "A robust 3D eye gaze tracking system using noise reduction", in Proceedings of the 2008 symposium on Eye tracking research & applications - ETRA ’08, New York, New York, USA: ACM Press, 2008, p. 189, isbn: 9781595939821. doi: 10.1145/1344471.1344518.

10. E. D. Guestrin and M. Eizenman, "Remote point-of-gaze estimation requiring a single-point calibration for applications with infants", in Proceedings of the 2008 symposium on Eye tracking research & applications - ETRA ’08, New York, New York, USA: ACM Press, 2008, p. 267, isbn: 9781595939821. doi: 10.1145/1344471.1344531.

11. H. Yamazoe, A. Utsumi, T. Yonezawa, and S. Abe, "Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions", in Proceedings of the 2008 symposium on Eye tracking research & applications - ETRA ’08, New York, New York, USA: ACM Press, 2008, p. 245, isbn: 9781595939821. doi: 10.1145/1344471.1344527.

12. C. Morimoto, A. Amir, and M. Flickner, "Detecting eye position and gaze from a single camera and 2 light sources", in Object recognition supported by user interaction for service robots, vol. 4, IEEE Comput. Soc, 2003, pp. 314–317, isbn: 0-7695-1695-X. doi: 10.1109/icpr. 2002.1047459.

13. S. S. Mohapatra and K. Kinage, "Iris tracking using a single web-cam without IR illumination", in Proceedings - 1st International Conference on Computing, Communication, Control and Automation, ICCUBEA 2015, IEEE, Feb. 2015, pp. 706–711, isbn: 9781479968923. doi: 10.1109/ICCUBEA.2015.144.

14. V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, Jun. 2014, pp. 1867–1874, isbn: 9781479951178. doi: 10.1109/CVPR.2014.241.

15. Kishor Datta Gupta, Automatic pupil detection and extraction by c# | Kishordgupta’s Blog, 2010.

16. Z. R. Cherif, A. Naït-Ali, J. F. Motsch, and M. O. Krebs, "An adaptive calibration of an infrared light device used for gaze tracking", Conference Record - IEEE Instrumentation and Measurement Technology Conference, vol. 2, pp. 1029–1033, 2002. doi: 10.1109/IMTC.2002.1007096.

14. V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, Jun. 2014, pp. 1867–1874, isbn: 9781479951178. doi: 10.1109/CVPR.2014.241.

15. Kishor Datta Gupta, Automatic pupil detection and extraction by c# | Kishordgupta’s Blog, 2010.

16. Z. R. Cherif, A. Naït-Ali, J. F. Motsch, and M. O. Krebs, "An adaptive calibration of an infrared light device used for gaze tracking", Conference Record - IEEE Instrumentation and Measurement Technology Conference, vol. 2, pp. 1029–1033, 2002. doi: 10.1109/IMTC.2002.1007096.

This page is intentionally left blank.

Application of Immersive Technologies for Virtual Teleconference

Dominik Bilik[1], Adam Martiška[1], Jakub Otruba[1] and Marek Vančo[1]

[1] Faculty of Electrical Engineering and Information Technology STU in Bratislava,
Ilkovičova 3, 812 19 Bratislava, Slovakia

Abstract. The immersive ways of communication using video are quite lacking. This paper introduces one of many means to improve on this problem, namely on a simplex communication in real time. It presents an application, which uses stereo camera setup on a presenter side, supports multiple presenters and displays them in a virtual space. Viewers have an option to use an android device in a virtual reality setup to spectate the lecture. It aims on improving current "tile" setup which lectures and similar activities take place currently. Application also includes a feed from the presenter's computer to visualize the point of talking, a presentation, or a whiteboard.

Keywords: Virtual Lecture, Virtual Reality, Telepresence.

1 Introduction

In a world, where long distance communication is so prominent, we lose a lot of nuances of face-to-face communication, such as body language and facial expressions. With improvements in connectivity capacity, we can explore ways which require large throughput. Improvement in this field could also hopefully also mean there would be less need of traveling to meeting, conferences and the like. We decided to work on a simplex type of communication, such as a lecture, to get our grip on the problem and test out the different methods available to us. That means, that our application is not symmetric: it has a side of the presenter, which is equipped with a stereo camera setup and only sees the feed from our virtual environment through a monitor. Then there is the viewer side, which has an option to connect either with a traditional display device, or with an android device in a virtual reality box. We chose to develop this application for this type of virtual reality, because we realize, not that many people own, or are willing to buy an expensive VR headset [1]. The downside of using the "cheap" setup, i.e. Google cardboard, is that it lacks the head movement tracking – it only supports rotation. This could however play to our advantage, since if the viewer position is stationary, we do not need to create a model of the presenter, all we need to do is link feeds from each of the cameras from our 2-camera setup to each eye of the viewer.

2 Realization

2.1 Picking the right tools

There are so many possibilities and tools to implement realistic telepresence that is literally making it harder to decide, which tool to use. Let's determine what kind of hardware and software do we need to implement our solution. At first we need to capture a person we want to display in our virtual reality. For this purpose we decided to use Kinect cameras from Microsoft. We are talking plural because for our purposes we will need at least 2 cameras for working solution. That is hardware part for people being displayed in virtual reality, but what about people that will be actually using our application? There is a variety of options to choose for displaying virtual environment scaling from cheap cardboard goggles to expensive virtual reality headsets. We wanted to make this solution available for as much people as possible. With that being said we decided that users of our application will be using combination of their mobile phones attached to cardboard goggles to access application. There are plenty of goggles to choose from and we didn't want to restrict users to buy some specific type [2]. While having hardware on both ends, we will need some software to develop this application. As development environment we agreed to use Unity because of its easy development and compilation for both platforms (Android and iOS). Unity offers many plugins to develop with VR, we decided to use Google Cardboard plugin to simplify development. It comes with native stereoscopic view for each eye and many more. Programming language used for both sending the data with our cameras and also scripting in Unity is C#.

2.2 Infrastructure

2.3 Modeling an environment

When we were creating a virtual reality environment, we had to take in account use case of our application – person who is being recorder – a lecturer and people who are watching his presentation – an audience. This situation fits best for some conference environment (stage) where lecturer is in the front presenting content on the whiteboard while audience is sitting in the back observing both lecturer and the content that is being presented. I want to remind that creating VR environment wasn't primary focus of this application, that's why we sketched and deployed really simple conference scene that you can see on the picture. This room is serving us only for testing purposes during development and final scene will be implemented in more detail.

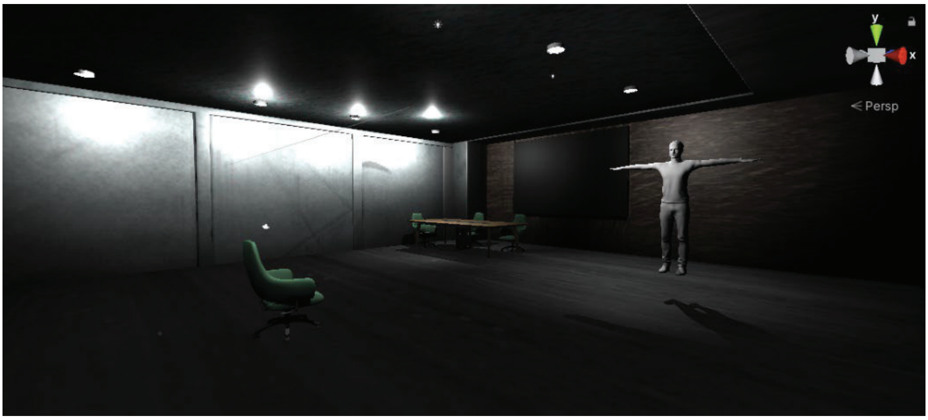


Fig. 1. Virtual conference room

2.4 Capture

We use Azure Kinect DK as our capture device to remove the need of a green screen when recording. the device has a depth camera, which we can use to mask out the background behind the presenter. This is done by utilizing Microsoft Kinect Sensor SDK and Kinect Body Tracking SDK libraries. We wrote a bit of code with these, which converts the depth camera feed into a mask, which we then use to mask out the background from colour camera feed and also add it as an alpha channel component of our video. The software then uses DirectShow libraries to create a virtual webcam. We then use ffmpeg to live encode the video and send it down the chain.

2.5 Display

By now we know how to capture a person, process captured data and send them to the server. But how can we display such data in Unity and even more how to display this person to have feeling that person is in room with us? After some research we found out that google cardboard offers displaying stereoscopic eyes separately. This means we can set different distance between eyes, or what is more important mask certain objects only for one particular eye. You might already guess why would we wanted to do that. We are receiving data from 2 cameras each camera is capturing person from different angle. If we can merge these feeds and display them such as they will overlap each other, but each eye will receive feed from slightly different angle, we will be able to create sense of depth in our brain and the person being displayed will start to feel more realistic for audience.

2.6 Data transfer

The big question about virtual conference using virtual or augmented reality is transferring data in real time. Because we are working with actual 3D models of objects like persons, rooms etc. we need to transfer a huge amount of data in real time to keep concept of live conference. Along with real time transferring huge amount of data in virtual reality come 2 major problems and that is delay and synchronization. Because we are creating multiple video connections along with audio, we need to keep synchronization of these parts.

2.7 Suitable format

As mentioned before, we not only work with multiple video connections at once, we also have to ensure alpha channel is supported. Easiest way would be to transfer raw data. That would eliminate the problem of delay caused from video coding but enormously increase a bandwidth needed for transfer. Regardless of coding delay, a video codec supporting alpha channel would be needed. We are not spoiled for choice with such codecs, because they were not needed that much before. With an expanding trend of virtual reality communication, there is also need of similar technologies as video format supporting alpha channel and transferring 3D objects in real time. Based on the conditions, the video format with alpha channel support would be best for us. If we were not using transparency supporting codec, we would have to transfer another standalone video just as mask for primary one. The video codecs that support transparency are usually high quality, lossless or near lossless. They run at very high bit rates compared to delivery codec like h.264. One such codec is VP8 or VP9, which we transfer by using WebM container. Often, video players have to wait until a WebM video is fully downloaded to play it. This however is not always the case, so with the right choice of video player it is suitable for livestreaming.

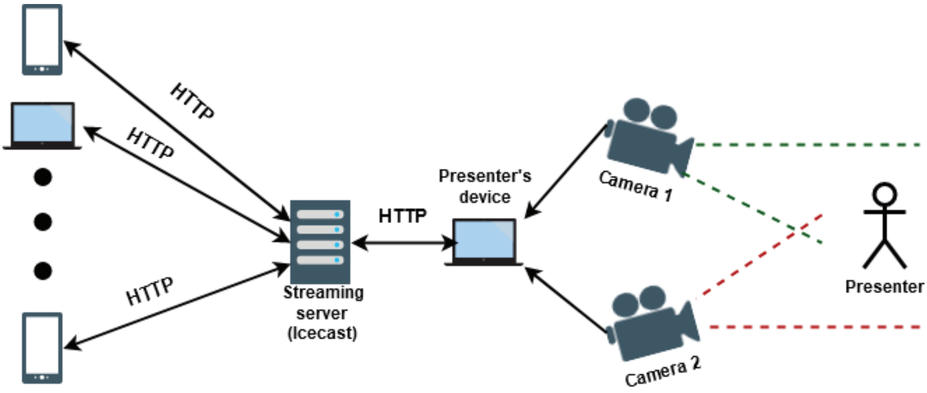
2.8 Delay

Because our goal is to transmit multiple video outputs from the presenter, the delay that affects us most is the delay from combining these parts into one object on the other side of the connection. Different delay visible on the objects representing presenters is something what can affect our live conference experience but it is not affecting it as much as mutual delay of video parts on the same object. If something like one second delay of a person happens, it does not affect us that much in live video conference. Main goal is to deal with delay of multiple videos forming a same object. Modeling a person with data of different timestamps can cause big differences in the arrangement of the body. The delay caused by a codec can be shrinked by using codec on more powerful server than on presenter machine. This however requires sending bigger raw data from presenter to server, so bigger bandwidth would have to be secured.

2.9 Streaming

Transferring data in a virtual conference requires changing data between participants. There are 2 main possibilities to solve communication. First is peer-to-peer method, which is not suitable for our solution because of a bigger number of participants in the video conference. Second method is using centralized video streaming server. Server would create multiple connections with every presenter to obtain all video outputs and also audio. This data would be then multi-casted to every spectator. For restreaming purposes we are using IceCast, originally used only for restreaming Audio, it also allows streaming of WebM container, which is perfect for our application. Transfer protocol used by IceCast is HTTP.

2.10 Infrastructure



To better visualize how this system could look in general, you can take a look at the image above. Main part of whole system is streaming server, which is in our case Icecast. Streaming server is for now used only for video streaming purposes, but will be extended with user and multi-room management features. So Icecast will have to be extended by a program to manage similar things Every participant is connected to this central server, regardless of what type of user it is. The streaming server is location-independent, so the HTTP protocol was chosen for remote communication with it. For demonstration purposes, we do not need to use secure version of this protocol. On the left we can see participants whose primary goal will be to watch the presenting participant, but the two-way arrow indicates a possible two-way communication, which could in the future include a form of text communication in the form of a global chat or the like. On the right is the moderator's device, ie a computer that processes the output from the 2 cameras that capture it and forwards this data to the streaming server.

3 Results

When evaluating results, we are not expecting that we would not spot the difference between a person standing in front of us in real life and a person displayed in our VR headset. What we wanted to achieve is to have at least slight feeling of immersion and depth when looking on a presenter. Sight and perception of every human is different and one can experience headaches from blurring screen while the other one can be overwhelmed of realistic feeling. From our point of view we accomplished task of displaying a person as realistic as we could in our VR headset. Although everyone can spot the difference between the real presenter and the virtual one. There were some significant downsides that are affecting user experience for example the person displayed does not really blend in our virtual room because of masking background of original video captures. We also experienced some laggy behavior because the amount of transmitted data (from two cameras) needed to be live encoded on server side and decoded on the client side.

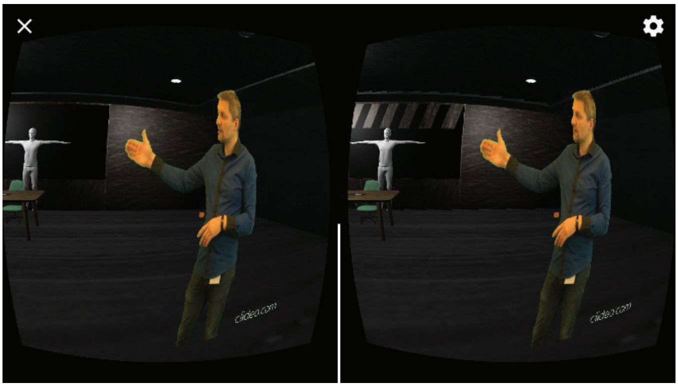


Fig. 3. VR mobile view of presenter with shifted captures

4 Closing thoughts

Since this is first step for complete virtual conference there were also a lot of issues and things to improve. We would like to point out the most significant ones:

- We haven't really used Kinect functionality to transmit whole 3D objects, we transmitted feeds from two cameras instead, which made our implementation really limited
- There is only one way communication from presenter to our audience. Currently our presenter have no feedback to his presentation. Also the presenter can not actually see his co presenters nor audience.
  Audience is static and only move they can make is to rotate their head to display corresponding angle of the scene – which we have currently no use of, the only direction viewer want to look at is the view of the presenter in front of him.

Acknowledgment.

This publication was created thanks to support under the Operational Program Integrated Infrastructure for the project: International Center of Excellence for Research on Intelligent and Secure Information and Communication Technologies and Systems - II. stage, ITMS code: 313021W404, co-financed by the European Regional Development Fund. Research described in the paper was financially supported by the Excellent creative team project VirTel.

References.

1. SHAFER, Daniel M.; CARBONARA, Corey P.; KORPI, Michael F. Factors affecting enjoyment of virtual reality games: a comparison involving consumer-grade virtual reality technology. Games for health journal, 2019, 8.1: 15-23.
2. LEE, Junghyo; KIM, Junghun; CHOI, Jae Young. The adoption of virtual reality devices: The technology acceptance model integrating enjoyment, social interaction, and strength of the social ties. Telematics and Informatics, 2019, 39: 37-48.

This page is intentionally left blank.

Stereoscopic 360 Video Estimation in Camera Nest Midpoints

Richard Paal[1], Jakub Tomáš Král[1] and Radoslav Vargic[1]

[1] FEI STU in Bratislava, Ilkovičova 3, 841 04 Bratislava, Slovakia
xpaalr@stuba.sk

Abstract. In this contribution, we present a system for stereoscopic 360° video frames estimation for camera nest solutions allowing watching the scene from the points located between cameras. Two dominant solutions are proposed - estimation using morphing and estimation using 360° scene stitching. The solutions are evaluated along with other approaches. The results show that the solutions are less disturbing than other evaluated approaches.

Keywords: First Keyword, Second Keyword, Third Keyword.

1 Introduction

Algorithms for aligning images and stitching them into seamless panoramas are among the oldest and most widely used in computer vision. Frame-rate image alignment is used in every camera that has a digital image stabilization feature [1]. Image stitching algorithms also create the high-resolution photo-mosaics used to produce today's digital maps and satellite photos [1]. Image stitching is a process which combines several images with enough overlap to produce one whole image or in other words a panorama [2]. When using image stitching techniques, it is important for images to have identical light exposures to create seamless stitch in the resulting image. To create the most effective results, it's in our best interests to take the images that we are about to use with the same camera, so we do not have to make any extra adjustments to the photos regarding the image sizes for example. Final results of the stitched images are influenced by many factors and its quality is derived directly from the comparison of the input images and the output image, depending on their similarity. As stated in [1] an early example of a widely-used image registration algorithm is the patch-based translational alignment (optical flow) technique developed by Lucas and Kanade in 1981. Variants of this algorithm are used in almost all motion-compensated video compression schemes such as MPEG and H.263. More recent work in this area has addressed the need to compute globally consistent alignments the removal of "ghosts" due to parallax and object movement and dealing with varying exposures. While the older techniques work by directly minimizing pixel-to-pixel dissimilarities, a different class of algorithms has emerged that works by extracting a set of features from an image pair and then matches these features to each other. Feature-based approaches have the advantage of being more robust against scene movement.

One of our techniques used for Image estimation is Morphing, which is a special effect used in animations to create a transformation(morph) from one image to another. The idea is to get a sequence of images, which together with the input images represent change from one image to the other. The simplest method of morphing is that the color of each pixel is interpolated over time from the first image value to the corresponding second image value. More refined methods use warping to achieve more smooth transition between input images. Warping the images distorts the shape of features in the photographs, but the images should not be uniformly warped, since not all the features need to move their location.

In section 2 we present proposed method and its variant. In section 3 we provide evaluation and comparison of the proposed methods. Finally in conclusion we summarize the achieved results and describe out proposed future work on the topic.

1.1 Feature detection algorithms and matching

Speed Up Robust Feature (SURF) technique is fast and robust algorithm for local, similarity invariant representation and comparison of images, it is an approximation of SIFT with faster performance and based on a descriptor and a detector as well [4]. This method approximates the Difference of Gaussian (DoG) with box filters. SURF uses a BLOB detector based on Hessian Matrix, capable of finding points of interest. For orientational assignment are used Haar wavelet responses in horizontal and vertical directions thanks to applying adequate Gaussian weights. Feature description is using wavelet responses as well.

Harris Corner Detector is a corner detection operator that extracts corners and infers features of an image. Corners are points which have become junction of two dominant and different edge directions, where edges are sudden changes in brightness. Detector takes the differential of the corner score directly into account with reference to direction.

Oriented FAST Rotated BRIEF (ORB) is a fusion of the FAST key point detector and BRIEF descriptor after some modifications [8] [9]. In the initial phase to determine the key points the algorithm uses FAST. Then Harris Corner measure is applied to find top points. FAST computes the intensity weighted centroid of the patch with located corner at its center. The direction of the vector from corner point to centroid gives the orientation. The descriptor BRIEF has poor performance with in-plane rotation involved. ORB has a rotation matrix computed through orientation of patch, so BRIEF descriptors can be steered according to orientation.

2 Proposed methods

Image estimation is a process which combines two or more camera angles or in our case images with enough overlap to produce a camera angle in between them, which allows us to watch the scene from different points.

In our attempt at image estimation, we tried using morphing technology on various photos and frames from videos and image stitching. The best results for morphing were created by taking frames from a video and trying our method of morphing them together. This way we created four different videos in MatLab environment with replaced frames. The best results for image stitching were created by using SIFT algorithm and blending the overlapping areas using feathering.

To create the most effective results, it is in our best interest to either use images from one camera or to use cameras with the same settings. This way we can skip the part of making extra adjustments of images regarding the image sizes for example.

2.1 Image estimation by morphing

The morphing has many variants. It is an image processing technique that changes (or morphs) one image into another, using cross dissolving and affine warping. By using warping, we can align the two images before cross dissolving and because of that the scene in the replaced frames is capable of "moving" in a way as in the original video, as opposing to cross-dissolve alone which only creates double exposure of the input frames. Morphing code using affine transformation for warping triangles in MatLab that we use is customized version of the one from this site [7]. The difference is that we have automized the code to the find feature points of input images by using detection algorithms, instead of manually selecting them for each pair of images. Process starts with input of images or frames in our case. The algorithm firstly splits the two images into triangles (with them being of the same number), depending on the number of feature points that we located and matched. The four corners of images are always marked to cover the entire image with triangles. After that comes warping. For morphing, we have evaluated the algorithms SURF, ORB and Harris corner detector to the frames of reference videos as depicted on Fig. 1. We have created three videos, using each of the algorithms individually to pinpoint the location of said feature points of a frame.

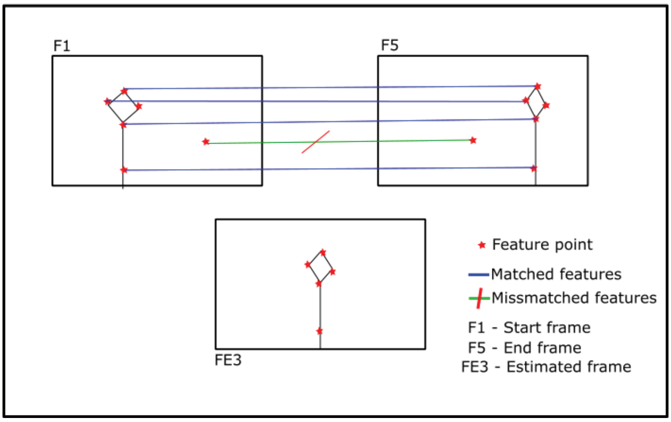


Fig. 1: Example how the feature points are detected by SURF, ORB or Harris Corner detector algorithm in our input frames (F1 and F5), then the features with matches (connected with the blue line) are used to create a new frame through the morphing algorithm (FE3).

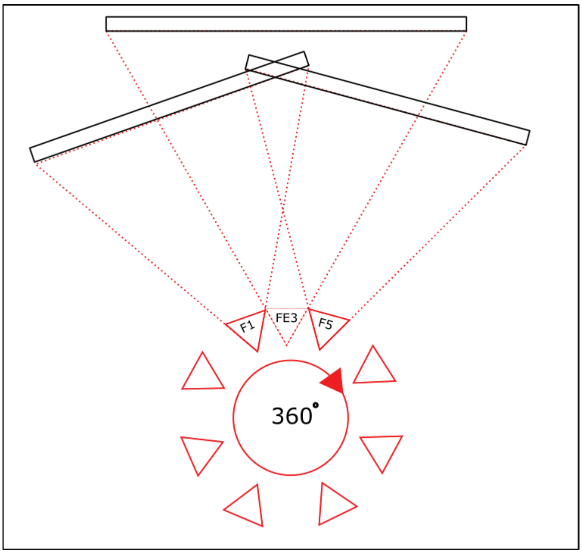


Fig. 2: Example layout of cameras used to capture our video. F1 and F5 are represented by real video frames, that we use as input to create our replacement frames between them (FE3).

Whether it was SURF, ORB or Harris Corner detector, each of these algorithms have in some cases problems creating smooth morph in our video, since the matching algorithm is not unmistakable. To thin out some of the mismatched key points we have created a filter that takes the average distance between key points of two frames and compares this value with others to determine which distances are passable as real matches.

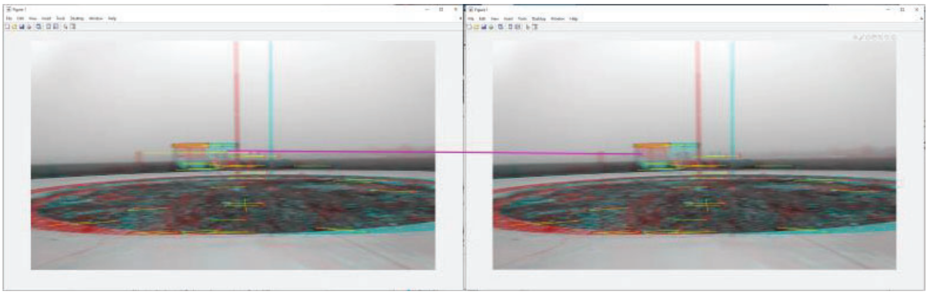


Fig. 3: Example of the change after using the filter. In the first image on the left, we did not use any additional filters and used only functions of MatLab to match features. In the second image on the right, we used filter to take out the matched features

2.2 Image estimation by stitching

The overall method for image estimation by stitching is depicted on Fig. 4. Here, to obtain a correct image alignment, we must first determine a mathematical model for relating pixel coordinates in an image pair. Next, we must estimate what is the correct alignment relating an image pair. Then we must choose a compositing surface onto which we place all of the aligned images that create the final panorama. Lastly, we must use algorithms to try to seamlessly blend the overlapping regions of images even in presence of unwanted effects such as parallax, lens distortion and exposure differences.

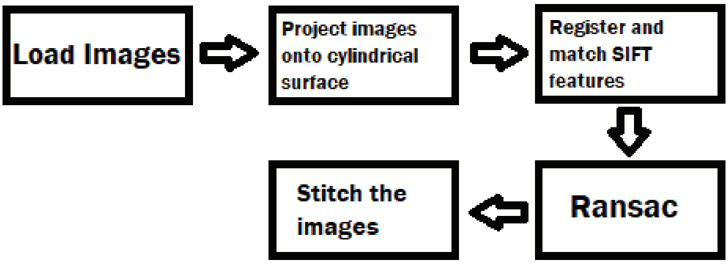

Load Images
Project images onto cylindrical surface
Register and match SIFT features
Ransac
Stitch the images

Fig.4 – Block scheme of image stitching algorithm

Before we can proceed to align images, we need to establish the mathematical relationships that map pixel coordinates from one image to another. Variety of motion models are possible. Once we have chosen a motion model to describe the alignment between a pair of images, we need to suggest some method for estimating the parameters according to which the images will be aligned. One approach is to shift the images relative to each other and check how much the pixels match. Approaches that use pixel-to-pixel matching are called direct methods. The other major approach is to first extract distinctive features from each image, to match these features to establish a global correspondence. A feature is a piece of information which is relevant for solving the computational task related to a certain application. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighbourhood operation or feature detection applied to the image [1]. One type of features is called keypoint. Keypoints are determined based on the algorithm used. Examples of such algorithms were summarized in the section 1.1 Next step in order to achieve alignment is to match the features detected, as shown in Figure 7. Matching is establishing correspondences between two images of the same scene or object. Keypoints between two images coul be matched by identifying their nearest neighbours, which is used in SIFT algorithm [4].

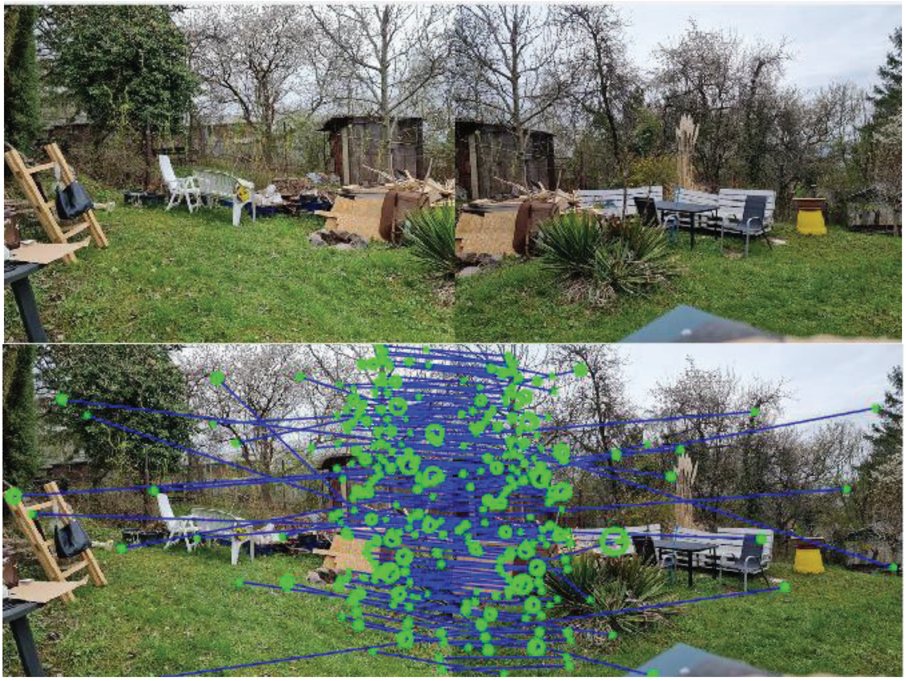


Fig. 5 – Example of feature matching. Upper image - initial image pair. Lower image - image pair with detected and matched SIFT features

Once an initial set of feature matches has been computed, we need to find a set of matches that will produce best possible alignment. One possible approach is to simply compute least squares estimate. Two most widely used solution to this problem are called Random Sample Consensus (RANSAC) and least median of squares (LMS). This step is necessary in order to get rid of incorrect matches that could influence the alignment of the image.

2.3 Composing the final image

Once we have registered the input images and proceeded to align them using above methods now, we need to stitch the images together to produce our final panoramatic image. This involves selecting which pixels will contribute to the final image and how to blend these pixels in order to minimize the visibility of the seams and other artifacts created with this process.

We chose to use cylindrical surface as we are dealing with 360 ◦ panorama and it simplifies the alignment. For blending the overlapping areas, we used is feathering, which is weighted averaging of pixels with a distance map.

3 Evaluation results

For evaluating the results numerically, we chose the Mean Squared Error (MSE) and Structural Similarity (SSIM) metrics. These two methods of objective quality measurement are used on created frames by morphing and cross-dissolve, while using the original frames of the video as reference.

As for the subjective visual evaluation we displayed input images next to result images and tried to detect any artifacts caused by our algorithms.

3.1 Morphing

At the beginning, we took a recording, which was made as a one take video, by rotating camera 360° degrees to the side from a static point. This recording was then used in MatLab, where we took all frames from our video and numbered them from the first to the last. We tried Harris corner detection, ORB and SURF. Example of using SURF is depicted on Fig.6.

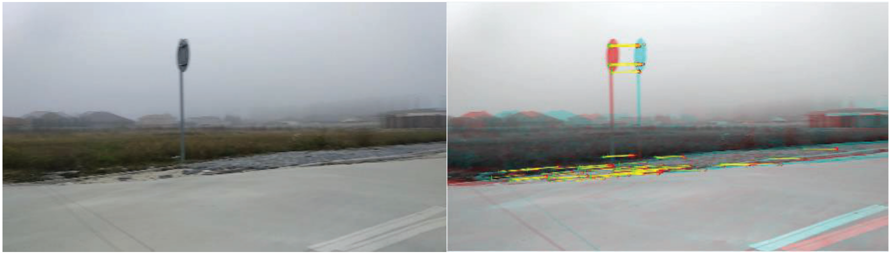


Fig. 6: On the left side we can observe FE3 made by morphing, while using the ORB algorithm for detection of feature points. On the right are actual matched feature points that were left after using our filter for matched feature points.

When comparing the original frames from our recording with the ones made by algorithms there is quite some resemblance. Thanks to the triangle interpolation technique in morphing, we can observe the traffic sign in the same place as the original F3. Since the morphing method is partially based on cross-dissolve, there are parts where the image is doubled.

Method of affine transformation for warping triangles in morphing can create quite pleasing results to the naked eye, however there are some cases, when the morphing method ends up making quite the opposite due to inadequate number of feature points in some parts of the frames or mismatched feature points. This can be seen in the figures below.

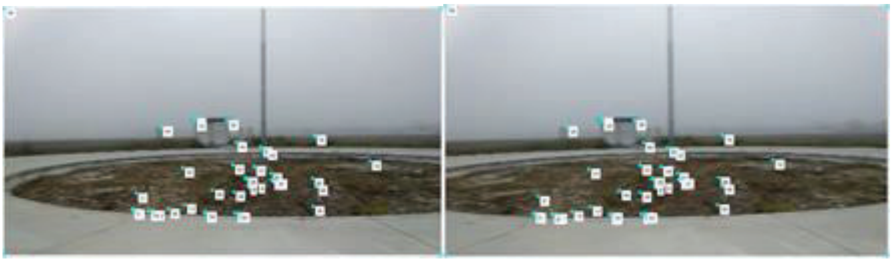


Fig. 7: In this figure we can observe the feature points of the frames used in morphing with ORB algorithm for detection. Start frame F1 on the left and End frame F5 on the right. Most of the feature points are in the middle of the frame, leaving the outer parts of the frame without any features to match.

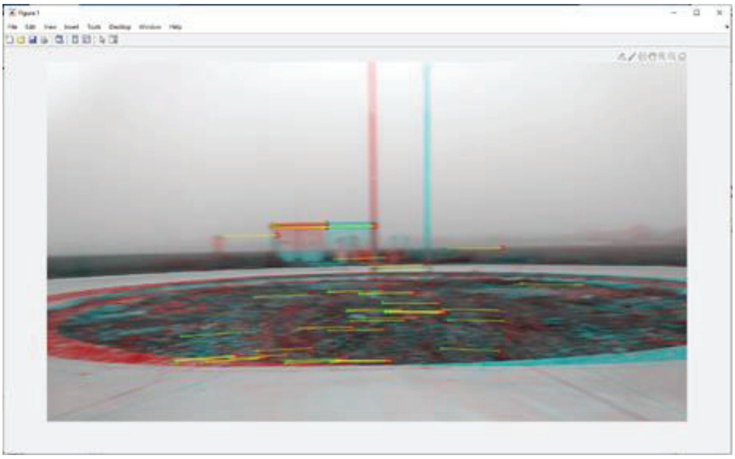


Fig. 8: Here are matched features of the input frames. All matched features have around the same average distance and angle, which should indicate correct matching and a good morphing.

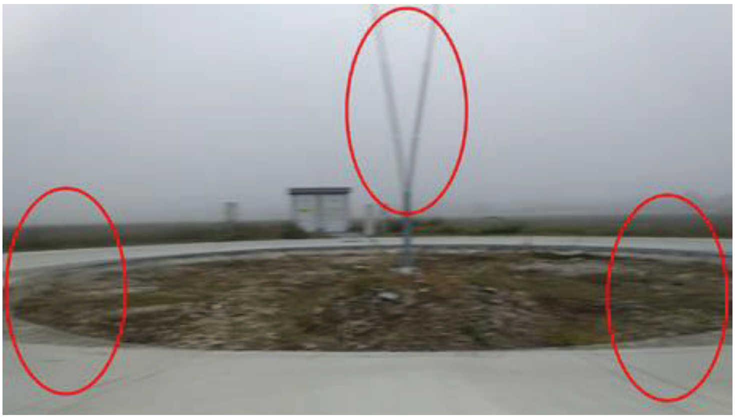


Fig. 9: In this created frame FE_3 is clearly seen how inadequate number of feature points and their layout in input frames creates double-exposure.

After successfully creating a video with replaced frames, we started an evaluation of our results via SSIM and MSE. Through the method of our subjective comparison via vision, we are unable to assume much of a difference between the three methods of morphing with feature detection algorithms. We compared original frames of the video that we replaced and the frames we created through our methods, giving us the following results in Table 1.:

Table 1. Statistics of MSE and SSIM comparison.

	Cross Dissolve	ORB	SURF	Harris Corners
SSIM	0,9163	0,9137470	0,9124	0,9137474
MSE	231,4644	117,02750	120,976	117,02754

3.2 Results using image stitching

At the beginning, we took a set of images by rotating camera 360° from a static point. Pictures in Scene 1 were taken every 24° (Figure 15) and 48° (Figure 16). These data sets were then used in Matlab, where we stitched the images together in order to create a panorama.

The results overall look good, however the image quality had some distortions that were created during its acquisition and processing, as can be seen in below figures. Especially Figure 16 displays these distortions due to the large amount of overlap between the pictures this issue could be alleviated by better by reducing the amount of overlap or better image acquisition as there is considerable scene movement in the Y axis.

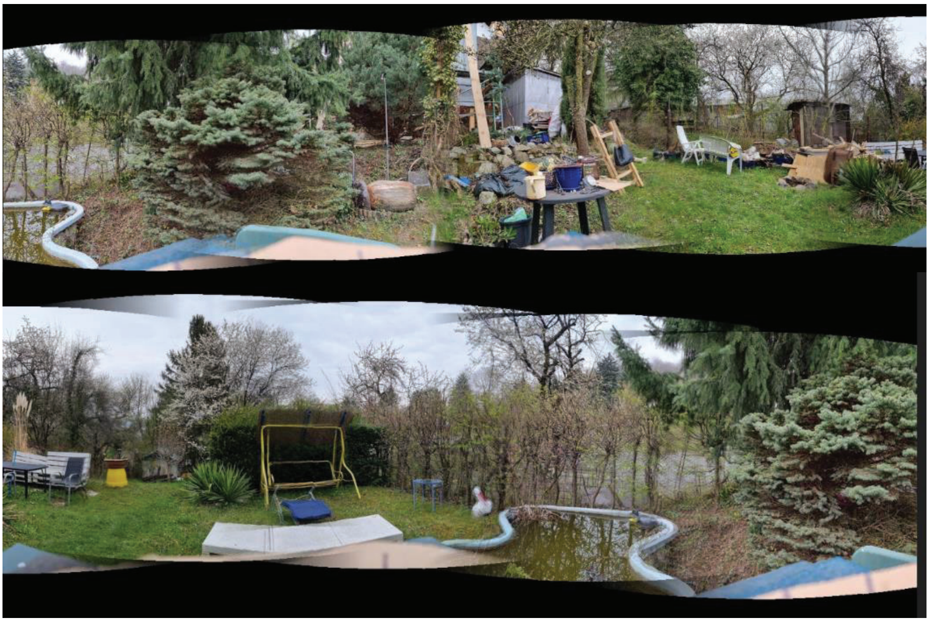


Fig. 10 – Scene 1 resulting panorama (top – left half, bottom – right half)
Pictures in this result were taken every 48° and feathering was used to blend overlapping regions. Blended areas have some artifacts due to scene movement in y axis which causes some ghosting.

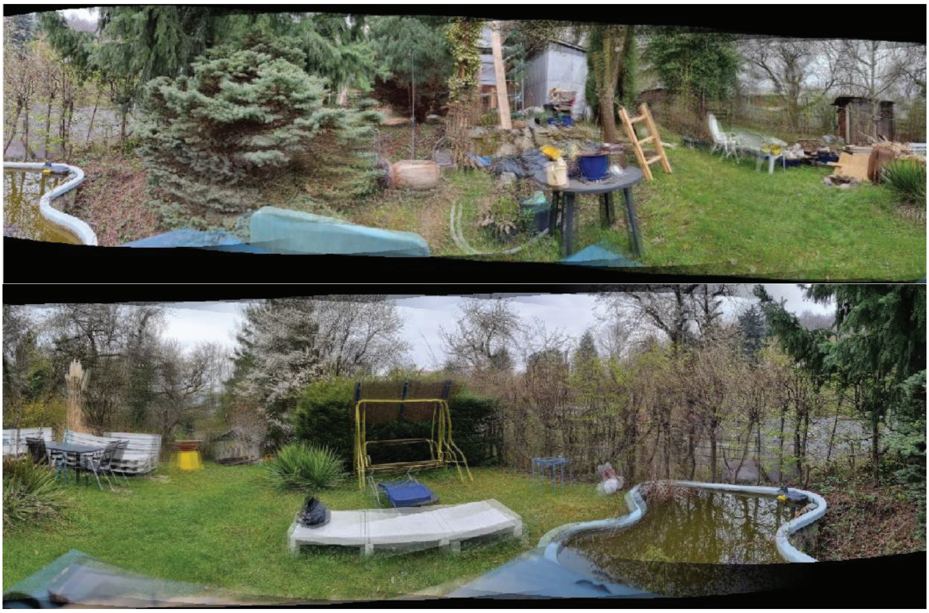


Fig 11. – Scene 1 resulting panorama (top – left half, bottom – right half)
Pictures in this result were taken every 24° and feathering was used to blend overlapping regions. Blended areas have some artifacts due to scene movement in y axis which causes ghosting. Due to the large amount of overlap between the pictures the whole scene has considerable blur and ghosting.

4 Conclusion

In this work we compared the above methods. Both methods produced good results. In both methods the image quality had some distortions that were created during its acquisition and processing.

Image estimation by stitching produced good results, but the method certainly could be improved. As can be seen in resulting images the overlapping regions have considerable amount of blurring and artifacts these issues could be alleviated by better image acquisition or using multi-band blending on the overlapping areas.

Morphing method proved to be quite useful in creating estimated images, which can be seen in comparison with the frames of the video they should represent. The comparison with MSE has shown in the statistics, that the most similar to the original are methods of Morphing with ORB algorithm, but MSE in matlab might not align well with the human perception quality. According to SSIM comparing method, the best results are made via cross dissolve and Morphing with Harris Corner Detector.

Acknowledgment

Research described in the paper was financially supported by the 2020-1-CZ01-KA226-VET-094346 DiT4LL project and the Excellent creative team project VirTel.

References

1. Richard Szeliski, R. (2006), *Image Alignment and Stitching: A Tutorial*, MSR-TR-2004-92

2. Shum, H.-Y. and Szeliski, R. (1997). *Panoramic Image Mosaicing*. Technical Report MSR-TR- 97-23, Microsoft Research.

3. Shum, H.-Y. and Szeliski, R. (2000). Construction of panoramic mosaics with global and local alignment. *International Journal of Computer Vision*, *36(2)*, 101–130. Erratum published July 2002, 48(2):151-152.

4. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91–110.
5. Martin A. Fischler & Robert C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography" (PDF). Comm. ACM. 24 (6): 381
6. Baker, S. and Matthews, I. (2004). Lucas-Kanade 20 years on: A unifying framework: Part1: The quantity approximated, the warp update rule, and the gradient descent approximation. *International Journal of Computer Vision*, *56(3)*, 221–255.
7. Huang, Y. (25. 04 2017). HYPJUDY Github. Dostupné na Internete: https://hypjudy.github.io/:https://hypjudy.github.io/2017/04/25/image-morphing/

8. Edward Rosten, T. D. (2006). Machine Learning for High-Speed Corner Detection. 9th European Conference on Computer Vision (s. 230-433). Graz: Springer.
9. Michael Calonder, V. L. (2010). BRIEF: Binary Robust Independent Elementary Features. 11th European Conference on Computer Vision (s. 778-792). Crete: Springer.

This page is intentionally left blank.

Use of Glass Pyramid in Videoconferencing

Simon Youssef, prof. Ing. Gregor Rozinaj, PhD

Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava 1, Slovakia
simon.youssef99@gmail.com

Abstract. The aim of this work is to create a video conferencing system enriched with the usage of a glass pyramid. This video conferencing system allows video and audio to be streamed between two users in one room. The processed stream is displayed on several video elements that are adapted for an inverted glass pyramid.

Keywords: WebRTC · Glass pyramid · Video conferencing.

1 Introduction

Today, during the Covid-19 pandemic, people have had to limit personal contact and find a way to communicate with each other as much as possible over the Internet. As a result, the usage of video conferencing, which provides real-time video and audio transmission, has increased several times, enabling people to largely replace personal communication. Although today's video conferencing systems allow us to stream video and audio in high quality, they still cannot replace face-to-face contact.

The aim of this work is to create a prototype video conferencing system that is web-oriented and will provide the basic functionality of video conferencing systems, video and audio streaming. This system will modify the video stream to use a glass pyramid, which will project the processed video stream as Pepper's ghost effect.

Pepper's ghost is an illusion technique used in theatre, haunted houses, dark rides and in some magic tricks. Using plate glass, Plexiglas or plastic film and special lighting techniques, it can make objects seem to appear or disappear, to become trans-parent, or to make one object morph into another [1].

In the second chapter we will describe the creation of a signaling server for our video conferencing system. In the next chapter, we will focus on video conferencing system and editing the video. In the fourth chapter, we will describe the creation of glass pyramids for our system. Subsequently, we will focus on testing and finally evaluate the results.

2 Signaling Server

In order to create a video conferencing system, We first needed to create my own signaling server. We have chosen Node.js, which is used for this purpose. Then we have installed the Express framework using npm (Node Packaged Modules) and the Socket.io library, which is used for real-time communication using web sockets.

2.1 Server creating

The first step in creating a signaling server was to include the necessary modules. Subsequently, we created a https server using the createServer() method, where we used the fs module to define the path to the key and certificate of our SSL, as we needed https to transfer stream and get video and audio from the device in a web browser. We set the server to port 9000, because only this port was open to use the web sockets. Then We initialize Socket.io instance and set it to listen on connection. However, for Socket.io to work, it had to be defined on the client's side as well. Subsequently, if the user opened a page on which Socket.io was also defined, the server could already detect it. Using Socket.io, We detected many events on the server side, which were sent by the client, but also vice versa. Each event had its own specific name, thanks to which we were able to distinguish them.

3 Video conferencing system

We used WebRTC technology to create real time peer to peer communication. WebRTC (Web Real-Time Communication) is a technology which enables Web applications and sites to capture and optionally stream audio and / or video media, as well as to exchange arbitrary data between browsers without requiring an intermediary. The set of standards that comprise WebRTC makes it possible to share data and perform teleconferencing peer-to-peer, without requiring that the user install plug-ins or any other third-party software [2]. We used the MediaDevices.getUserMedia() method to get the audio and video stream and we used the RTCPeerConnection interface to create the peer-to-peer.

Our video conferencing system consists of two pages. On the home page users can create rooms or join created rooms. The second page is the video conferencing room itself, in which a video conference takes place between a maximum of two users within one room. The entire video conferencing system is responsive and functional not only on desktop devices but also on tablets and mobile phones.

3.1 Creating and Joining Rooms

Users on home page can create or join a room without any registration. When user creates a room, a room code is created that allows other users to connect to room thanks to it. When connecting, users must be accepted by the founder of the room in order to be able to connect.

3.2 Video conference room

The video conference room consists of 3 parts. The largest part of the screen, up to 80% of the screen width, takes up space for the video stream. The video stream of the communicating user is displayed in the maximum size and is fully responsive. The remaining 20% of the screen is occupied by room and chat information. Users can either view the chat or the room information in which the connected users and the room code are displayed. The last part of the subpage is the taskbar, which occupies 10% of the screen height and is located at the bottom of the page. The taskbar provides several buttons that allow the user to turn the camera and microphone on and off, including a Call End and Settings button.

3.3 Video editing

We stored the obtained video stream in several video elements, thanks to which we created three modes of displaying the video stream of the communicating partner

The first mode shows all the elements of the page where the video of the communicating partner takes up the largest part.

In the second mode, all elements on the page are hidden and only the fullscreen video of the communicating partner is displayed without video clipping.

In the third mode, three videos of the communicating partner will be displayed, this type is intended for the use of a glass pyramid. In this type, as with fullscreen, all page elements except video elements are hidden. Each of the video elements is oriented for one wall of the glass pyramid and displayed in the largest possible screen size. The shape of these videos is adjusted to the shape of an isosceles trapezoid, in order to display the video in the entire wall of the pyramid and not just in the middle.

To switch between these types, we have created buttons that allow users to choose which type they prefer. On mobile devices and tablets, we added the ability to switch types by rotating the device. When you turn the device to landscape mode, the third mode is automatically displayed, when you click on the screen, it switches to full-screen mode, and when you click Close Button, the first type is displayed.

4 Glass Pyramid

The Glass Pyramid is a device that allows you to create multiple Pepper's ghost effects at once. It consists of three or four transparent walls, which have the shape of an isosceles parallelogram. The glass pyramid is most often used with one LCD display that projects multiple images for each wall, or with multiple displays where one display projects an image for one wall. When using a single LCD display for the entire pyramid, the video or image is most often edited into four or three straight triangles or rectangles with a black background to increase the effectiveness of this effect. In order to achieve the Pepper's ghost effect, the angle between the wall and the LCD display must be 45 degrees.

4.1 Creating Glass Pyramids

Since our video conferencing system is available for desktop devices, but also tablets and mobile devices, we have decided to create a glass pyramid for each of these devices.

To make our homemade glass pyramids, we have used the two most commonly used materials for this purpose. One of them is polymethyl methacrylate (plexiglass) with a thickness of 1 mm and the other is a plastic CD cover. Due to the limited size of the CD cover, we have decided to use this material only to make a pyramid for tablets and mobile phones, and we have used polymethyl methacrylate to make a pyramid for a 21-inch computer monitor. You can see the sizes of one side of the pyramid for individual devices in Fig. 1.

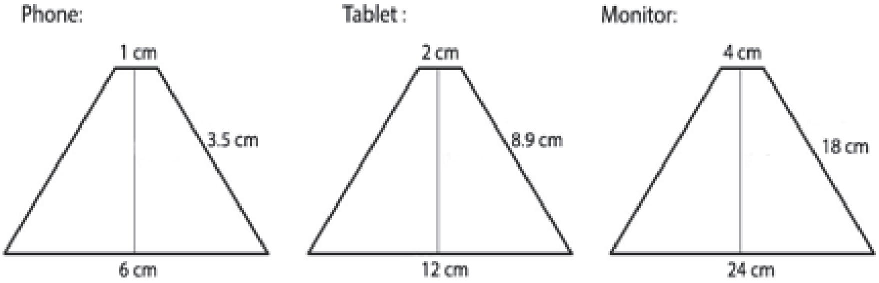


Fig. 1. Dimensions of one side of the glass pyramid for mobile phone, tablet and monitor

5 Results

We created ideal conditions for testing the Pepper's ghost effect. First, we created a virtual camera, in the OBS program, where we set up our USB camera and then we deleted the background using a green background. We did the testing in dark places and set the brightness on the devices to high brightness. You can see the test results in Fig. 2, Fig. 3 and Fig. 4.

We have achieved the best results when displaying on a pyramid made of plexiglass, the video is displayed without blurring the image. You can see a slight blur in the video image in Figure 4 and Figure 3.

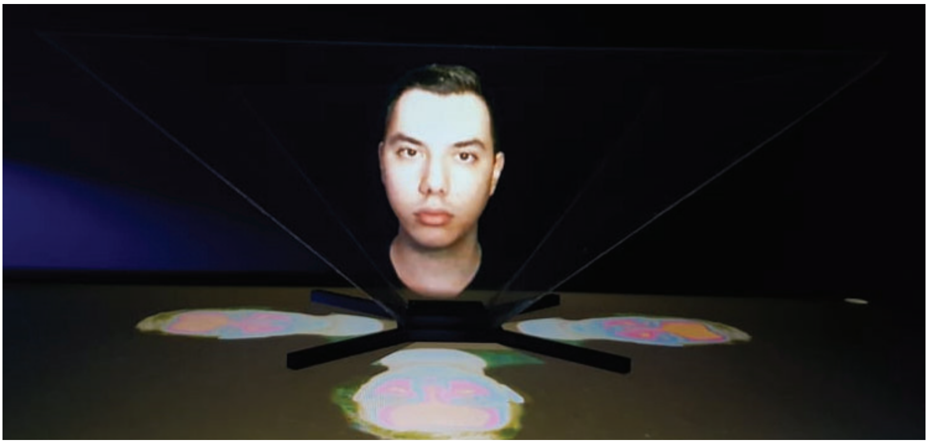


Fig. 2. Glass pyramid tested on a computer monitor

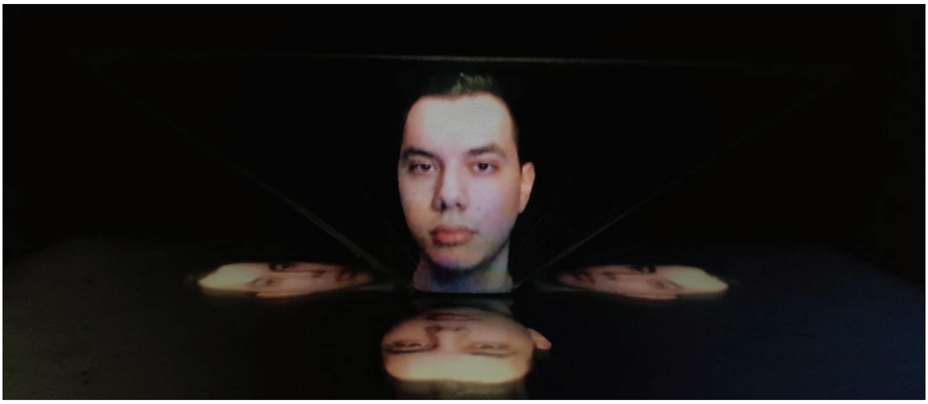


Fig. 3. Glass pyramid tested on a tablet

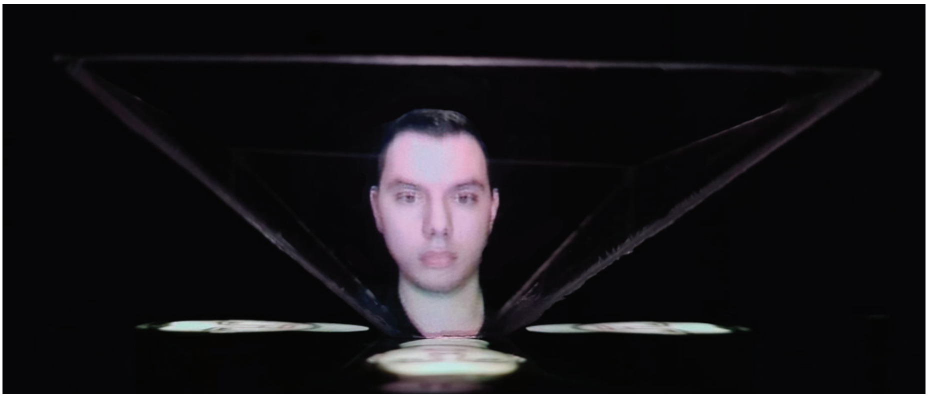


Fig. 4. Glass pyramid tested on a mobile

6 Conclusion

In this project, we have created a functional video conferencing system available via a web browser and accessible via mobile, tablet and desktop devices. We have enriched the system with the use of a glass pyramid. We have also created glass pyramids for this system, on which we have tested the Pepper's ghost effect. We have found out that the best results of displaying Pepper's ghost effect were obtained with a glass pyramid made of plexiglass, tested on a computer monitor. In this pyramid we had a video of high quality, in the other pyramids the image quality was not so good, but sufficient.

References

1. IllusionPedia, https://optical-illusions.fandom.com/wiki/Pepper%27s_Ghost.
2. MDN, https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API .

Technical Solution of a Camera System for 3D Video

Matej Mihálik and Gregor Rozinaj

Slovak university of technology in Bratislava,
Faculty of Electrical engineering and information technology,
Ilkovičova 3, 812 19 Bratislava.

Abstract. The objective of this paper was to design and build solution for stereo camera system, which purpose is observing objects of different sizes in horizontal line. Where the camera system is in remote location, and it sends data to the second 3D displaying capable device. Main data processing is done in the first device with help of minicomputer, that sends data to the second device, which process and visualize the data. In the theoretical part we will analyze image processing and equipment that suits our purpose. The practical part is focused on the designing and implementing commercially available equipment and evaluating results for the possible future improvements.

Keywords: 3D modeling, Virtual reality, Unity.

1 Virtual reality

1.1 Introduction

For those who are not familiar with the term virtual reality, in simple terms it is simulated experience that place user in virtual environment that can be similar or completely different from the real world and often user can interact with it [1].

1.2 Types of VR

In this section we will go through VR types that suits our purpose most, compare their advantages and disadvantages and create possible use cases for each type.

Immersive VR.

It is what people most often connects with VR, using some sort of "head-set" as displaying device with one or multiple screens close to your eyes for immersive feeling of virtual world [2]. But let us start from the ground and build up what are the reasons that it is great for our use case and at the same time it has some downfalls [3]. Firstly in our use case with six camera setup with pairs of two for each section it is most logical because two cameras represents human eyes and can be easily implemented in VR as inputs for displaying screens downfall of this solution is that if we are ok with slow refresh rate in radius of ten pictures per minute or static picture for observing even small low power computers like Raspberry PI can handle task like this but if we want real immersive feeling of world and capture moving object or fast response times in changing directions we need stronger hardware to power our setup with this comes penalty for flexibility, ease of use and price for creating such a device. In this direction future might change thing a lot because every year there are more powerful and cheaper devices on the market. Our target for frame rate is ninety frames pers second which looks like unreachable goal in comparison with our current ten frames per minute but is necessary for comfortable observing of moving objects.

Table 1. Target frame rate for best user experience.

Refresh rate	User experience	Note
>90	Good	This is best what current hardware can do to power high resolution displays and it can be used for extended periods of time.
90 – 60	Moderate	With older hardware we can still achieve great results but for long term use It can create headaches for users.
60 - 30	Bad	At this point it is not recommended to capture moving objects and make fast transitions between changes of FOW.
<30	Static image	It is great for static image and can be powered by small compact computer.

Desktop-based VR.

Involves displaying 3D world on basic screens without using any specialized VR equipment, we can move around selected object in panorama like image. This was also viable solution but requires more image processing and it is also more demanding on raspberry by because it must take picture from all cameras not just the selected pair as in the immersive VR. On the other hand, it is cheaper because it can be displayed on any device and there is no requirement for 3D headset.

2 Technical solution

In this section we will go through thoughts behind creating optimal design, 3D modeling and possible variations, technical solutions, and

2.1 Cameras

Before we start discussing design let’s have a closer look on our selected cameras to understand choices and steps behind our solution. We are using small industrial cameras with parameters shown in table 2.

Table 2. Camera parameters.

Model	ELP-USB500W05G-FD100
Lens Size	1/2.5 inch（4:3）
Max. Resolution	2592(H) X 1944(V)
Dynamic Range	70dB
Connecting Port type	USB2.0 High Speed
Lens Parameter	Size: 1/2.5, Iris: F2.4, Focus: 3mm, FOV(D): 100 Degree
Power supply	USB BUS POWER 4P-2.0mm socket
Operating Voltage	DC5V
Board size	38X38mm（compatible 32X32mm）

Selected camera type allowed us to use it wide warranty but for simplicity and hardware limitations we chose to use six cameras in par of two to represent huma eyes. Next step was to decide on lens distance in general it depends on object that you are observing it can warry from few centimeters to meter in our use case we decided to use four centimeter as it is our hardware limitation, and it works great for our observing distance around half meter to one and half meter in greater or lesser distances picture my become distorted and not suitable for use as panorama or fisheye effect will occur.

2.2 Design

In **Fig. 1** we can see model of holder four our camera system custom designed in 3D modeling software to meet expectations and needs for lens distance, cable management and modularity that can provide observing in variety of distances.

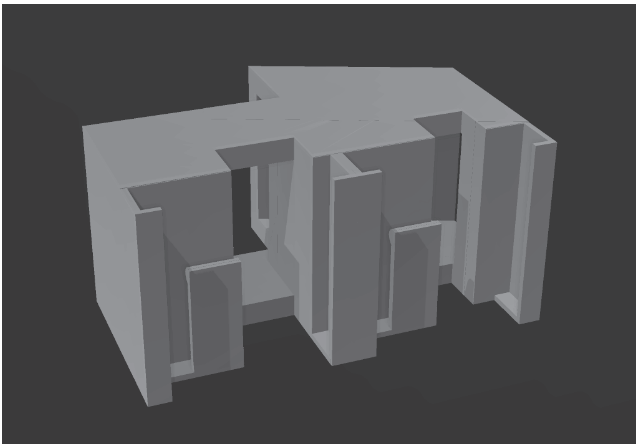


Fig. 1. Camera holder

Next step was to connect and focus all cameras at one point in such way that non other camera can interfere in displaying image and its noticeable change between switching the view. Manufacture provides information about one-hundredth degree horizontal view angle but in real world tests we calculated that view angle is less than ninety-two degrees which gives us option with some image processing to create camera system around the object with ninety-degree view angle and three possible view angels parallel to the object (**Fig. 2**).

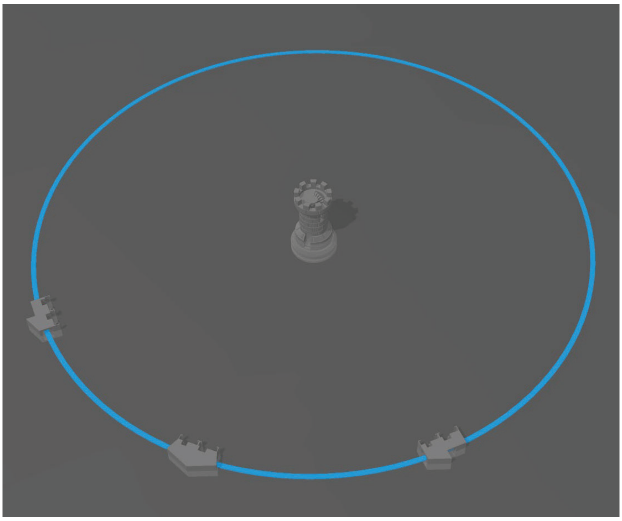


Fig. 2. Camera placement

2.3 Construction

In case of mounting hardware, we choose basic photographic tripod mounting plate for its simplicity and wide variety of options to mount camera system, the mounting plate is connected to the middle part which has built in photographic screw thread. the remaining three pieces are connected by lightweight aluminum U profile with dimensions 30 mm x 25 mm and length of 7500 mm which is strong enough to hold cameras and provide sufficient internal space for cable management that way we can achieve clean look and mor robust and solid construction. And it provides us whit resizable camera system as the camera holders can slide on the profile and can be set to optimal observing distance. All the cables are led to the circular opening of the middle part where they are connected to the USB-Hub and to the raspberry pi. There is also mounting solution for our raspberry pi as it can be attached to the middle part for easy handling.

2.4 Raspberry PI

For connecting cameras, we must use USB-Hubs as the computer has only 4 usb port witch two are usb2 and two are usb3 usb2 throughput speed are sufficient to handle one camera so we split usb 3 ports with hubs into 4 additional usb slots that gave as six ports to operate with. As we mention before raspberry pi is not very powerful and it struggles running all cameras and stable framerates and not crashing at the same time, so we decided to address one camera at time to take picture and switch to another with this solution our frame rate drop significantly but it was necessary to have enough power left for simple image processing and data transfer.

2.5 Image processing

Image processing is not handled by raspberry by because of its limitations but we were able at least pair right photos crop them in requested way and send them to the displaying device in our case next computer.

3 Conclusion

We were able to design and construct functional camera system with commercially available parts. With help of 3d modeling and printing we created compact and perfectly fitting components for our use case. Downside was limited computing power of small computes that cannot provide enough resources to completely process images and we like but this was necessary for us to make it small functional, easy to transfer and setup.

In the future we can see some potential upgrades in cameras itself for better image quality and computer unit for more computing power. In conclusion objective of our paper was fulfilled with some limitations which can be addressed in future upgrades od the system.

Acknowledgement

Research described in the paper was financially supported by project KEGA 015STU-4/2021 and the Excellent creative team project VirTel.

References

1. AZUMA R.T. A survey of augmented reality, Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355–385, August 1997.
2. J.CH. Pomerol. Virtual Reality and Augmented Reality, Myths and Realities, edited by B.Arnaldi, P. Guitton, G. Moreau, ISBN 978-1-78630-105-5 , Online: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119341031.fmatter
3. CUMMINGS, James J., BAILENSON, Jeremy N. How immersive is enough? A meta-analysis of the effect of immersive technology on user presence. Media psychology, 2016, 19.2: 272-309.

Removing Unwanted Objects from Video

Jaroslav Venjarski[1] and Jaroslav Polec[1]

[1] Institute of Multimedia ICT, Slovak University of Technology in Bratislava, Slovakia
jaroslav.venjarsky13@gmail.com

Abstract. This article focuses on removing objects from video and image. Its aim was to study and analyze the methods used in computer vision to mask errors and remove objects from images and video and to design a solution by implementing selected methods. The introduction describes the problems and errors that occur in the transmission and encoding/decoding of visual data, and how those losses affect the resulting transmitted data, and how they set errors/unwanted objects in video and image. Then follows an overview of possible coding/decoding techniques that can prevent the occurrence of errors, or at least the propagation of errors in the transmitted data. This is followed by the described methods for preprocessing the image and subsequently masking and removing unwanted objects /errors from the video or image, which very often occur as a result of problems in video or image transmission, or in encoding/decoding. They are followed by the analytical and practical part of the work. In the practical part, we proposed a solution that is a simulation software, using two methods for error concealment.

Keywords: Video, Image, Objects, Removing.

1 Introduction

In recent decades, we had the opportunity to witness rapid development of technologies that have already been a part of all aspects of our lives. Information technologies in particular are developing extremely fast, which we can perceive thanks to the devices that reached every area of our lives and daily needs. Communication plays an important role in our lives. Informatics and information technologies are not just computers, printers, scanners and software. We can easily include cameras, camcorders and mobile phones as a part of this group. A fast and reliable transmission channel is required to transmit images and videos. Errors often occur in wireless networks, especially in areas with poor signal coverage or in places with too many devices. Error masking is an error control technique capable of mitigating the effects of errors in multimedia using all available decoded data (correctly received and also corrupted data). The term texture is widely used and easy to understand. Image texture analysis is a fundamental problem in image processing and an important area in computer vision [1]. Our intuitive understanding of this phenomenon is so strong that there is no precise definition. Human observers usually describe a given texture according to its qualitative attributes such as smoothness or roughness, which are easily perceptible by their senses [2].

2 Unwanted objects/errors in video and image

Data compression is achieved by removing redundancy, i.e., components that are not necessary for a good representation of the data [4]. If some data contains statistical redundancy, they can be effectively compressed using lossless compression. By decoding the compressed data without loss, we get an exact copy of the original data. However, lossless compression of image and video data provides very little data reduction. Image compression methods use spatial redundancy, video compression uses both spatial and temporal redundancy to obtain the data reduction. In the time domain, there is often a high correlation (similarity) between images that were captured at about the same time [3]. Neighboring frames are usually highly correlated, especially for videos with a high frame rate. In the spatial domain, there is often a high correlation between pixels that are close to each other At the same time, there are three main types of error-resistant techniques:

1. Error-resistant source coding - These techniques deal with the conversion of digital video/image input into an efficient and robust display. More error-resistant coding generally means less compression efficiency, but helps in stopping error propagation.

2. Channel coding and decoding - This channel encoder actually systematically inserts additional bits into the coded bitstream in order to detect errors. Channel coding and decoding is mostly independent of source coding and decoding.

3. Error-resistant decoding and error masking - These techniques minimize the negative impact of transmission errors on the final video/image that is displayed. All decoded data (correctly received and also erroneous data) is available for error concealment. Error-resistant decoding and error concealment include all techniques that allow the decoder to reduce the negative impact of errors by using available corrupted and properly decoded data. The decoder generally goes through three consecutive steps [4]:

1. Error detection - detects if any errors occurred,
2. Error location - detects with the highest possible accuracy where the errors occurred,
3. Error concealment - reduces the negative impact of localized errors.

3 Removing objects from video/image

The most commonly used error-resistant modes for images and videos work with blocks. Therefore, incorrect or missing data will immediately damage entire blocks or block clusters. We use these assumptions to build a problem formulation for masking image and video errors. Mathematically, image and video error concealment is an ill-posed inverse problem since there is no well-defined unique solution. In this chapter, we have made an overview of possible methods for removing objects from video and image. These methods are most commonly used in the field of error masking:

- **Frequency selective methods** - The problem with FSE methods in general is that they always approximate the square area of the lost area, which can often contain parts of the image with unrelated textures.

- **Image Inpainting methods** - Another important group of methods that are not primarily intended to mask the transmission error, but which can be used for this purpose, are the so-called inpainting methods. Inpainting methods can be divided into two basic categories: geometry-based methods and pattern-oriented methods.

- **Inter-frame interpolation** - Time, resp. Inter-frame interpolation is the interpolation of a sequence of frames. The consecutive images in the sequence are very similar to each other.

4 Proposed solution

In this study a simulation software was implemented, in which we tested and compared two methods. The first method was implemented in the MATLAB program, using a function that allows us to apply the widely used and well-known method of spatial patch-based inpainting. We also implemented and tested the second method in MATLAB. It is important to say that the first method is the so-called Intra-frame method, i.e., works within a single frame of video. The second method is the so-called Inter-frame method, and can perform on-frame operations considering the previous frame and/or the next frame. We applied the methods in a video with a large number of synthetic errors, which we created and added to the original video in MATLAB. As a result, we got a video with corrupted frames with unwanted objects that we had later concealed.

4.1 Pre-processing

As part of the preprocessing, it was necessary to segment the video into frames and to save the frames. This was crucial to do so that we can only work with one frame separately or multiple frames of a video on which the unwanted object (error) is located. Of course, we needed to take this step before inserting errors into the video, because we want to obtain undamaged frames.

4.2 Creating a corrupted video (inserting errors)

After pre-processing, we created a new video that contained unwanted objects (errors). This was done by inserting synthetic errors into the original video, resp. we were using the original video as a source of undamaged frames.

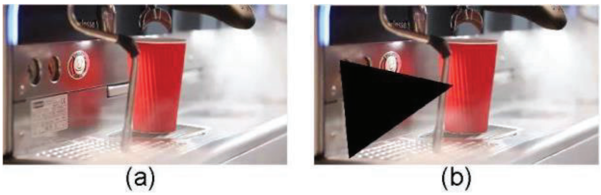


Fig. 1. (a) original frame, (b) frame with inserted unwanted object with triangular shape.

Unwanted objects (errors) that we have inserted into the original video were black objects with randomly generated coordinates across whole frame (objects of different sizes and with random positions in the video frames), and we could choose the shape of unwanted objects. Also, our program was creating binary mask and inverse mask for every unwanted object we inserted. In our program, we could also specify in which frame we wanted to insert unwanted object, or we could choose to automatically insert errors in the whole video, but we had to specify the parameter of error occurrence (parameter f) which is basically a parameter that specifies how often we want the error to occur in our video. For example, if we start with frame No. 2 and our parameter f is 2, then every even frame of the video will be damaged. Lower the parameter f, the error will occur more often.

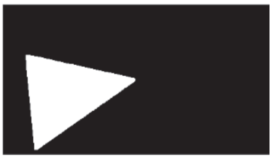


Fig. 2. Mask created for unwanted object inserted in frame.

4.3 Intra-frame method - Spatial patch mixing (SPM)

All error concealment methods in space domain are based on the same idea which says that the pixel values (luma and chroma) within the damaged area can be recovered by a specified combination of the pixels surrounding the damaged area. Our method is using Image Inpainting function in MATLAB to conceal damaged area. Our program automatically was detecting damaged area using mask and concealed damaged area using Image Inpainting. This process was iterated for every damaged frame.

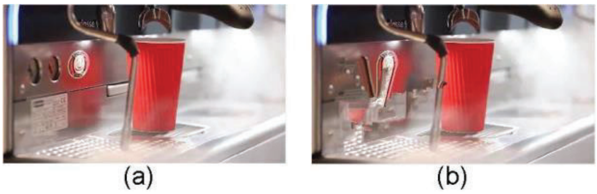


Fig. 3. (a) original frame, (b) frame with concealed error using Intra-frame method.

4.4 Inter-frame method - Inter-frame interpolation (IFI)

Temporal error concealment is one of the most important error concealment techniques. To conceal the errors in the current frame, it utilizes temporal neighbors, that are, the previous frame or the next frame. In our program, we could specify if we wanted to conceal error using only the previous frame (from left), or only the next frame (from right). We also had the option of error concealment using both (previous and next) frames. Error concealment using only one frame (previous or next) is used when we can't use the third option, which is, for example, when we have more damaged frames in a row. In this case, we assumed that we also lost information about motion between frames, and we must replace damaged area with the undamaged area form previous or next frame. When we are using both frames, we are calculating linear combination between the previous and the next frame, and then we are replacing damaged area with the same area from resulting frame.

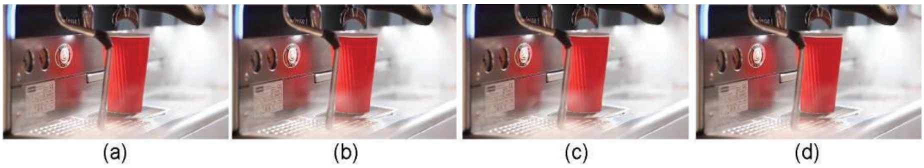


Fig. 4. (a) original frame, (b) IFI using previous frame, (c) IFI using next frame, (d) IFI using both (previous and next) frames.

5 Experiment results

These two presented algorithms (methods) were simulated in computing environment MATLAB using two model video sequences. First one had 375 frames and duration of 6 seconds (Video 1), second video had 1995 frames and duration of 1 minute and 6 seconds (Video 2). Both videos had resolution of 640x480 pixels. In the all the previous figures we used frame No. 175 from Video 2 as an example. For evaluating experiment results we were using MSU VQMT software. This software contains more image and video quality measurement metrics than MATLAB. We were calculating average MSE (Mean Square Error), PSNR (Peak Signal-to-Noise Ratio), VQM (Video Quality Measurement), SSIM (Structural Similarity Index Measure). Also, it is important to mention that these are results from concealing errors from both video sequences when parameter f was 2 and starting frame was No. 2. It means that, in both videos, every even frame contained unwanted object (was damaged), and both methods removed unwanted object from every single one of them.

Table 1. Results of error concealment using SPM (Intra-frame method).

Video No.	MSE	PSNR [dB]	VQM	SSIM	Damaged frames	Elapsed time
Video 1	297	31.294	1.413	0.955	186	27m 93s
Video 2	361	31.721	1.536	0.969	996	2h 51m 28s

Table 2. Results of error concealment using IFI (Inter-frame method).

Video No.	Method	MSE	PSNR [dB]	VQM	SSIM	Damaged frames	Elapsed time
Video 1	From left	189	37.197	0.921	0.983	186	3.271s
	From right	190	37.176	0.922	0.983	186	3.259s
	Both	187	37.241	0.902	0.983	186	3.678s
Video 2	From left	249	35.744	1.078	0.989	996	39.070s
	From right	250	35.736	1.081	0.989	996	38.683s
	Both	245	35.781	1.059	0.989	996	44.296s

6 Conclusion

As we can see from the tables (Table 1. and Table 2.) and from the previous figures, IFI method has achieved better results and is much faster than SPM. SPM method be-longs to the group of methods that are not primarily intended to mask the transmission errors in images and videos. We can see that SPM is much slower than the IFI. This method does not achieve good visual results either (Fig. 3). On the other hand, IFI method is fast and effective and is a good choice for error concealment. However, we must not forget that in some cases we can’t use IFI method, not even from one side (left side – previous frame, right side – next frame) but we can use SPM method always. For example, if every single frame of the video contains unwanted object, we cannot use IFI method because we don’t have any undamaged reference frame, and SPM method works intra-frame so it is usable in every case.

Acknowledgment. This research was supported by the VEGA 1/0440/19 research grant.

References

1. I. Hamouchene, S. Aouat, and H. Lacheheb. Texture segmentation and matching using LBP operator and GLCM matrix. In Intelligent systems for science and information, pages 389–407. Springer International Publish ing, 2014.
2. M. Sonka, V. Hlavac, and R. Boyle. Image Processing, Analysis and Ma chine Vision. Springer US, 1993.
3. I. E. Richardson. The H.264 Advanced Video Compression Standard. John Wiley & Sons, Ltd, 2010.
4. I. E. G. Richardson. H.264 and MPEG-4 Video Compression. JohnWiley & Sons, Ltd, 2003.

Error-rate of Weighted KNN Based on Distance Calculation Method

Samuel Bachratý, Boris Chmeľ and Juraj Kačur

Faculty of electrical engineering and information technology,
Ilkovičova 3, 812 19 Bratislava, Slovakia
bachratysamuel@gmail.com

Abstract. This document contains the theory of Weighted KNN classification algorithm, distance calculation using different methods and serves as documen-tation for code written in Matlab used for calculation of error-rate. Code is given training set of numbers, with defined classes. Based on this set it can predict to which class next sample belongs. By using weighted KNN and multiple distance calculation methods (Euclidean distance, Absolute Difference, Chebyshev dis-tance) we get different distance parameter for each method used, this influences calculated frequency and therefore prediction of class and error-rate, if the fre-quency is equal for both classes, majority criterium of neighbors is used to deter-mine which class sample belongs to. Input parameters: data from each class, num-ber of nearest neighbors and method used for distance calculation. Output is er-ror-rate of code, assuming we know to which class sample belongs to, as well as graph that contains all samples and classes they belong to.

Keywords: Matlab, Weighted KNN, Distance calculation

1 Introduction

Machine learning is focused on building applications that learn from data and improve their accuracy over time. First step toward creating a machine learning model is to select and prepare training data set. Training data set is processed by application to solve the problem application it was designed for. Next step is to choose which algorithm to run on the training data set.

Algorithms find patterns and features in massive amounts of data to make predictions about new data. The better the algorithm the better the predictions will become. Weighted K-Nearest Neighbor (WKNN) is instance-based algorithm that uses classification to estimate how likely a new data point is to be a member of one group or another based on its proximity to other known data points. After training the algorithm we use evaluation subset where we compare the output to results it should have produced. Adjusting distance calculation method might yield a more accurate result. [1]

1.1 How does KNN work?

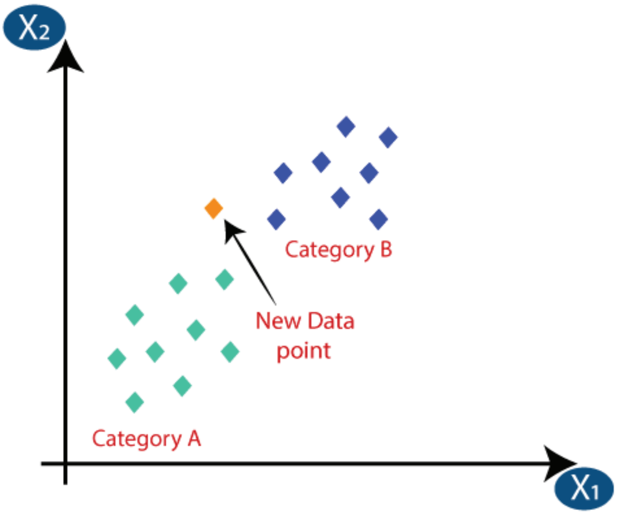


Fig. 1. KNN [2]

KNN is one of the simplest machine learning algorithms to implement based on supervised learning technique. Suppose we have dataset with two categories A and B. If we receive a new data point x1, by using KNN we can predict to which category will this point belong to by selecting the number of nearest neighbors (K) and use them to calculate distance to these points.

After finding K neighbors we count the data points in each category and determine using majority criterion to which category does x1 most likely belong to. (If we use Weighted KNN we calculate weight of each neighbor instead of using majority criterion) [2]

1.2 Advantages and Disadvantages of KNN Algorithm

Pros:

- It is simple to understand and implement
- With large enough data set can be very accurate
- Can be used for both classification and regression

Cons:

- As data set grows speed of algorithm declines
- Selection of optimal number of neighbors can be complex
- Is sensitive to outliers

2 Distance Calculation Method

Selection of nearest neighbors is based on distance calculation method used. Some methods can be more suited then others for specific use cases.

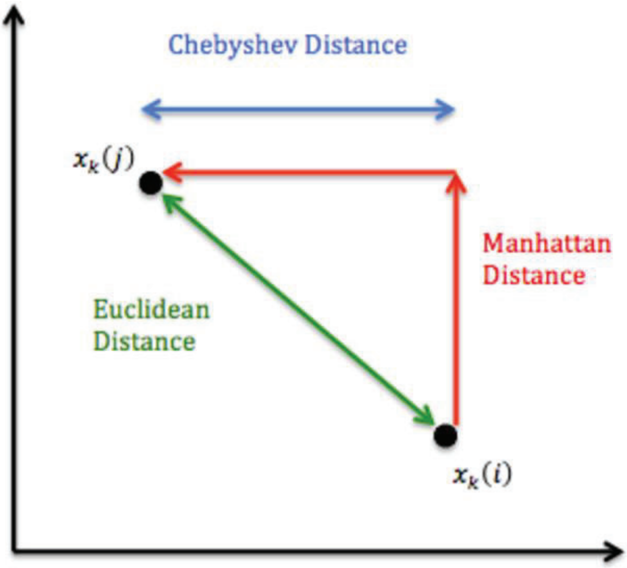


Fig. 2. Distance Methods [3]

Euclidean Distance:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

Manhattan Distance (Absolute Distance):

$$|x_1 - x_2| + |y_1 - y_2| \tag{2}$$

Chebyshev Distance:

$$\max\left(|x_1 - x_2|, |y_1 - y_2|\right) \tag{3}$$

Other popular distance calculation methods in machine learning include: Minowski Distance, Hamming Distance and Cosine Distance.

3 Matlab Implementation

3.1 Input parameters

```
% x,y and c are used as training data set and a1, b1 and
ocak is used as evaluation subset, k is number of nearest
nighbors.

dt = input('select distance method used(default=Euclidean): \n 1=Euclidean \n 2=Manhattan \n 3=Chebyshev \n' );
k = input('select value of k:');

x = readmatrix('500_Person_Gender_Height_Weight_Index','Range','A2:A281');
x = x'

y = readmatrix('500_Person_Gender_Height_Weight_Index','Range','B2:B281');
y = y'

c = readmatrix('500_Person_Gender_Height_Weight_Index.xlsx','Range','C2:C281');
c = c';

a1 = readmatrix('500_Person_Gender_Height_Weight_Index','Range','A282:A301');
a1 = a1'

b1 = readmatrix('500_Person_Gender_Height_Weight_Index','Range','B282:B301');
b1 = b1'

ocak = readmatrix('500_Person_Gender_Height_Weight_Index','Range','C282:C301');
ocak = ocak'

spolu=[];
p=1;

%as source of data we used 500 Person Gender-Height-
Weight-Body Mass Index [4] in xlsx format with classes
defined as male=0 and female=1
```

3.2 Plot of data

```
%Plot of data where class 0 is represented as red and
class 1 as green, p is number of tested data +1 in our
case we tested 20 data points

while p < 21

c = readmatrix('500_Person_Gender_Height_Weight_In-
dex.xlsx','Range','C2:C281');
c = c';

a=a1(p);
b=b1(p);

for i=1:length(x)
    if(c(i)==0)
    plot(x(i),y(i),'r+','linewidth',4);
    else
    plot(x(i),y(i),'g+','linewidth',4)
    end
    hold on;
end

plot(a,b,'ko','linewidth',2);
```

3.3 Distance Calculation

```
distance=[];
distance2=[];
distance3=[];

for i=1:length(x)

    e=sqrt((x(i)-a)^2+(y(i)-b)^2);      % Euclidean
    e2=abs((x(i)-a)) + abs((y(i)-b));   % Manhattan
    e3=max(abs(x(i)-a),abs((y(i)-b)));  % Chebyshev

    distance=[distance e];
    distance2=[distance2 e2];
    distance3=[distance3 e3];

end
```

3.4 WKNN Algorithm

%WKNN algorithm finds nearest neighbors and calculates frequency of classes 0 and 1 using freqX=freqX+(1/distance) formula.

```
if dt==1
distance = distance;
end
if dt==2
distance = distance2;
end
if dt==3
distance = distance3;
end

temp=0;
gemp=0;

for i=1:length(distance)
    for j=1:(length(distance)-i)
        if(distance(j)>distance(j+1))
          temp=distance(j);
          distance(j)=distance(j+1);
          distance(j+1)=temp;
          gemp=c(j);
          c(j)=c(j+1);
          c(j+1)=gemp;
        end
    end
end

classy=[];
freq0=0;
freq1=0;

for i=1:k
    classy=[classy c(i)];
    if(c(i)==0)
      freq0=freq0+(1/distance(i));
    else
      freq1=freq1+(1/distance(i));
    end
end
```

3.5 Output Data

% if freq0 and freq1 is equal majority criterion is used

```
if(freq0==freq1)

output=mode(classy);
spolu= [spolu output]

elseif(freq0>freq1)

    output=0;
    spolu= [spolu output]

else

    output=1;
    spolu= [spolu output]
end

p = p+1;

end
```

% based on ‘spolu’ (calculated classification) and ‘ocak’ (expected classification) matrices we can calculate error rate of calculated classification

```
ocak
error = (spolu~=ocak);
er = (sum(error)/20)*100
```

4 Conclusion

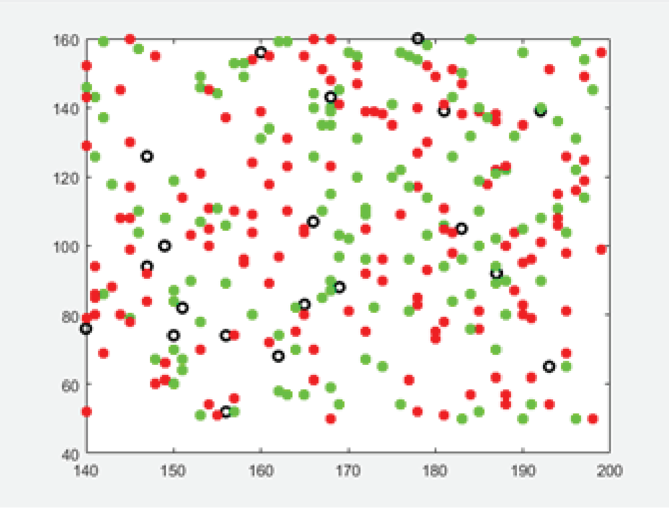


Fig. 3. Graphical representation of data

In this experiment we used our code for predicting gender based on height and weight. We used 280 samples for training and set number of neighbors as K=5. Then we tested next 20 data points and managed to get following results for each distance calculation method:

Euclidean: 70% accuracy
Manhattan: 65% accuracy
Chebyshev: 60% accuracy

It is impossible to predict gender with 100% accuracy purely on weight and height, however we managed to get the most accurate results using Euclidean distance. In some other cases Manhattan or Chebyshev may yield better results.

Acknowledgment

This work was supported by VEGA 1/0440/19 grant.

References

1. https://www.ibm.com/cloud/learn/machine-learning
2. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
3. https://medium.com/@balaka2605/distances-in-machine-learning-289afbce8148
https://www.kaggle.com/yersever/500-person-gender-height-weight-bodymassindex

Optimal Areas of Classification for Classes using Bayes' Classifier

Jakub Jakab, Erik Kaľavský, Roman Kuštor, Juraj Kačur

Faculty of Electrical Engineering and Information Technology STU
Ilkovičova 3, 812 19 Bratislava
jakub@jakab.tech, kalavsky.erik6@gmail.com,
kustorro@gmail.com

Abstract. Naïve Bayes is using a family of algorithms to assign class labels from a pool of labels that are finite. This classifier is a very useful and efficient one and we can see it being used in machine learning and data mining in order to solve multiclass classification. In this paper, we define this classificatory, list its pros and cons and use it in a 2D example. We assume that each feature is independent of others and we disregard any correlations between different features when determining the correct class.

Keywords: Naïve Bayes, class labels, class probability

1 Introduction

In today's world, one encounters a large amount of data at almost every step. What can we imagine under the term data? According to [1], data is information that is stored on a computer and is further processed in some way. Because the internet is widely used nowadays, there is more and more data to process.

In order to avoid chaos and mixing information with each other, various methods have been developed to sort and categorize the information. For example, in the field of information retrieval (Data mining) [2] or in a very evolving machine learning [3].

Very often, the data is divided into different classes according to the observed features and also according to the probability that the event belongs to a particular class based on the input parameters.

In this work we describe the data classification based on the Bayes classifier [4] [5] [6], while generating the input data using a normal distribution with several variables (2D Gaussian normal distribution) [7] [8] [9].

2 Designing the Classification

The classification was designed in the Matlab development software. For correct classification according to Bayes's theorem, we need input data on basis of which the classification takes place.

1. Observation space - As an observation space we chose a 2D grid of size NxM, while the size of a given grid is defined by a set of vectors of mean value [x y], which represent points on a given grid. Subsequently, the grid was defined by the position of the farthest point (1).

$$\text{NxM} = \text{maximum } [x\ y] \tag{1}$$

2. Number of classes - This is the number of classification classes to which a given grid point is assigned based on the classifier.
3. Mean value - This is a necessary data on the basis of which the 2D Gaussian distribution is subsequently determined
4. Covariance matrix - A matrix that determines what a given Gaussian distribution will look like. More in chapter III.
5. Costs - A parameter that determines the individual cost of a given class, which is used by Bayes in classification. This parameter determines the error rate of the classification for a given class.
6. A priori probability - This is the probability that determines the incorrect detection of a given class from another class.
7. Probability density function - this parameter determines that a given observed point has a certain "probability" that it belongs to a particular class. We obtained this probability function for a specific point from the normal multivariate distribution described in Chapter III.

3 Generating input data using normal multi-variable distribution

To generate the probability density function of a given observed point (input data for the Bayes classifier) in individual classes, we used the Gaussian normal distribution with multiple variables. Each class consists of its own normal distribution where the input parameters for a particular class may be different.

To calculate the probability density function at a particular grid point for different classes, it is first necessary to construct a D-dimensional Gaussian distribution for that class. The D-dimensional Gauss is parameterized by a vector of mean μ (2) and a covariance matrix ∑ (3) [9].

$$\mu = (\mu_1, \ldots.\mu_d)^T \tag{2}$$

Equation (2) describes the vector μ which contains the input mean values.

$$\Sigma = (\begin{matrix}\sigma_{11} & \sigma_{1d} \\ \sigma_{d1} & \sigma_{dd}\end{matrix}) \tag{3}$$

Equation (3) describes a covariance matrix where σ represents the variance of a given sample from the mean. This covariance matrix contains only positive numbers on the diagonal.

The resulting probability density function (4) at a given point x is defined by equation (4) [9].

$$p(x|\mu,\ \Sigma) = \frac{1}{(2\pi)^{2/d}|\Sigma|^{1/2}} \exp\,(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) \tag{4}$$

According to equation (4), we then calculated the probability density function for each point in the grid for each class.

Subsequently, we applied the classification based on Bayes' theorem.

4 Bayes classifier

4.1 Pros

- easy and fast class prediction, good performance in multi class prediction
- assuming independence holds, Naïve Bayes performs better and needs less training data compared to other models
- good performance in case of categorical input variables compared to numerical variable(s)

4.2 Cons

- we need to use a smoothing technique (i.e. Laplace transformation) if categorical variable has a category, which was not observed in training data set as the model will assign a 0 probability
- Naïve Bayes is known as a bad estimator so probability outputs should be taken lightly
- impossibility to get an independent set of predictors in real life, which are needed for Naïve Bayes

4.3 Applications of Naive Bayes Algorithms

Real time prediction. Naive Bayes is a fast-learning classifier. That allows it to be used for making predictions in real time.

Multi class prediction. This algorithm is also well known for multi class prediction feature and can predict the probability of multiple classes of a variable.

Text classification. Naive Bayes are mostly used in text classification, thanks to its better results in multi class problems and independence rule, and generally have higher success rate than other algorithms. Therefore there is a wide use of this classificatory in spam filtering and sentiment analysis, which determines if a sentiment is a positive or a negative one.

5 Experiments

First, we dealt with the influence of the covariance matrix on the result of the Gaussian distribution. While changing the parameters σ for deeper observation.
Subsequently, the solution contains several variants. The classification can be based on input predefined parameters, or on random generation of only certain parameters or complete random generation of all input parameters.

5.1 Experiment 1 - Behavior of the output Gaussian distribution based on the change of the parameter σ

In this experiment, we chose the same mean values in each step. Where the mean values are $\mu = [0\ 0]$, and we only changed the covariance matrix.
In Figure 1 it is possible to see Gaussian distributions if the covariance matrix $\sum$ contains values:

$$\sum = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

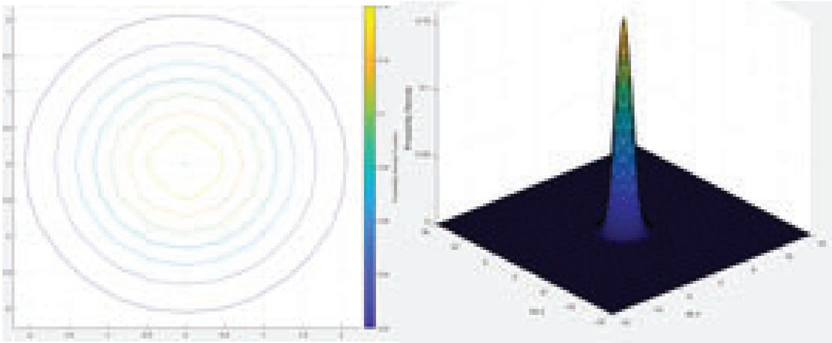


Fig. 1. Gaussian distribution on the left, drawing of the Gaussian distribution on the right

It can be seen from Figure 1 that the distribution is uniform in all directions, and if we increase the parameters σ_{11} and σ_{dd}, the distribution will increase evenly in all directions.

In Figure 2 and Figure 3 it is possible to see Gaussian distributions if the covariance matrix $\sum$ contains values:

$$\sum\nolimits_1 = \begin{pmatrix} 1 & 0 \\ 0 & 6 \end{pmatrix} \sum\nolimits_2 = \begin{pmatrix} 6 & 0 \\ 0 & 1 \end{pmatrix}$$

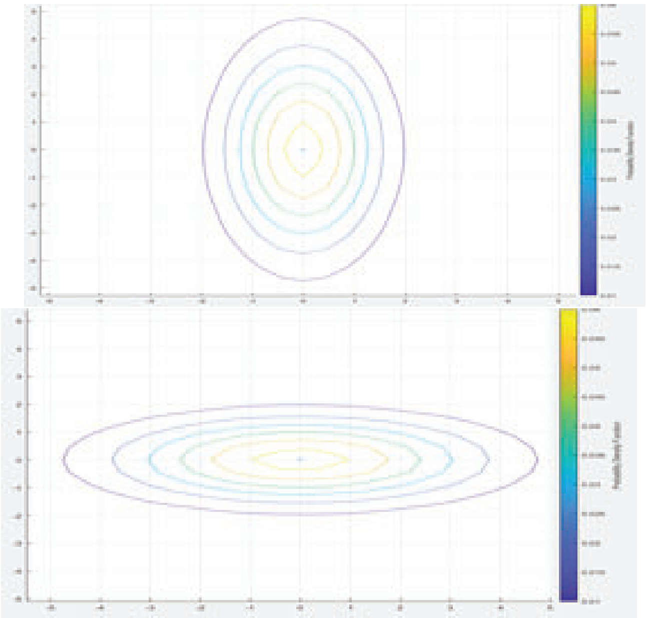


Fig. 2. The Gaussian distribution is shown above if the covariance matrix ∑1 is used. The lower part shows the Gaussian distribution if the covariance matrix ∑2 is used.

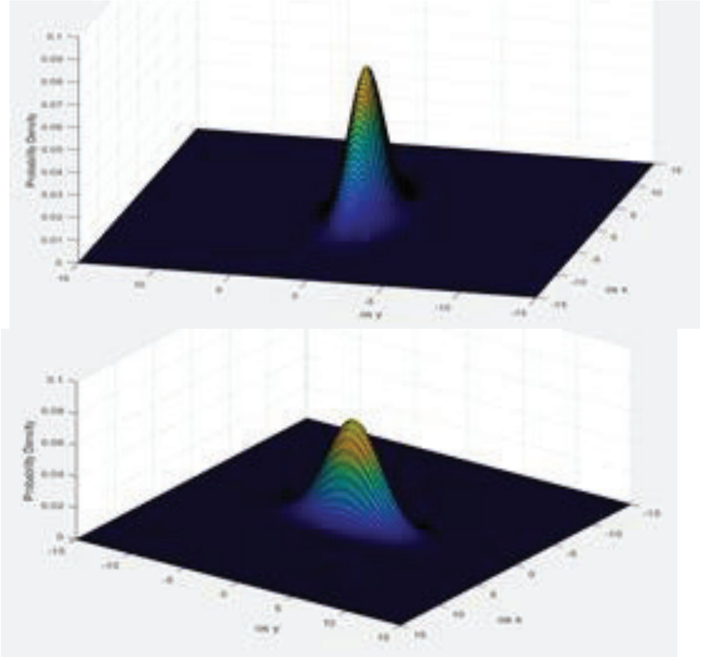


Fig. 3. The Gaussian distribution is shown above if the covariance matrix ∑1 is used. The lower part shows the Gaussian distribution if the covariance matrix ∑2 is used.

From Figure 2 and Figure 3, the difference is visible as the Gaussian distribution changes due to the change of the parameters σ_{11} and σ_{dd}.

If $\sigma_{11} > \sigma_{dd}$ then the Gaussian distribution is in width. If $\sigma_{11} < \sigma_{dd}$ then the Gaussian distribution is in height.

Figure 4 shows the Gaussian distributions if the covariance matrix $\sum$ The figure shows that the Gaussian distribution is inclined from its axis if σ_{1d} and σ_{d1} contain negative values.

$$\sum = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$$

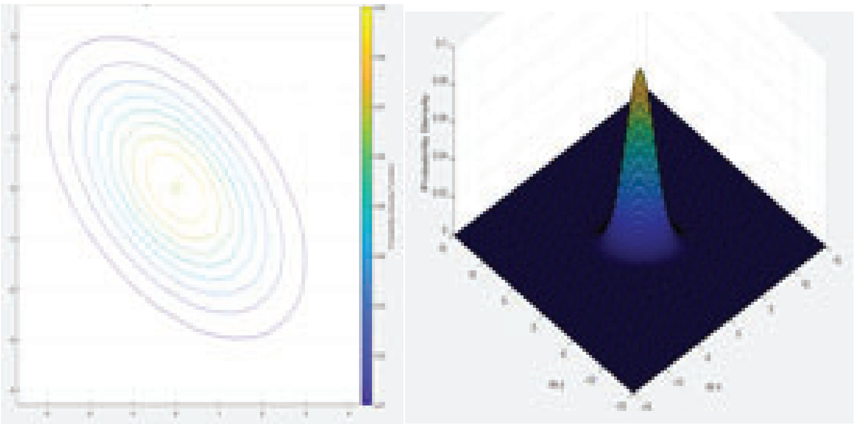


Fig. 4. Left Gaussian distribution, right plot of Gaussian distribution if σ_{1d} and σ_{d1} contain negative values.

5.2 Experiment 2 - Classification based on entered parameters in three classes

Each class T was defined by a separate input vector of mean values:

$$T_1 = [0\ 0],\ T_2 = [-10\ -10],\ T_3 = [8\ 7]$$

Subsequently, the input parameters of the covariance matrix were defined for each class separately:

$$T_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ T_2 = \begin{pmatrix} 5 & 4 \\ 4 & 5 \end{pmatrix},\ T_3 = \begin{pmatrix} 5 & -2 \\ -2 & 5 \end{pmatrix}$$

In Figure 5, it is possible to see the position of the mean values and the Gaussian distribution according to the input parameters.

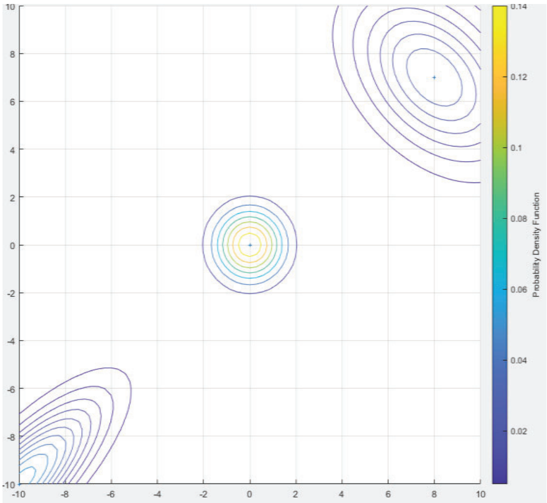


Fig. 5. Mean values and their probability of distribution

Figure 6 is a plot of the given Gaussian classes.

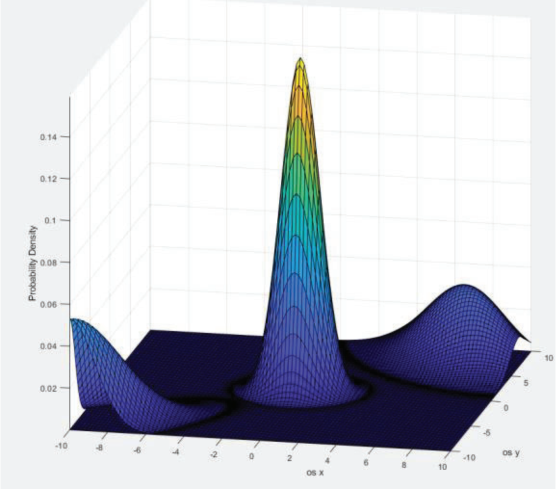


Fig. 6.

Subsequently, we defined a priori probabilities for specific classes. Which are then used in the Bayesian classifier.

$$T_1 \ = 0.25 \ , T_2 = 0.25 \ , T_3 \ = 0.5$$

And we also defined the cost, which determines what is the probability that we will get to the given class T_x from another class T_y

$$T_1 \ (T_1/T_2) = 0.5 \ , T_1 \ \ (T_1/T_3) = 0.33$$

$$T_2\ (T_2/T_1) = 2\ ,\ T_2\ (T_2/T_3) = 0.66$$

$$T_3\ (3/T_1) = 3\ ,\ T_3\ (3/T_2) = 1.5$$

Figure 7 is a plot of the resulting classification based on Bayes' theorem. While it is possible to see errors in the classification in some places, it is due to the costs that are high for the third class, which means that it is highly error-prone.

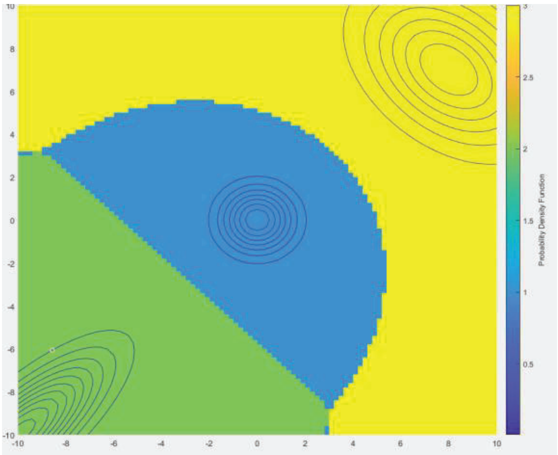


Fig. 7. The result after classification based on Bayes' theorem, where class T1 is shown in blue, class T2 is shown in green and class T3 is shown in yellow.

5.3 Experiment 3 - Classification based on given parameters for Gaussian distribution and random generation of costs and a priori probability

In this experiment, we tested a classification where parameters were manually entered for the five classes of the Gaussian distribution and the subsequent random generation of a priori probabilities for the given class and their costs.

The mean values for the individual classes were given as follows:

$$T_1 = [0\ 0],\ T_2 = [-7\ 7],\ T_3 = [7\ -7]\ ,\ T_4 = [-4\ -4],$$
$$T_5 = [6\ 6],$$

Subsequently, the input parameters of the covariance matrices were defined for each class separately:

$$T_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ T_2 = \begin{pmatrix} 6 & 0 \\ 0 & 1 \end{pmatrix},\ T_3 = \begin{pmatrix} 1 & 0 \\ 0 & 6 \end{pmatrix}$$
$$T_4 = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix},\ T_5 = \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix}$$

Figure 8 shows the subsequent Gaussian distribution based on the input parameters. And in Figure 9 it is possible to see their distribution in the 3D grid.

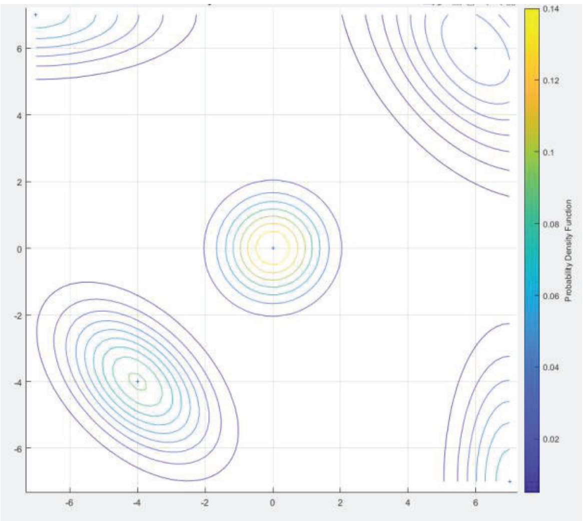


Fig. 8. Rendering of a Gaussian distribution in a grid

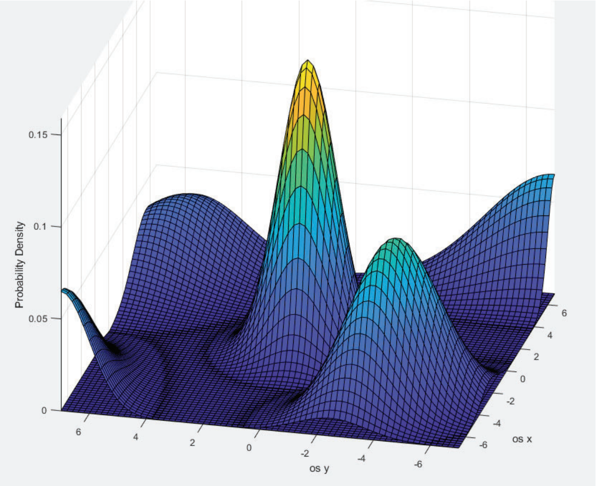


Fig. 9. Render of a Gaussian layout in a 3D grid

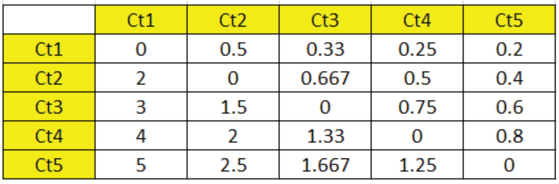
Subsequently, a priori probabilities were randomly generated as follows:

$$T_1 = 0.2697 , T_2 = 0.1257, T_3 = 0.0539$$
$$T_4 = 0.2644, T_5 = 0.2863$$

And also costs were generated, according to the following generation:

$$C_{T1} (C_{T1} / C_{Tx}) = \frac{C_{T1}}{C_{Tx}}, \text{ while } C_{T1} (C_{T1} / C_{T1}) =0.$$

The resulting costs can be seen in Table 1.



	Ct1	Ct2	Ct3	Ct4	Ct5
Ct1	0	0.5	0.33	0.25	0.2
Ct2	2	0	0.667	0.5	0.4
Ct3	3	1.5	0	0.75	0.6
Ct4	4	2	1.33	0	0.8
Ct5	5	2.5	1.667	1.25	0

Table 1. Individual costs for a given class

Figure 10 shows the final classification based on the Bayess qualifier and input parameters.

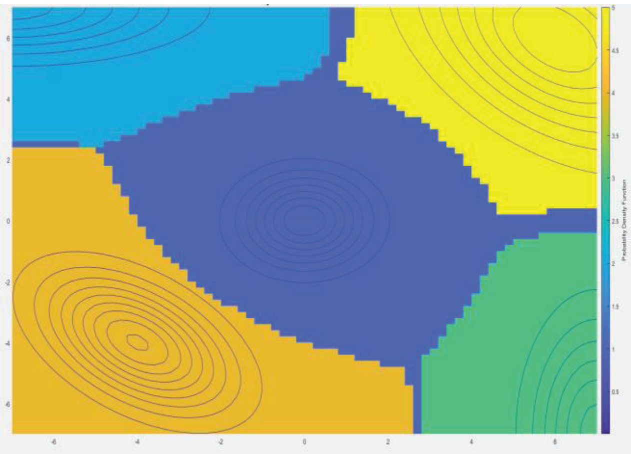


Fig. 10. Result after Bayes classification

5.4 Experiment 4 - Classification based on the generation of all input parameters randomly

In this experiment, we played with the idea of random generation of all input parameters.

This experiment was a demonstration that this is possible but inefficient due to the fact that covariance matrices are generated randomly and do not always meet the conditions for the correct plotting of Gaussian distributions or subsequent classification. The program must be run repeatedly until suitable classification data is generated.

After successfully generating six classes, we received the following results, while the input generated parameters were as follows:

Generated mean values:

$$T_1 = [5\ 5],\ T_2 = [-8\ 4],\ T_3 = [-1\ -6]\ ,\ T_4 = [-8\ 7],$$
$$T_5 = [-7\ -7],\ T_6 = [3\ 8],$$

Generated covariance matrices:

$$T_1 = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}, T_2 = \begin{pmatrix} 10 & -3.5 \\ -3.5 & 10 \end{pmatrix},$$
$$T_3 = \begin{pmatrix} 4 & 0.5 \\ 0.5 & 4 \end{pmatrix}, T_4 = \begin{pmatrix} 6 & -5 \\ -5 & 6 \end{pmatrix},$$
$$T_5 = \begin{pmatrix} 8 & 2.5 \\ 2.5 & 8 \end{pmatrix}, T_6 = \begin{pmatrix} 4 & 0.5 \\ 0.5 & 4 \end{pmatrix}$$

Generated a priori probabilities:

$$T_1 = 0.2488, T_2 = 0.1896, T_3 = 0.1110$$
$$T_4 = 0.3241, T_5 = 0.0986, T_6 = 0.0279$$

The resulting costs can be seen in Table 2.

	Ct1	Ct2	Ct3	Ct4	Ct5	Ct6
Ct1	0	0.5	0.33	0.25	0.2	0.1667
Ct2	2	0	0.667	0.5	0.4	0.333
Ct3	3	1.5	0	0.75	0.6	0.5
Ct4	4	2	1.33	0	0.8	0.667
Ct5	5	2.5	1.667	1.25	0	0.833
Ct6	6	3	2	1.5	1.2	0

Table 2. Individual costs for a given class

Figure 11 and Figure 12 show the Gaussian distribution for the six classes

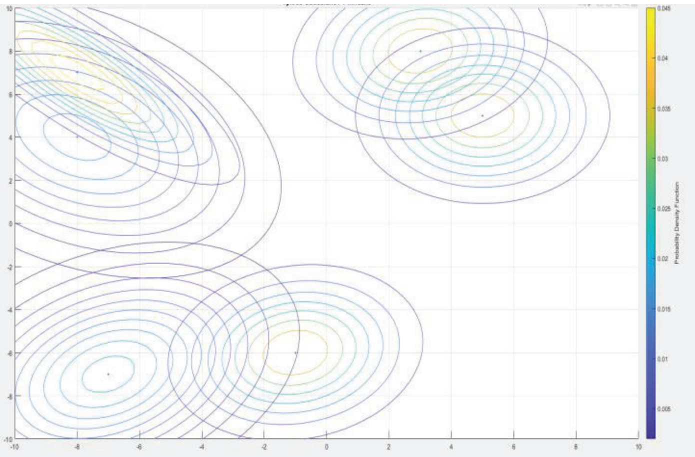


Fig. 11. Gaussian distribution in a grid

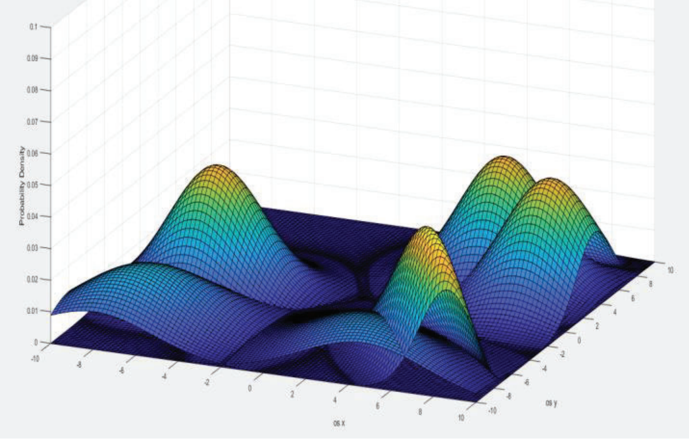


Fig. 12. Gaussian distribution in a 3D grid

Figure 13 shows the final classification based on Bayes' theorem and input values.

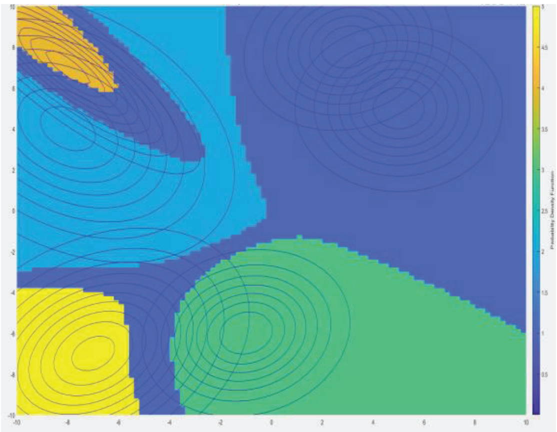


Fig. 13. Resulting classification based on Bayes' theorem

6 Conclusion

In our work, we focused on various methods of generating input data and their subsequent classification based on Bayes' theorem. From the results it is possible to see how the Gaussian normal distribution changes based on the modification of σ. Subsequently, we entered the input data and classified the individual points of the grid. From the results it is possible that the classification is successful but sometimes there are errors. These errors can be optimized by changing the individual cost, or by changing the Gaussian distribution and A priori probabilities for individual classes.

References

1. BUCKLAND, Michael K. Information as thing. Journal of the American Society for information science, 1991, 42.5: 351-360.
2. KESAVARAJ, Gopalan; SUKUMARAN, Sreekumar. A study on classification techniques in data mining. In: 2013 fourth international conference on computing, communications and networking technologies (ICCCNT). IEEE, 2013. p. 1-7.
3. KOTSIANTIS, Sotiris B.; ZAHARAKIS, I.; PINTELAS, P. Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 2007, 160.1: 3-24.
4. RISH, Irina, et al. An empirical study of the naive Bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence. 2001. p. 41-46.
5. KEOGH, Eamonn. Naive bayes classifier. Accessed: Nov, 2006, 5: 2017.
6. BERRAR, Daniel. Bayes' theorem and naive Bayes classifier. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics; Elsevier Science Publisher: Amsterdam, The Netherlands, 2018, 403-412.
7. SHIMODAIRA, Hiroshi; RENALS, Steve. Hidden Markov Models and Gaussian Mixture Models. 2017.
8. SHIMODAIRA, Hiroshi. Classification with gaussians. Edinburgh. Retrieved August, 2015, 5: 2016.
9. Hiroshi Shimodaira “ Gaussians“ , 24 February 2015
   http://www.inf.ed.ac.uk/teaching/courses/inf2b/learnnotes/inf2b-learn-note08-2up.pdf

This page is intentionally left blank.

Linear Discriminant Analysis System using MATLAB

Bc. Ervín Rutšek, Bc. Christopher Viktor Ulm, Bc. Vladimír Vrabec and Juraj Kačur

Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia
xrutsek@stuba.sk, xulm@stuba.sk, xvrabecv@stuba.sk,
public@stuba.sk

Abstract. Linear Discriminant Analysis (LDA) is a technique used for dimensionality reduction problems as a pre-processing step for machine learning and pattern classification applications such as text recognition, face recognition, etc. The aim of this paper is to build a solid foundation for what is LDA, and how LDA works and know how to apply its technique in different applications. Then, in a step by step approach, we will create a tutorial which will be implementing certain equations in the form of a MATLAB code. Which will be used to imple-ment a system that reduces the dimension of the data space based on the input data divided into classes, while preserving the essential classification information between the two classes. Its input parameters will be the number of classes and training vectors for each class. Where as the output of the system will be trans-forming vectors, where if the training vectors are in 2D, they will be represented in a graph with its data. The before and after data will be depicted on a graph and compared.

Keywords: LDA, MATLAB, training vectors, eigenvalues, eigenvectors, class, dimensionality reduction.

1 INTRODUCTION

There are many useful techniques for classifying data. Linear discriminant analysis (LDA) and principal component analysis (PCA) are the two most commonly used techniques for classifying data and reducing dimensions. They are often used as the first benchmarking methods before using more complex and flexible methods. Linear discriminant analysis easily handles the case when the frequencies in the class are unequal and their performances were investigated on randomly generated test data. This method maximizes the ratio of variance between classes to variance within a class in any particular data set and ensures maximum separability. Linear discriminant analysis is for example used to classify data in speech recognition. We decided to implement an algorithm in MATLAB for LDA in the hope of providing a better visualization of multidimensional data and its reduction[1].

2 LDA in 4 steps

Before we start it should be mentioned that LDA assumes normal distributed data, features that should be statistically independent, and same/identical covariance matrices for every used class. However, this exclusively applies for LDA as classifier, but LDA for dimensionality reduction can also work reasonably well even if those assumptions are not met. And even for classification tasks LDA seems to be quite robust to the distribution of the data[1].

2.1 Creating data

We are creating our own data sets by using the MATLAB `function randi()`, and its parameters are the range of possible numbers, the number of dimensions and the number of classes. Using this method, we have created a main matrix of the size of R-rows x (d-Dimensions x N-Classes) as shown below:

$$M = \begin{bmatrix} a_{11} & \cdots & a_{1dxN} \\ \vdots & \ddots & \vdots \\ a_{R1} & \cdots & a_{RdxN} \end{bmatrix} \tag{1}$$

After the creation of our main matrix we will now extract our classes (data sets) with their respective dimensions. After the extraction we can plot them in their original dimension before the dimension reduction as shown on the figure 1 below:

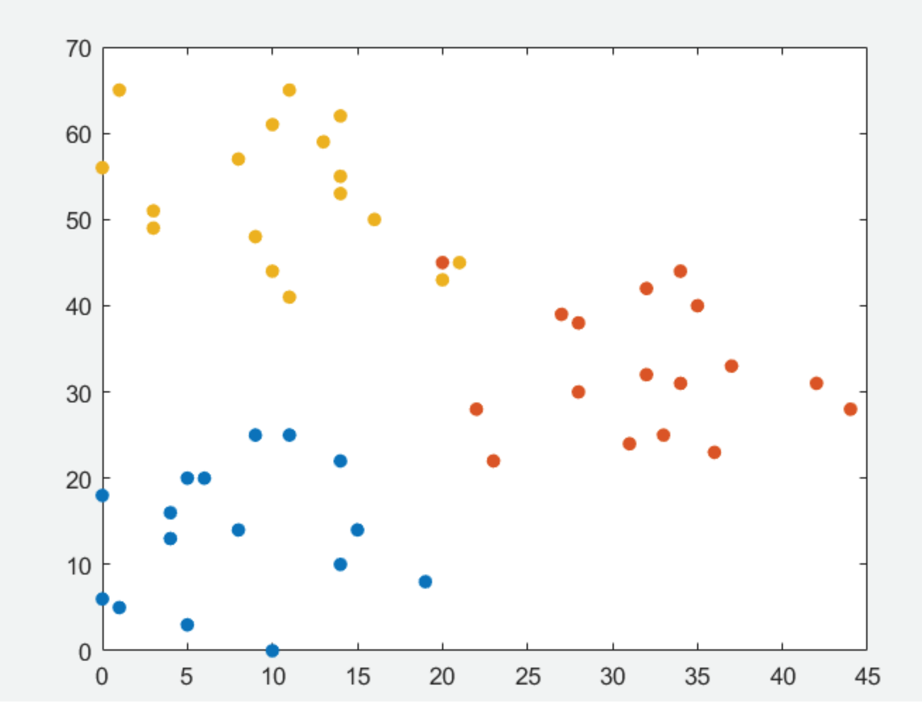


Fig. s1. 3 classes as 2-Dimensional created from the main matrix.

2.2 Computing Scatter Matrices

After our creation of classes or if you have other ways of acquiring of class data sets, the first step in the LDA will be finding two scatter matrices referred to as the "between class" and "within-class" scatter matrices. We will start the within-class scatter matrix $\sum_W$ and is computed by the following equation:

$$\sum_W = \sum_{i=1}^{c} \; S_i \tag{2}$$

Where

$$\sum_i = \sum_{x \in D_i}^{n} \; (x - m_i)(x - m_i)^T \tag{3}$$

and m_i is the mean vector.

$$m_i = \frac{1}{n_i} \sum_{x \in D_i}^{n} \; x_k \tag{4}$$

It is also possible that we could compute the class covariance by adding the scaling factor 1/(N-1) to before mentioned within-class scatter matrix and that would create the following alternative equation:

$$\Sigma_i = \frac{1}{N_i - 1} \sum_{x \in D_i}^{n} \; (x - m_i)(x - m_i)^T \tag{5}$$

and

$$\sum_W = \sum_{i=1}^{c} \; (N_i - 1)\Sigma_i \tag{6}$$

Where N_i is the sample size of its respective class in our case R, which in this case we do not need the term (N_i – 1) because all our classes will have the same sample size. However, this will result in eigenspaces being identical (identical eigenvectors, only the eigenvalues are going the be scaled differently by a constant factor).

Next, we are going to compute the between-class scatter matrix S_B that has the following equation[2]:

$$\sum_B = \sum_{i=1}^{c} \; N_i (m_i - m)(m_i - m)^T \tag{7}$$

where m is considered as overall mean and m_i and N_i are the mean of sample and size of each respective class. S_B can be thought of as the covariance of data set whose members are the mean of vectors of each class. Visual representation of Scatter matrices can be seen on figure 2.

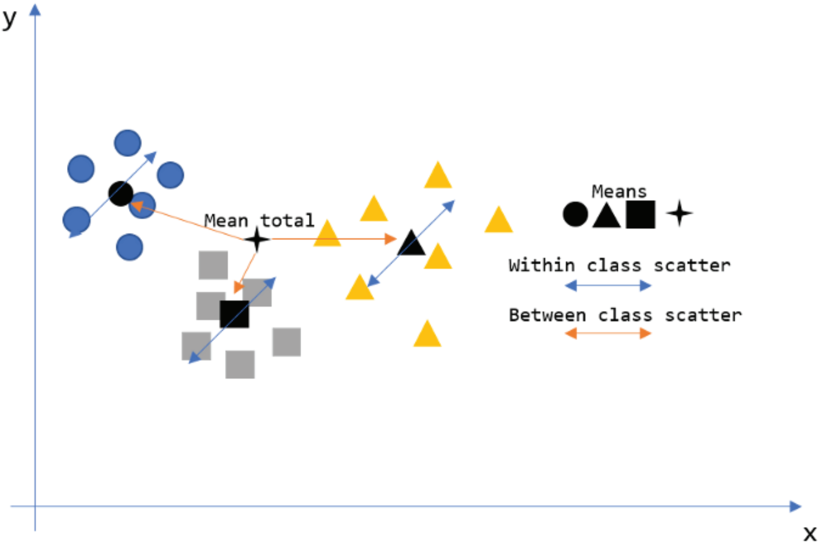


Fig. 2. Visual representation of scatter matrices

Transformation T (LDA) must simultaneously minimize the scattering of data in its classes and subsequently maximize the scattering of classes (their center). That is done through the Fisher's criterion, note: determinant is geometrically oriented volume determined by the matrix. That is the bigger the determinant of the covariance matrix the more is data scattered. Which is why the ratio of determinants dictates the degree of class scattering to data scattering inside the respective classes. Usually, the higher ratio is better for computing. Fisher's criterion equation for determinant:

$$\underset{T}{MAX} = \frac{\det\,(T'S_BT)}{\det\,(T'S_WT)} \tag{8}$$

2.3 Eigenvalues

The solution of eigenvalues is sequentially done through individual directions v_i after:

$$max_{v_i}\frac{\sum_b * v_i'}{\sum_w * v_i'} = d_i \tag{9}$$

Where

$$\sum_b * v_i' = d_i * \sum_w * v_i' \tag{10}$$

General problem of finding eigenvalues and vectors is:

$$\sum_w^{-1} * \sum_b * v_i' = d_i * v_i' \tag{11}$$

Which after calculation of:

$$M * v_i' = d_i * v_i' \tag{12}$$

Equates to:

$$M = \sum_w^{-1} * \sum_b \tag{13}$$

We are looking for non-trivial solution for v_i, which is why det(M-Idi) = 0 with the requirement being $|v_i|=1$. By subtracting eigenvalue M from d_i it is possible to get the ratio of dispersion in the direction of v_i between the classes and dispersion within classes. The bigger the ratio is the better the separation is. Again, the vectors v_i are ordered by its own eigenvalues $d_1 \geq d_2 \geq d_n$. V_i are then arranged as columns of the Transformation matrix. Note that there will only be C – 1 dimension.

$$T = \begin{bmatrix} v_1' & v_2' & \dots & v_{c-1}' \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Now it is time to solve the generalized eigenvalue problem for the equation (13), to obtain the linear discriminants. As it computes a matrix with that contains information about our eigenvectors and their respective eigenvalues. It is necessary to decompose to matrix into eigenvectors and eigenvalues either using MATLAB command `[evec,eval]=eig()` which creates our eigenvectors with eigenvalues and already sorts them into descending order, however it is possible to use other means to achieve these values. Now to interpret the results. Both eigenvectors and eigenvalues are providing us with information about linear transformation being distorted. The eigenvectors are essentially describing the direction of the distortion, and the eigenvalues are the referencing to the scaling factor for eigenvectors which in other words describes the magnitude of the distortion.

When performing the LDA for dimensionality reduction, the respective eigenvectors are important because they are forming the new axes for our new feature subspace. Their associated eigenvalues are also of particular use because they will provide us information about how informative the new axes are. It should be possible to check your calculation using the equation below:

$$Av = \lambda v \tag{14}$$

Where,

$$A = \sum_W^{-1} \sum_B v \tag{15}$$

v=Eigenvector and λ=Eigenvalue

2.4 Choosing linear discriminants for new subspace

Since our goal is to project data into a subspace that is improving on class separability and dimensionality reduction of our feature space, that is where the eigenvectors are creating the axes of our new feature subspace. However, eigenvectors have all the same unit length of one which means they only define the directions of the new axis[2].

We in order to decide which eigenvectors we want to eliminate for our new lower dimensional subspace, it is necessary to use their respective eigenvalues. In other words the eigenvectors with their lowest eigenvalues have the least information about how the data is distributed and those are the ones we want to eliminate. Common application of this is through rank of the eigenvectors from highest to lowest corresponding eigenvalue and choose the top eigenvectors. Through the use of our mentioned Matlab function. After reducing our eigenvector properly we can compute to transform our samples onto the new subspace using the equation:

$$Y_i = C_i \times \text{evec} \tag{16}$$

3 Our MATLAB implementation

3.1 Generating a main matrix for samples and class

Using input parameters we can define our main matrix. The number of rows is defined by the number of classes and the number of columns is defined by number of class*dimensions. MATLAB code: `Z = randi([0, Rozptyl], [R,(N*D)]);` After generating our matrix we than use scattering and indexing to create our classes.

3.2 Computing Scatter matrices

To create our between class and within class scatter matrices, we first need to compute our means, that is done through this code: `P=zeros(1,D*N);`

```
  for i=1:D:(N*D)
     P(:,i:(i+Dp)) = sum(M(:,i:(i+Dp)))/R;
  end
P2=zeros(1,D*N);
  for i=1:D:(N*D)
      P2(:,i:(i+Dp)) = P(:,i:(i+Dp))- Uc;
  end
```

After our computed means we can move to computing the scatter matrix $\sum_B$

```
P3=zeros(D,D);
P4=P2.';
for i=1:D:(N*D)
    O1=P4(i:(i+Dp),:)*P2(:,i:(i+Dp));
    P3=P3+O1;
end
```

```
Eb=P3/N;
```

And the within-class scatter matrix $\sum_W$

```
MT=M.';
Ew=zeros(D,D);
for i=1:D:(N*D)
    for j=1:R
      O2 = ((MT(i:(i+Dp),j))-
(P(:,i:(i+Dp)))).*((M(j,i:(i+Dp)))-(P(:,i:(i+Dp))));
       Ew = Ew +O2;
    end
end
```

3.3 Computing the transform vectors

Reducing our eigenvector and computation transform vector in code:

```
trevec=evec;
trevec(:,D:D:end) = [];
MiM=zeros(R,N*Dp);
t = 1;
for i=1:D:(N*D)
    Tot = (M(:,i:(i+Dp)))*trevec;
    MiM(:,t:(t+(Dp-1))) = Tot;
    t = t + Dp;
end
```

4 Conclusion

After successfully finding out the unusable eigenvectors and using them for our transformation matrix, we were able to achieve dimensionality reduction, class separability and keep its class information intact, where we can reduce it for example from 3 dimensional to 2 dimensional or 2 dimensional to 1 dimensional.

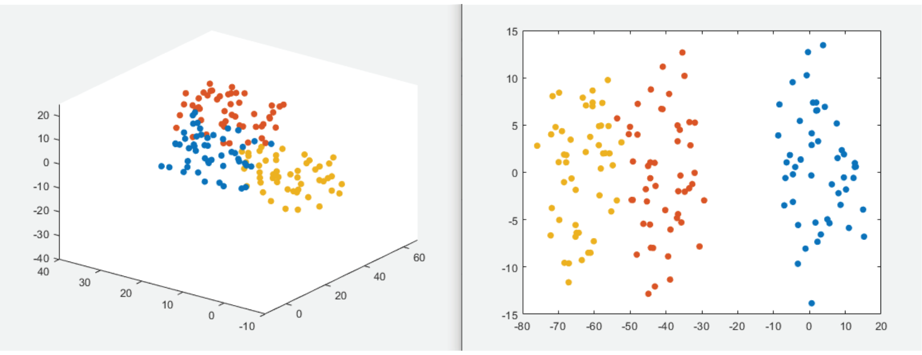


Fig. 3. 3D Class reduction to 2D

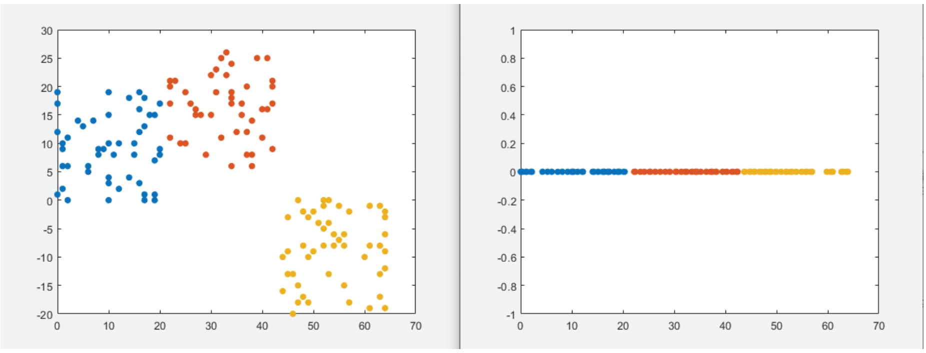


Fig. 4. 2D Class reduction to 1D

Acknowledgment

This work was supported by VEGA 1/0440/19 grant.

References

1. Linear discriminant for machine learning, https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning/, last accessed 2021/05/04.
2. Linear Discriminant Analysis,https://sebastianraschka.com/Articles/2014_python_lda.html, last accessed 2021/05/04.
3. Using Linear discriminant analysis (LDA), https://www.apsl.net/blog/2017/07/18/using-linear-discriminant-analysis-lda-data-explore-step-step/, last accessed 2021/05/04.

Evaluation of Digitial Watermarking on Subjective Speech Quality

Yann Kowalczuk[1][0000−0002−6834−7604] and Jan Holub[1][0000−0003−3350−534X]

Czech Technical University in Prague, 160 00 Prague 6, Czech Republic
cvut@cvut.cz
https://www.cvut.cz/en

Abstract. This paper reports subjective test results of watermarked speech samples.

An open-source software, designed to embed watermarking patterns in audio files, is used to produce a set of samples that satisfies requirements of modern speech-quality subjective assessments. Different level of watermark robustness levels are used, that allow to determine the threshold of detection to human listeners.

Further analysis tries to determine the effects of noise and various disturbances over the perceived quality of watermarked speech.

Finally, a threshold of intelligibility is estimated, to allow further openings on speech compression techniques with watermarking.

The subjective listening tests were conducted following ITU-T P.800 Recommendation, that precisely defines the conditions for subjective testing and their requirements.

Keywords: Speech quality, Watermarking, Speech processing, Transmissions.

1 Introduction

Watermarking of digital mediums is a process that has been on the scene of copyright management since 30 years already. Commercial usage of audio watermarking for public distribution has not picked-up as expected, due to quality and user distribution issues.

However, usage of watermarking in specific markets has revealed some interesting potential, such as communication identification, air traffic control, military and sensible operations requiring security and robustness.

In essence, watermarking is a useful tool, that can be virtually integrated to any digital channel, depending on its contents (audio, video, or text). Thanks to a simple key exchange process, it may be used in trusting non-encrypted transmissions, such as telephone or radio transmission; this principle may include emitter identification, using a dedicated watermarking pattern that is decoded in the receiver.

An extension of this principle in encrypted, compressed audio transmissions is of significant interest in modern cybersecurity. The combination of watermarking signal patterns with compression algorithms may provide significant advantages in the scope of deploying a transparent watermarking solution.

Therefore, our aim is here to discriminate the impact of digital watermarking on speech quality, when modern techniques are used. Further investigation points at finding the limits of speech that remains intelligible, while watermarking robustness is increased in sacrifice of quality.

In order to determine these two limits, selected speech samples will be gradually watermarked with increased robustness, leading in progressive speech quality degradation. A primary threshold of quality will be determined, and retained as a baseline value for common, public voice exchange.

Further effects of noise and environmental disturbances will be added, and the corresponding shift of quality observed and noted.

Finally, a distortion limit will be observed, leading to a retained maxima of watermarking robustness that may be potentially used in speech signals. Practically, operational conditions dictate various scenarii containing heavy noise and variable transmission issues.

Audio quality assessment is regulated by multiple standards. In telecommunication transmission quality tests the ITU-T P800 Recommendation is widely used. It states that to perform a proper subjective audio quality assessment, the subjects must be seated in an anechoic or semi-anechoic listening environment and fully focused on the listening test procedure.

2 Subjective Testing

2.1 Samples preparation

Preselected, available speech samples published by ETSI were chosen, to avoid any potential divergence from self on-site recording and editing. The targeted objective being purely the watermarking impact on the audio quality, potential influencing variables were kept as neutral as possible.

Samples' length and gender voices are normalized at 4 seconds, with male and female speakers alternatively recorded, therefore allowing for favorable post-statistical analysis.

Audiowmark, an open-source software developed by Stefan Westerfeld, was used for embedding watermarks in these samples.

It is a command-line application, that allows to read a chosen sound file, and stores a 128-bit message (defined as a key in the documentation) in the output file.

Audiowmark is using the patchwork algorithm to embed the watermark in the spectrum of the input file. Technically, the audio signal is split in 1024 sample frames. After computation of their FFT coefficients, the frames' amplitude spectrum is altered with pseudo-randomly selected values. Those slight variation of amplitudes serve as a base for the watermark detection post-treatment.

The algorithm used here is inspired by Martin Steinebach, in his thesis ”Digitale Wasserzeichen für Audiodaten”.

As covered in [6], the patchwork method is a dual-channel statistical approach based on a pseudo-random and mathematical process. Patchwork is so called as it is applied to a small segment of the host audio signal which is selected randomly and get added with a specific statistic (for example Gaussian distribution).

The patchwork algorithm is analyzed in details in [2].

2.2 Practical Testing

Several watermarking degrees were inserted into different recording conditions, mainly:

-Original studio recording with clean voice and silent environment. Watermarked strength with values of 10, 30 75, 200 and 650.

-Simulated engine noise from HMMWV tactical transport, with a 3 dB Signal to Noise Ratio. Added watermark with strength of 10 and 30.

-Simulated restaurant / pub noise with a 6 dB Signal to Noise Ratio. Watermark strength set also at 10 and 30.

-Acoustic recording with mild effects such as reverb, and mixed variably with previous noise. Here watermark strength was spread at 30, 100 and 500.

Subjective testing methodology is following ITU-T Recommendation P.800.

12 samples per listening condition were compiled, for a final selection of 16 listening conditions.

A panel of listeners was invited to evaluate the listening quality of those 192 samples, using a dedicated professional voting system.

A total of 8 votes per sample was chosen in order to be representative and obtain exhaustive statistical data.

The 16 specific conditions are described below:

- C01 - Studio recording, clean reference sample.
- C02 - Studio recording, watermarked strength 10.
- C03 - Acoustic recording of original studio sample.
- C04 - Studio recording, watermarked strength 30.

- C05 - Pub noise, added in background of studio sample.
- C06 - Pub noise, watermarked strength 10.
- C07 - Studio recording, watermarked strength 75.
- C08 - HMMWV tactical vehicle noise, added in background of original studio sample, watermarked strength 10.
- C09 - HMMWV tactical vehicle noise, added in background of original studio sample.
- C10 - Pub noise, watermarked strength 30.
- C11 - Studio recording, watermarked strength 200.
- C12 - HMMWV tactical vehicle noise, added in background of original studio sample, watermarked strength 30.
- C13 - Pub noise, recorded acoustically, watermarked strength 30.
- C14 - Studio recording, watermarked strength 650.
- C15 - Studio recording, recorded acoustically, watermarked strength 500.
- C16 - Studio recording, recorded acoustically, watermarked strength 100.

Individual listening and voting were recorded and results analyzed in the next section.

2.3 Testing Results

The distribution of votes is sorted for each condition, and averaged to obtain a comparison basis to the initial, clean studio recording.

The votes are based on a MOS (Mean Opinion Score) scale, as described by ITU-T Recommendation P.800. The scores are described in the next table.

Table 1. MOS Score quality equivalence

MOS score	Corresponding Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Reference speech samples shall match a score located around 5 or 4.5 for narrow-band recordings covering the 300Hz - 3.5kHz spectra, while heavily distorted or unintelligible speech quality scores shall be leveled around 1.

The first graph compares studio recordings with increasing watermarking strengths.

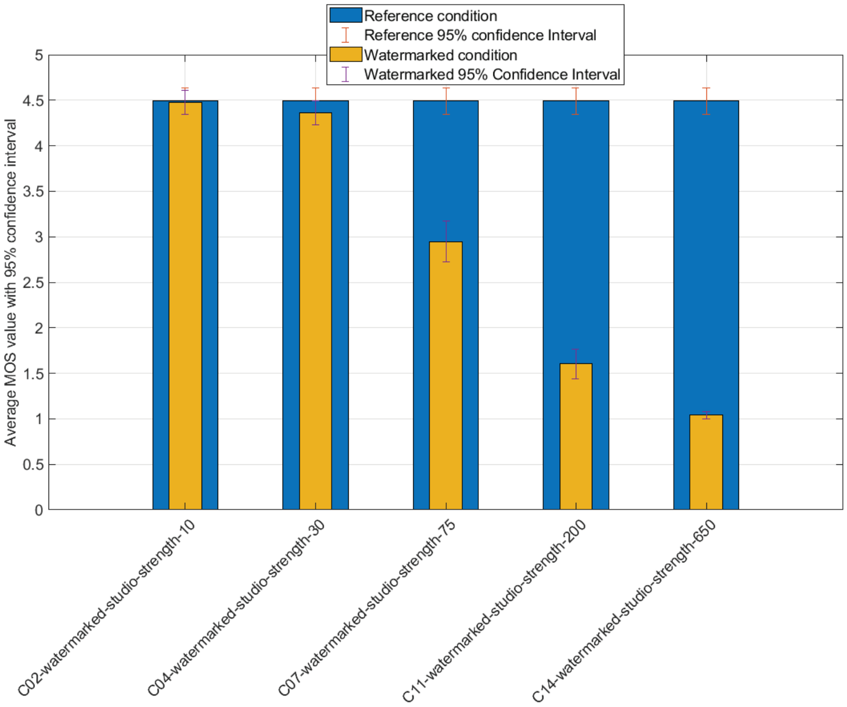


Fig. 1. Results of reference studio condition with increasing watermark strength.

We obtain an even distribution of the scores, with a noticeable degradation of the speech quality happening only at fairly high level of watermarking (strength 75 and above).

All watermarks could be retrieved even at the lowest strength settings.

In order to determine the statistical differences between our results, we use Student's Dependent Groups t-test, single-sided at 95% confidence level. The calculated values are inserted below, following the Figure 2 analysis.

Table 2. t-Test - Results of reference studio condition with increasing watermark strength.

Condition	Reference Condition	t-value
C02	C01	0.104
C04	C01	1.243
C07	C01	*11.402
C11	C01	*25.928
C14	C01	*44.882

Note: statistically important differences (α=0.05 critical value 1.662) are marked with * character.

We remark noticeable differences starting with a watermark strength of 75 and above, while the T-value is increasing proportionally with the strength parameter.

For noise influence visualization, we next plot the scores between the clean studio sample, and the corresponding sample with added noise.

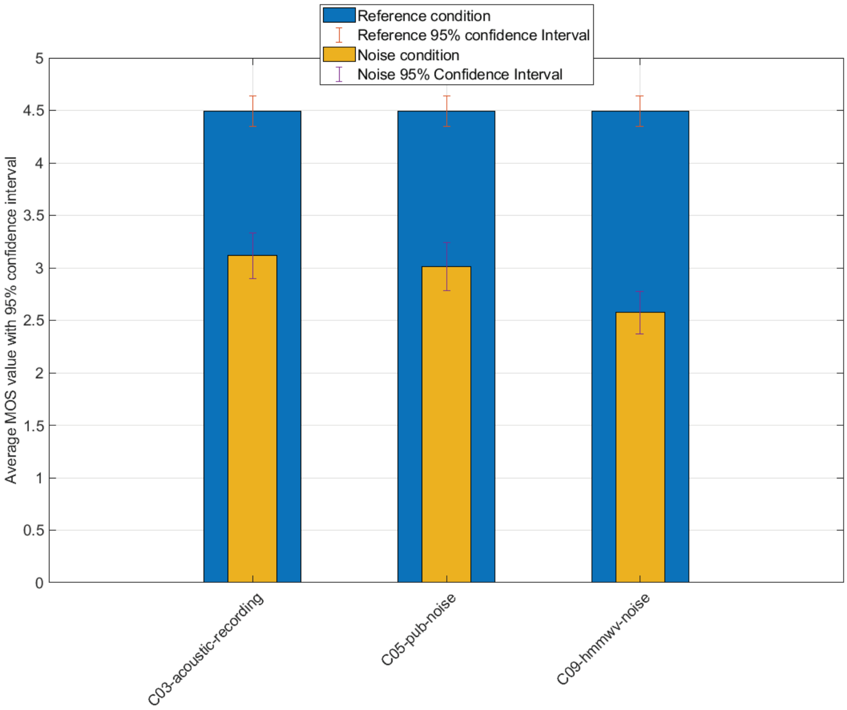


Fig. 2. Results of reference studio condition and noise conditions without watermark.

Noise introduction significantly lowers the initial quality of speech, which was expected. The voters still ticked mostly fair scores, with slight variations depending on the type of disturbance employed.

We introduce the watermarked sample with embedded noise, and notice that we obtain a quite uniform distribution of the scores. This can be interpreted as the low perceptibility of the watermark effect on speech compared to the actual noise of the condition.

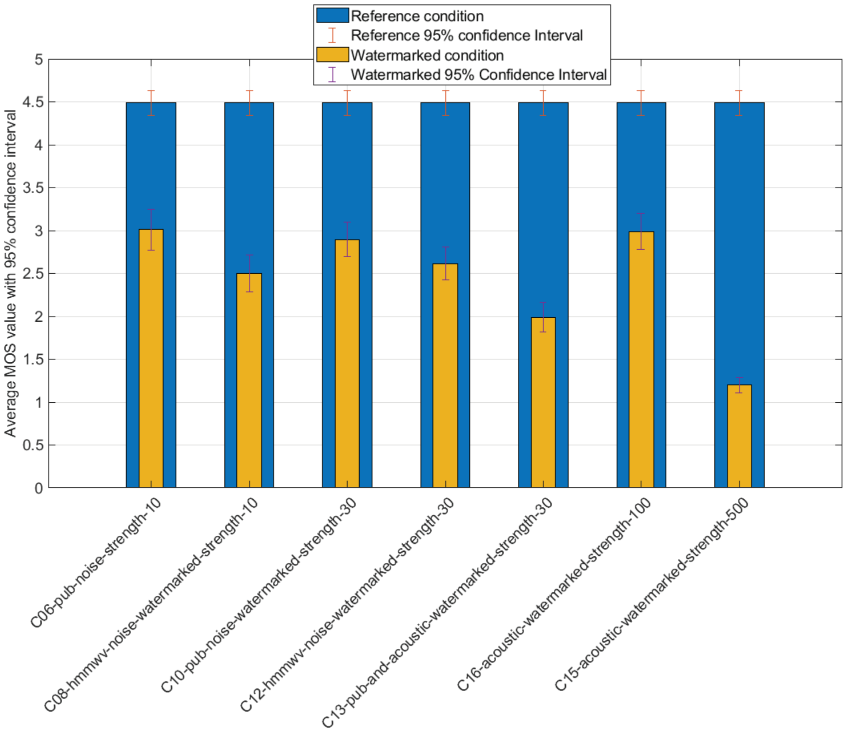


Fig. 3. Results of reference studio condition with noise and increasing watermarking strength.

As we can see, the most severe effect is experienced with acoustic recording. It may be explained by the speech envelope being downgraded, and the corresponding dynamic lowered, adding to the noise effect.

Again, watermarks could be retrieved despite the noise.

This is in-line with the assumption that the slight speech distortion will remain mostly unnoticed in normal or higher noise conditions, making it suitable for transmissions of such nature.

The previous hypothesis may be confirmed by plotting the samples containing the background noise as a reference, and the vote results of the same noisy samples with increasing watermark strength.

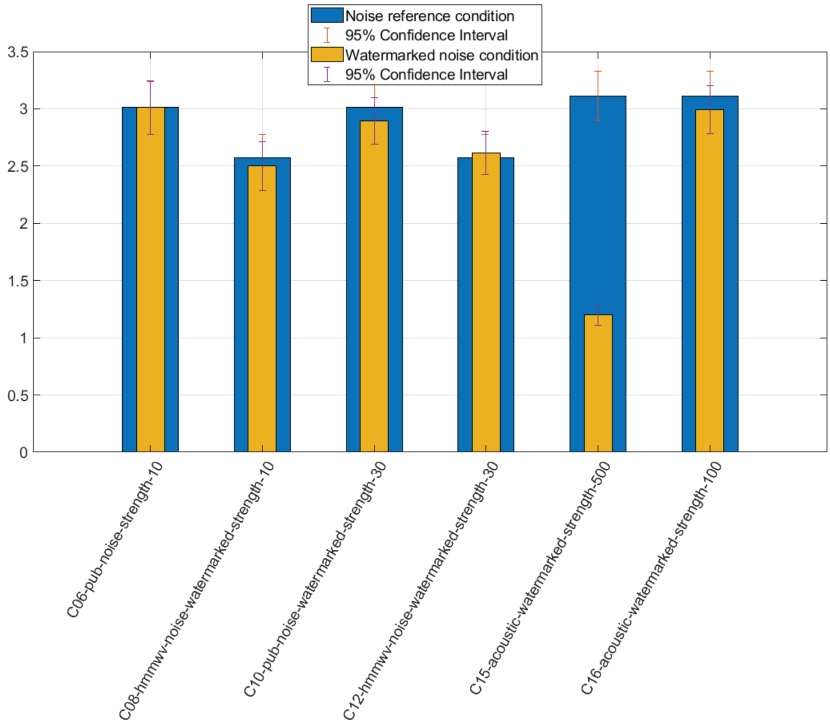


Fig. 4. Results of reference noise condition and increasing watermarking strength.

We see clearly that reasonably watermarked speech in noisy environment remains in an acceptable "fair" quality range. Very high watermarked samples are not of an acceptable quality, however mild-strength values between 30 and 75 lead to a compromise that is promising in specific noise environments.

We repeat the statistical evaluation with Figure 5, in the next table.

Table 3. t-Test Table - Results of reference noise condition and increasing watermarking strength.

Condition	Reference Condition	t-value
C06	C05	0.000
C08	C09	0.488
C10	C05	0.738
C12	C09	0.293
C15	C03	*16.117
C16	C03	0.815

Note: statistically important differences (α=0.05 critical value 1.662) are marked with * character.

Studio samples constitute well-suited candidates for watermarked Signal to Noise Ratio evaluation. The absence of background noise may give an objective, physical way to measure the impact of watermarking on the original speech.

A future study may determine if a degree of correlation exists between the subjective speech quality scores and a given range of SNR values. As multiple parameters come in interaction inside our samples, a precise protocol will need to be defined for such an investigation.

Conclusion

As reviewed in this paper, and through subjective testing with results that confirm a specific potential, watermarked speech samples using patchwork algorithm show that this technique is robust and may be retrieved at low watermark strength, even in noisy conditions.

Our threshold of perceptibility is located at a watermark strength ranging from 50 to 75, while values up to 100 seem to be acceptable in terms of speech quality. At those levels of strength, the watermark may be retrieved in very challenging conditions, and opens the door to further experiments with high distortion and low-bitrate compression testing.

A proper scaling of the watermarked distortion shall be determined by further testing, as SNR values alone may provide guidance, but not a direct representation of the actual modification of the host signal. Selected research introduced a notion of "Signal to Watermark Ratio", that might be a viable metric for further scaling.

Finally, the results obtained under heavy noise conditions reveal that perception of speech remains correct, while introducing modest amount of distortion. Therefore, further experiments may be relevant in situations where voice and noise are directly recorded and injected on the transmission medium with the watermark embedded.

2.4 Authors and Affiliations

The authors are not affiliated to any parties or involved into any conflict of interest related to the publication of this paper.

Acknowledgment

We would like to address our gratitude to Stefan Westerfeld for publishing Audiowmark under GPL license, and his support in the scope of the application usage. All the participants of this experiment are also highly appreciated for their added value in this research. The authors would like to thank Mesaqin.com Ltd for providing the listening room, play-out equipment and its calibration and test subjects throughout this experiment.

References

1. Nedeljko Cvejic, Tapio Seppanen, ”Digital Audio Watermarking Techniques and Technologies: Applications and Benchmarks” IGI Global, pp. 229–247, August 2007.

2. Sascha Zmudzinski, ”Digital Watermarking for Verification of Perception-based Integrity of Audio Data”, Technical University in Darmstadt, April 2017.

3. Ali Al-Haj, ”An imperceptible and robust audio watermarking algorithm”, EURASIP Journal on Audio, Speech, and Music Processing, 2014.

4. Qiuling Wu and Meng Wu, ”A Novel Robust Audio Watermarking Algorithm by Modifying the Average Amplitude in Transform Domain”, MDPI Applied Sciences, May 2018.

5. Akanksha J. N., Diya Dhiraj, Hemanth Reddy, Shikha Tripathi, ”Robust and imperceptible digital speech watermarking”, Proceedings Volume 11719, Twelfth International Conference on Signal Processing Systems, January 2021.

6. Y. D. Chincholkar, S. R. Ganorkarmm ”A Patchwork-Based Audio Watermarking: Review”, International Journal of Scientific & Technology Research Volume 8, Issue 09, September 2019.

Automatic Speech Recognition Training with Automatically Transcribed Data

Slavomír Gereg[1[0000-0003-0283-2997]] and Jozef Juhár[[0000-0002-1596-9258]]

[1,2] Technical University of Košice, Košice, Slovakia
slavomir.gereg@tuke.sk, jozef.juhar@tuke.sk

Abstract: The main theme of this article is results improvement of automatic speech recognition systems based on Kaldi toolkit. This improvement is achieved by training with automatically rewritten data. The article describes method designed for this purpose and its experimental results in training process of system Kaldi. Brief introduction to automatic speech recognition systems is in the first part of article. Also, the basic scheme of Kaldi toolkit is described. The main part of this article is experimental evaluation of designed method for results improvement of ASR systems. Designed method is simple and easy applicable on any speech recognition system because its main part consists of enhancement of training database of ASR system.

Keywords: automatic speech recognition, automatically rewritten data, Kaldi toolkit, training database

1 Introduction

In general, automatic speech recognition systems have the same function as human ear and brain. Simplified, automatic speech recognition systems rewrite speech acoustic signals to their corresponding text forms. Considering that speech acoustic signals are unstable, non-stationary random processes with various instability sources, automatic speech recognition is complicated process [1].

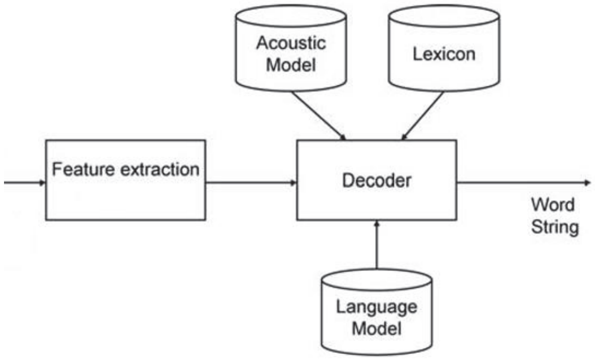


Fig. 1 Basic block scheme of automatic speech recognition system

Almost all speech automatic systems have similar main structure, containing feature extraction, acoustic modeling, language modeling and decoding.

1.1 Feature extraction

Main goal of feature extraction is to extract sequence of acoustic observations that obtain all information for recognition part. After this phase, signals are represented in form of feature vectors (for example MFCC, PLP, …) [2].

1.2 Acoustic modeling

Acoustic modeling phase is focus on determination of degree of similarity between acoustic model assigned to text word and input acoustic representation [2].

1.3 Language modeling

Language modeling phase helps with suppressing those sequences of words, that have minimal frequency in training dataset [2].

1.4 Decoding

This phase is the most crucial part of automatic speech recognition systems. Main goal of this phase is to find best sequence of words that can correspond with acoustic signal input represented by extracted features. This phase use acoustic and language models for recognition process [2], [3].

2 Modern automatic speech recognition approaches

The most widely used methods for automatic speech recognition are currently statistical methods. These methods use statistical decision making. This kind of decision making is based on statistical acoustic and language models. For application of automatic speech recognition in unlimited domain we need a large amount of acoustic and language data for parameter estimation. Thus, one of the key elements of modern automatic speech recognition systems is the ability to process large amount of acoustic and language data [4], [5].

Words can be modeled as whole word, so we need to find one result model for each of words that we are trying to recognize. We can also use smaller units as phonemes for word modeling. In this case we can find result model by concatenation of phoneme models. Further we can associate word model to language model. This model provides additional information about the consecutive word occurrence statistics [4], [5].

We can divide statistical methods into three main groups by the used technology. HMM (Hidden Markov Model), ANN (Artificial Neural Network) and hybrid methods - using both HMM and ANN).

2.1 DNN based automatic speech recognition

Deep neural networks based approaches are the most modern and discussed automatic speech recognition methods for past few years.

Deep neural networks are a complex architectures with structure based on human brain. They are composed of neurons arranged to several layers. First and last layer (input and output layer) are visible and other layers are hidden. Network complexity grows with increasing number of neuron layers [6], [7], [8], [9].

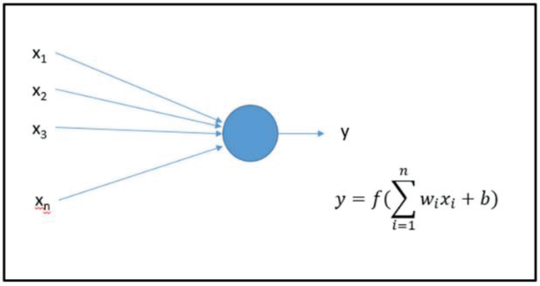


Fig. 2 Mathematical model of one neuron

The output of the y-th neuron is calculated as the non-linear weighted sum of its inputs. The input x_i of neuron may be either an user input if neuron belongs to the first layer or an output of another neuron [10].

Automatic speech recognition system Kaldi can be classified as a hybrid method, but there is a significant number of new DNN-based approaches have been published in past few years. For example Context-Dependent Deep Neural Networks (CD-DNN) based approach [11], Time Delay Neural Networks (TDNN) based approach [12], CNN-TDNN architectures [13], TDNN-LSTM architectures [13] and TDNN-LSTM-Attention architectures [13].

3 Speech recognition toolkit Kaldi

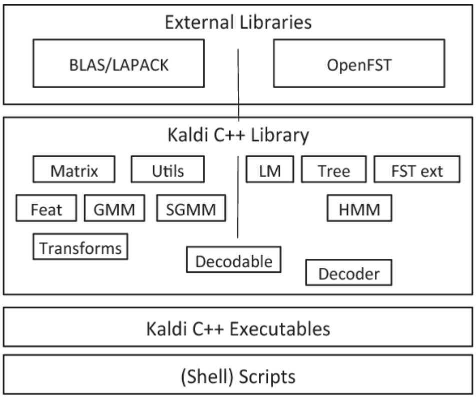


Fig. 3 Basic block scheme of Kaldi toolkit [14]

Kaldi toolkit is an open source automatic speech recognition system that is designed to be modern, flexible, and easily extensible. This toolkit is written in C++. The main advantage of automatic speech recognition toolkit Kaldi is its non-limiting license that is great for research and publication [14], [15], [16]. This toolkit has been used in our work.

4 Training with automatically rewritten data

In our experimental work we try to make training phase of automatic speech recognition system Kaldi more effective by adding a high amount of automatically rewritten data into this process. High amount of input data is crucial for acoustic and language modelling and their effectiveness. The main issue is that manual annotation of speech audio data is rather time-consuming process. When we tried to manually annotate some acoustic speech data, only one and half hour of speech audio data has been transcribed in one day (8-hour shift). For better performance of system, we would need to train on hundreds of hours, what requires a lot of time and people for annotation.

In our work we trying to enhance database used for training of automatic speech recognition toolkit Kaldi with data recognized and rewritten by some automatic speech recognition system. This data can increase system's results and automatic speech recognition's accuracy and this method is rather easy to apply.

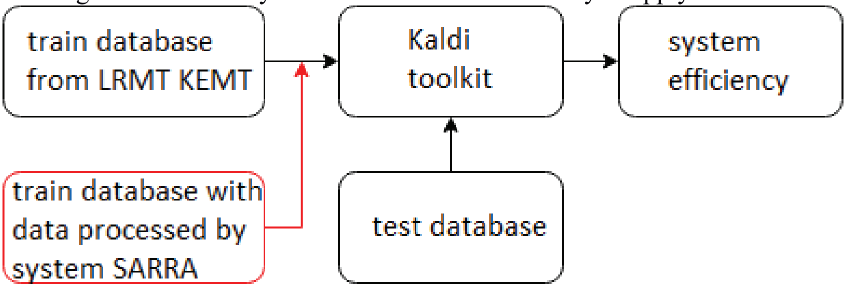

train database from LRMT KEMT
Kaldi toolkit
system efficiency
train database with data processed by system SARRA
test database

Figure 4 Block diagram of proposed automatic speech recognition system enhancement

We chose to enhance training database used and developed by our department (LRMT KEMT TUKE). Training with this database has a really good results, final WER (word error rate) parameter is less than 16%. We have merged our database with database made from meeting audio recordings that was automatically recognized and rewritten by SARRA (system for automatic speech annotation developed by our department). WER of system SARRA recognition is 15%. Our automatically transcribed database consists from 247 hours 31 minutes and 57 seconds of audio signals [17].

In our experiment we were adding parts of this database into the training LRMT KEMT TUKE database.

Table 1 Results for a test database extended by data from automatic transcript

	Volume of data added to training	WER (%)	Number of errors/number of test database words
1.	100%	15.08	5374/35626
2.	75%	14.95	5326/35626
3.	50%	15.07	5369/35626
4.	40%	15.03	5355/35626
5.	30%	15.01	5349/35626
6.	25%	14.95	5325/35626
7.	20%	14.93	5319/35626
8.	10%	14.99	5339/35626
9.	0%	15.79	5626/35626

In our first experiment we added 100%, 75%, 50%, 25% of automatically transcribed data to LRMT KEMT TUKE training database. This part of our experiment shows that this type of procedure can reduce system WER by 0.84%. This part of experiment also shows that with higher amount of added data, computational time increases and from certain limit improvement of result is minimal. In second part of experiment we only worked with smaller amount of data. In this part we reduced system WER to 14.93%. That is reduction by 0.86%. An objection may be raised, that this improvement of system's efficiency is rather small, but on the other hand application of this method is simple and easy. Further, when we use fine-tuned system with good results its improvement can be enhanced only little in comparison with systems which are in process of tuning.

5 Conclusion

Presented article introduced simple and easy applicable method for improvement of results of automatic speech recognition systems and experimental evaluation of presented method. The principle of automatic speech recognition was briefly described too, with automatic speech recognition toolkit Kaldi as an example. The core of this article described experimental results that demonstrate possibility of reducing automatic speech recognition system WER by adding high amount of automatically transcribed data.

Acknowledgment

The research presented in this paper was supported by the Ministry of Education, Science, Research and Sport of the Slovak Republic under the research project VEGA 1/0753/20, KEGA 009TUKE-4/2019 and by Slovak Research and Development Agency project APVV-15-0517.

References

1. Juhár, J. et al.: Rečové technológie v telekomunikačných a informačných systémoch. Košice: Equilibria, s.r.o., 2011. ISBN 978-80-89284-75-7.
2. ALYOUSEFI, S. H.: Digital Automatic Speech Recognition using Kaldi, Melbourn, 2018.
3. S, Karpagavalli & Chandra, Evania, A Review on Automatic Speech Recognition Architecture and Approaches, International Journal of Signal Processing, Image Processing and Pattern Recognition, 9, 393-404, 2016, doi: 10.14257/ijsip.2016.9.4.34.
4. Juhár, J. et.al.: Development of Slovak GALAXY/VoiceXML Based Spoken Language Dialogue System to Retrieve Information from the Internet. In: Proceedings of 9th International Conference on Spoken Language Processing / Interspeech 2006, Pittsburgh, PA, USA 2006.
5. Li L, Zhao Y, Jiang D, Zhang Y, Wang F, Gonzalez I, et al. Hybrid deep neural network–hidden Markov model (DNN-HMM) based speech emotion recognition. 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction. 2013. pp. 312–317. https://doi.org/10.1109/ACII.2013.
6. N.L.W. Keijsers, Neural Networks,Encyclopedia of Movement Disorders, Academic Press, 2010, Pages 257-259, ISBN 9780123741059.
7. Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, Fuad E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing, Volume 234, 2017, Pages 11-26, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2016.12.038.
8. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539.
9. Noda, K., Yamaguchi, Y., Nakadai, K. et al. Audio-visual speech recognition using deep learning. Appl Intell 42, 722–737 (2015). https://doi.org/10.1007/s10489-014-0629-7.
10. Bundzel, M., Sincak, P.: Combining gradient and evolutionary approaches to the artificial neural networks training according to principles of Support Vector Machines. In: Proceedings of the IEEE International Joint Conference on Neural Network, Vancouver, Canada, July 16-21, 2006, DOI: 10.1109/IJCNN.2006.246976.
11. Seide, Frank & Li, Gang & Yu, Dong. (2011). Conversational Speech Transcription Using Context-Dependent Deep Neural Networks. Proceedings of Interspeech. 437-440.
12. Peddinti, V., Povey, D., & Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. INTERSPEECH.
13. Georgescu, Alexandru & Cucu, Horia & Burileanu, Corneliu. (2019). Kaldi-based DNN Architectures for Speech Recognition in Romanian. 1-6. 10.1109/SPED.2019.8906555.
14. Povey, D., Ghoshal A., et al.: The Kaldi Speech Recognition Toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Hilton Waikoloa Village, Big Island, Hawaii, US, 2011. ISBN 978-1-4673-0366-8
15. Peter Smit, Sami Virpioja, Mikko Kurimo. Improved Subword Modeling for WFST-Based Speech Recognition. In Annual Conference of the International Speech Communication Association (INTERSPEECH), Stockholm, pages 2551–2555, August 2017.
16. Alyousefi, S. H.: Digital Automatic Speech Recognition using Kaldi. MSc. Thesis, Florida Institute of Technology, Melbourne, Florida, May 2018.
17. Lojka M., Viszlay P., Staš J., Hládek D. and Juhár J.: Slovak Broadcast News Speech Recognition and Transcription System. In: 21st International Conference on Network-Based Information Systems (NBiS 2018), Comenius Univ, Bratislava, Slovakia 2018, DOI: 10.1007/978-3-319-98530-5_32.

Eliminating Ambiguity in Voice Command Comprehension for Smart Room

Milan Poništ and Ivan Minárik [0000-0002-7973-0123]

Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19, Bratislava, Slovakia
{xponist,ivan.minarik}@stuba.sk

Abstract. Voice command recognition is one of the key control domains of next generation households. The smart speech assistants have already been introduced into our lives and while they are able to deal with most of our requests, they usually require speaker to communicate precise commands. This limitation forces users to focus on exact naming, making the use of the voice control system impractical and troublesome. We focus on two aspects of command processing: object- and action-based command grouping and context-aware decision making. Our simplified approach shows promising results.

Keywords: Smart household, Voice control, Ambiguous commands.

1 Motivation

The focus of this paper is addressing the ambiguity, or vagueness, that comes with the freedom of expression in spoken language. This freedom comes in use of synonyms, which tend to cause uncertainty by naming one object with different names, although with slightly different meaning. The problem is amplified by focusing on Slovak language where much less research exists in this field compared to the so-called world languages.

We created web application with simple design, which is connected to Microsoft's Azure Cognitive Services [2]. These services provide listening server, which listens to device's microphone and passes the data to Azure's center where, using machine learning, speech is converted into text, namely words compiled as a string.

2 Command Comprehension

As we want to eliminate ambiguity in voice command comprehension, we looked at how already created voice command software and its algorithms work. In [1] we have found out that they tested AM greedy decoding. AM greedy decoding have only 48.6% success rate on Libri-speech test-clean dataset. It means that their dataset is generally smaller than most of today's used datasets.

Table 1. Preview of AM greedy speech-to-text decoding, as presented in [1].

AM greedy decoding	Ground truth
the **recter pawsd** and den shaking his **classto** hands before him went on	the rector paused and then shaking his clasped hands before him went on
tax for **wone o thease** and **itees he** other	facts form one of these and ideas the other

By getting these results, we would not be happy why our commands do not work. They have tried to improve the AM greedy decoding by adding variations that capture common and consistent errors from the acoustic model to original command set. They set grammar as a set of valid voice commands (e.g. play music, stop music, etc.) and they add variations to the original grammar as a grammar augmentation (Table 2).

Table 2. Grammar augmentation, as proposed in [1].

Command (*C*)	Original grammar	Candidate set for grammar augmentation (*G*)
play music	play music	pla music, ply music, play mesic, …
stop music	stop music	stap music, stup music, stup mesic, …
pause music	pause music	pose music, pase mesic, pause mesic, …
previous song	previous song	previs song, previous son, …
next song	next song	nex song, lext song, nex song, …

As bulletproof, as this approach may seem, it does not cover the ambiguity problem all on its own.

We found that, particularly in Slovak language, there is strong use of synonymic words to describe the same object or the same action. For example, the verbs used in connection to opening curtains (in imperative) include “odtiahni”, “roztiahni” or “otvor”. Another example can be words describing lighting and illumination in the room, such as “svetlo”, “svetlá”, “lampa”, “luster”, “osvetlenie”, etc.

3 Context Awareness

Next step to take control over a smart home with our application is to understand context. By context awareness we mean making our application “smarter” with our application knowing the state of every individual device. With the state of the device, we can get data needed for us to determine how should a command cooperate and what the command should do. In [3], we have found out their approach to context awareness in the smart home with multiple devices and multiple rooms which are controlled via “artificial intelligence” is that in their work they have used Fuzzy Logic based Context Aware Algorithm, based on smart home layout:

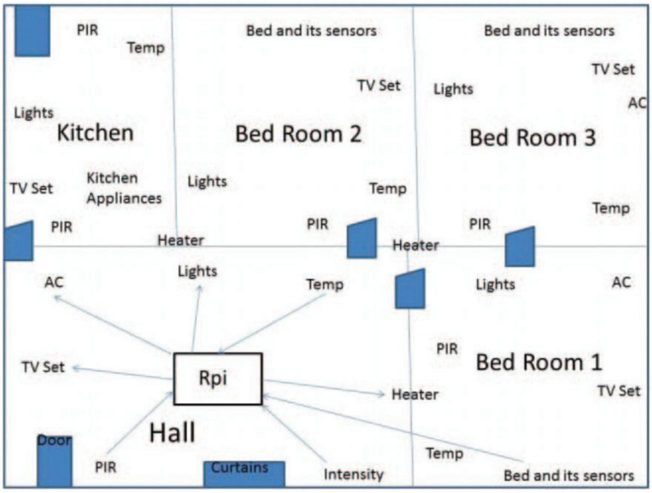


Fig. 1. Layout of a household, as shown in [3].

This algorithm includes various scenarios for various approaches to devices. For example scenario, where sensor detects the presence of person and accordingly turn ON and OFF device, or scenario, where based on soil moisture and environment sensor they turn ON and OFF a sprinkler.

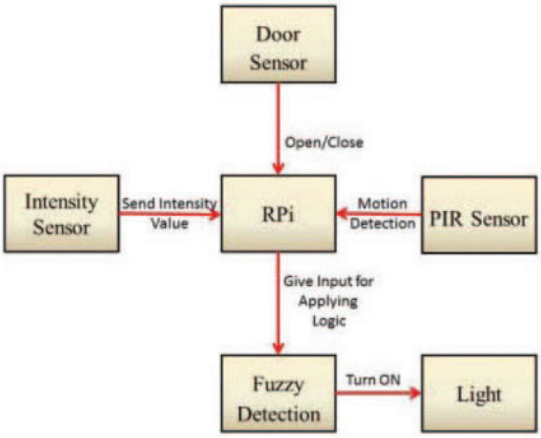


Fig. 2. Logical data flow in household control, as shown in [3].

Their results are accuracies of the context awareness of every scenario after running the algorithm for each scenario 1 000 times. Overall accuracy for all scenarios is 95.14 %.

4 Our Approach

In this article we want to focus on Slovak voice command recognition, its command grouping and context awareness and our approach to cover these features. As we mentioned earlier, we use Microsoft Azure's Cognitive Services for voice recognition. All recognized voice is decoded into data as a string. This string is passed to a variable which is used in algorithm. Algorithm consists of a "sentence creator" and phrase catcher.

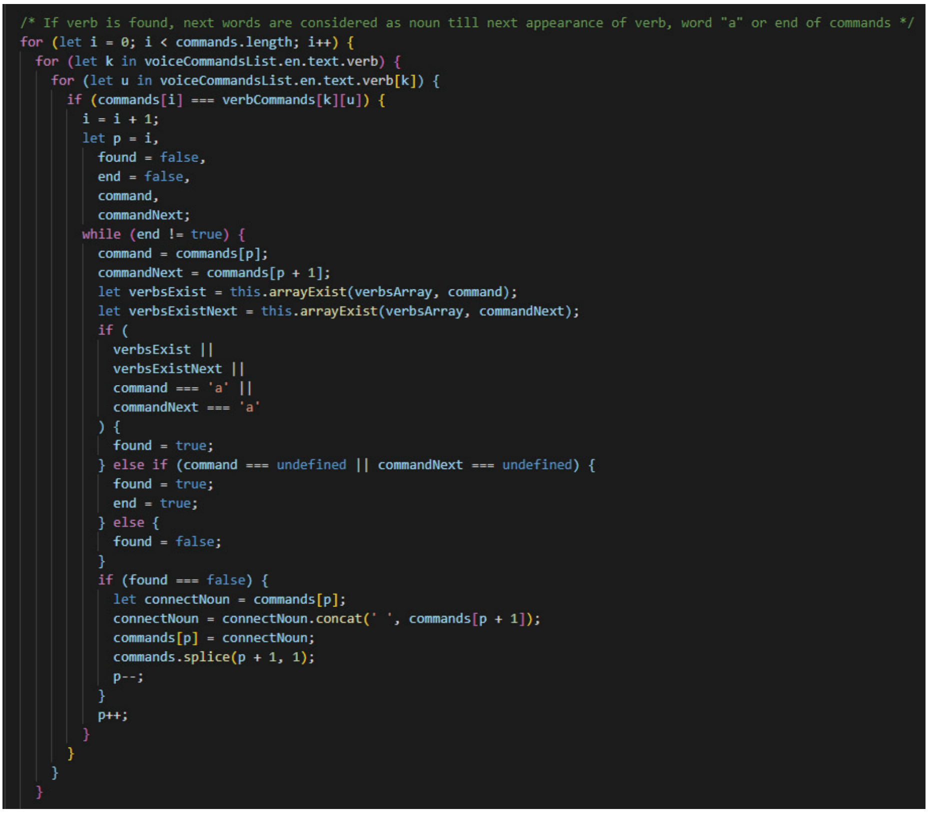
4.1 Commands composed of Actions and Objects

In order to better respond to various ways in which a command can be said, not taking into account the grammar/utterance imperfections, we introduce the concept of action types and object types.

Each action and object type has a defined command which must be performed. The algorithm identifies the spoken words and classifies them in respective action or object types. Given combination of action and object then triggers assigned command.

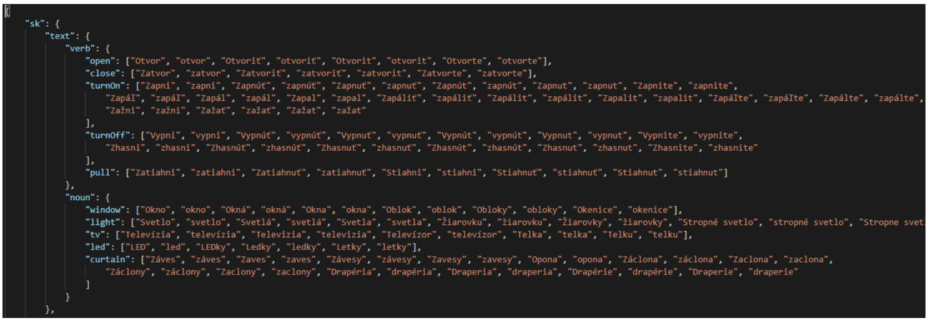
Each identified word may belong to several action or object types. If such a word is detected, probability assessment is performed on all action and object combinations which may rise from the given set of uttered words. On this level, usually one action or object can determine the correct command.

By sentence creator the algorithm decomposes the whole said sentence and put a verb in front of a noun. These algorithm helps phrase catcher catch the selected words combination and find selected phrase in a JSON object, which is used as a command to fire a backend API and execute the command. Phrase catcher uses identification of a verb in the JSON object and after the verb is identified, the algorithm knows that next words are identified as an object (noun). Afterwards this object is being identified with nouns used in JSON. If the identification is correct, the application will send the data (correct catched phrase) to the backend, otherwise the data with an error message are send. The algorithm identifies the object until there is appearance of end. The end in our code is meant as a dot, comma and words identified as for continuation of another catch phrase such as "or", "and".



```
/* If verb is found, next words are considered as noun till next appearance of verb, word "a" or end of commands */
for (let i = 0; i < commands.length; i++) {
  for (let k in voiceCommandsList.en.text.verb) {
    for (let u in voiceCommandsList.en.text.verb[k]) {
      if (commands[i] === verbCommands[k][u]) {
        i = i + 1;
        let p = i,
          found = false,
          end = false,
          command,
          commandNext;
        while (end != true) {
          command = commands[p];
          commandNext = commands[p + 1];
          let verbsExist = this.arrayExist(verbsArray, command);
          let verbsExistNext = this.arrayExist(verbsArray, commandNext);
          if (
            verbsExist ||
            verbsExistNext ||
            command === 'a' ||
            commandNext === 'a'
          ) {
            found = true;
          } else if (command === undefined || commandNext === undefined) {
            found = true;
            end = true;
          } else {
            found = false;
          }
          if (found === false) {
            let connectNoun = commands[p];
            connectNoun = connectNoun.concat(' ', commands[p + 1]);
            commands[p] = connectNoun;
            commands.splice(p + 1, 1);
            p--;
          }
          p++;
        }
      }
    }
  }
}
```

Fig. 3. Creating the command for the smart room system



```
{
    "sk": {
        "text": {
            "verb": {
                "open": ["Otvor", "otvor", "Otvoriť", "otvoriť", "Otvorit", "otvorit", "Otvorte", "otvorte"],
                "close": ["Zatvor", "zatvor", "Zatvoriť", "zatvoriť", "zatvorit", "Zatvorte", "zatvorte"],
                "turnOn": ["Zapni", "zapni", "Zapnúť", "zapnúť", "Zapnuť", "zapnuť", "Zapnút", "zapnút", "Zapnut", "zapnut", "Zapnite", "zapnite",
                    "Zapáľ", "zapáľ", "Zapál", "zapál", "Zapal", "zapal", "Zapáliť", "zapáliť", "Zapálit", "zapálit", "Zapalit", "zapalit", "Zapáľte", "zapáľte", "Zapálte", "zapálte",
                    "Zažni", "zažni", "Zažať", "zažať", "Zažat", "zažat"
                ],
                "turnOff": ["Vypni", "vypni", "Vypnúť", "vypnúť", "Vypnuť", "vypnuť", "Vypnút", "vypnút", "Vypnut", "vypnut", "Vypnite", "vypnite",
                    "Zhasni", "zhasni", "Zhasnúť", "zhasnúť", "Zhasnuť", "zhasnuť", "Zhasnút", "zhasnút", "Zhasnut", "zhasnut", "Zhasnite", "zhasnite"
                ],
                "pull": ["Zatiahni", "zatiahni", "Zatiahnuť", "zatiahnuť", "Stiahni", "stiahni", "Stiahnuť", "stiahnuť", "Stiahnut", "stiahnut"]
            },
            "noun": {
                "window": ["Okno", "okno", "Okná", "okná", "Okna", "okna", "Oblok", "oblok", "Obloky", "obloky", "Okenice", "okenice"],
                "light": ["Svetlo", "svetlo", "Svetlá", "svetlá", "Svetla", "svetla", "Žiarovku", "žiarovku", "Žiarovky", "žiarovky", "Stropné svetlo", "stropné svetlo", "Stropne svet
                "tv": ["Televízia", "televízia", "Televizia", "televizia", "Televízor", "televízor", "Telka", "telka", "Telku", "telku"],
                "led": ["LED", "led", "LEDky", "Ledky", "ledky", "Letky", "letky"],
                "curtain": ["Záves", "záves", "Zaves", "zaves", "Závesy", "závesy", "Zavesy", "zavesy", "Opona", "opona", "Záclona", "záclona", "Zaclona", "zaclona",
                    "Záclony", "záclony", "Zaclony", "zaclony", "Drapéria", "drapéria", "Draperia", "draperia", "Drapérie", "drapérie", "Draperie", "draperie"
                ]
            }
        },
```

Fig. 4. Preview of some of action and object types (denominated as “verbs” and “nouns”)

With synonyms and variations of the same word we get various approaches to final commands. The results are really promising as with growing vocabulary (JSON object) or recreating this vocabulary into database queries, we can achieve high accuracy of identifying voice commands and eliminating ambiguity in voice command comprehension. A lack of high accuracy can appear in a lack of dataset used in voice recognition itself as we use accurate words vocabulary only with variations and synonyms.

References

1. YANG, Yang, et al. Automatic grammar augmentation for robust voice command recognition. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019. p. 6376-6380.
2. Speech to Text – Audio to Text Translation | Microsoft Azure. https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/. Accessed 27 Feb 2021
3. PATEL, Arpit; CHAMPANERIA, Tushar A. Fuzzy logic based algorithm for Context Awareness in IoT for Smart home environment. In: 2016 IEEE Region 10 Conference (TENCON). IEEE, 2016. p. 1057-1060.

Comparison of Native vs Web-based Application for Smart Room Control

Matej Vanek, Denis V. Bílik, and Ivan Minárik[0000-0002-7973-0123]

Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19, Bratislava, Slovakia
{xvanekm1,xbilikd,ivan.minarik}@stuba.sk

Abstract. With only two mainstream mobile operating systems currently available, development of applications for iOS and Android has become simple with the help of cross-platform frameworks. On the other hand, progressive web applications allow for using platform-independent development with using standardized web technologies and can run in common web browsers. In this paper, we try to compare the competing technologies in a scenario where we design a smart home application and find the advantages and disadvantages of each approach.

Keywords: Smart household, Progressive web app, Native app development.

1 Introduction

Our task was to develop a way of controlling a smart room. There have been many ideas to what the best approach would be, but only two sticked to us the most. One is Progressive web app, and second is classic mobile application. While I was keen on working with native application, my colleague preferred the progressive web application. None of us could prove his approach to be better, so we decided to do some research and testing. Main goal of this article is to introduce both approaches to the reader, along with its advantages, disadvantages, to show our tests and conclude which development is better for smart room control, PWA or Native.

1.1 Previous Research

In article [1] we have found a strengths and weaknesses of individual approach, along with its features and internal workings. Author of this article recommends the use of PWA over hybrid or Native approach.

Another very interesting bachelor thesis we stumbled across was [2]. It is a thesis that compares Native and PWA applications from users perspective. They took 10 participants, 9 of them were students of Swedish Jönköping Engineering University, and last was IT specialist. They were all handed a copy of Twitter, one being Native application, and other Progressive web app. Participants were all given 12 questions and two tasks to perform. Tasks were to mute a offensive keyword, and second was to create a tweet with image in it. Surprisingly most of participants rated both platforms equally consistent despite noticing differences in UI. This shows that when it comes to user experience, both are equivalent rivals.

Another article that inspired us is a bachelor thesis [3] in which a response time is measured. They measured a response to Hardware access, Camera, Geolocation and Applications. While for some features Native app had shorter response time, for others it was PWA that won.

Some articles were in favour of Native apps, some were in favour of Progressive Web application. Since we were still divided, and there is no better way than hands-on experience we decided to each make application in our favoured approach and test it ourselves.

2 Native/Cross-platform app development

Native mobile applications are a binary executable file made for specific operating system and its devices. There are more ways to get a native application. One of them is to build fully native application only for one final platform. Another approach can be building a cross-platform application with help of a frameworks like Flutter, React Native or Xamarin. Cross-platform apps become native apps on some level since their components are being rendered to specific platforms.

Native applications are being installed directly to operating system of a device and users can launch them without any container or third-party software. Purely native apps have free access to all API's and basic functionalities such as GPS, contact list or camera. While native development requires more knowledge compared to other app development approaches, this strategy also brings the highest quality of user experience when working with it. Native applications are mostly written for two biggest platforms, Android and iOS. Applications for Android are usually written in Java, Kotlin or Objective-C, where's applications for iOS are mainly written in Swift or Objective-C.

- **Advantages:** Biggest advantage of native apps is that they are supported by all native UI and API and are receiving frequent library updates. This greatly enhances their performance since no bug remains for long time. Native apps have ability to leverage hardware and software functions for specific device. Therefore, they are taking advantage of latest technologies available and can communicate with already pre-built-in apps.
- **Disadvantages**: As mentioned above, the technology is very fragmented and it grows rapidly and native apps are taking advantages of it. While it is a positive fact, it allso is a double edged sword because these apps require constant maintanence. Another problem is that as stated before, native apps are platform specific, so for more platforms we require more source codes, making this approach very time and cost innefective. Platform-specific SDK's are allso needed, since each platforms has it's own unique set of tools.

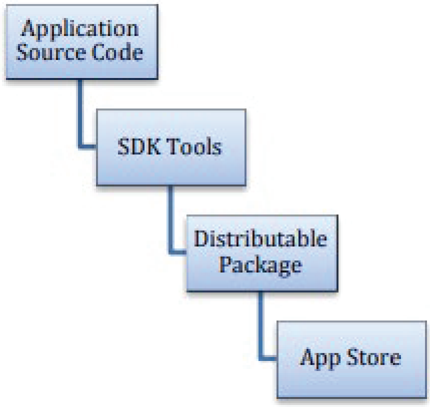


Fig. 1. To illustrate logical flow of development of a Native application

3 Progressive Web Application

Progressive web application (PWA) is a combination of the best from web technologies and mobiles applications. Like any others web sites, it consists of bunch of HTML, CSS and javascript files which are necessary to run application but in additional there are service worker file and manifest file. Service worker (SW) is a script which is write in javascript and runs in separately thread on the background. SW handles network request from our pages, processing the push notifications or do heavy calculations and main advantage is that it stores app data in caches so you can run application even when network is down. JSON file manifest is there for store metadata about app, icons, and it allows users to add your application to their home desktop like native application.

Application can run on server or you can access it locally in your PAN network. To gain full PWA certificate our application must run over HTTPS to ensure best security. To perform best user experience, apps are greatly well optimized for different screen size and the way you are interacting with it.

- **Advantages**: The Main advantage over the native application is that we don´t need to develop app for a specific platform and the visual structure of the app looks like native application. In addition, you can find PWA by web search engine, so it has wider reach for people. You can run application on any device which support compatible web browser and it is much easier to edit source code because you don't need to send request to application store and wait for their response.
- **Disadvantages**: It is relatively new technology so the older device could have limited functionality. Another disadvantage is that iOS block access to many important features such as Face/Touch ID, Bluetooth and to some sensors. We may lack performance in more complex applications and PWA can be more energy intensive than native application.

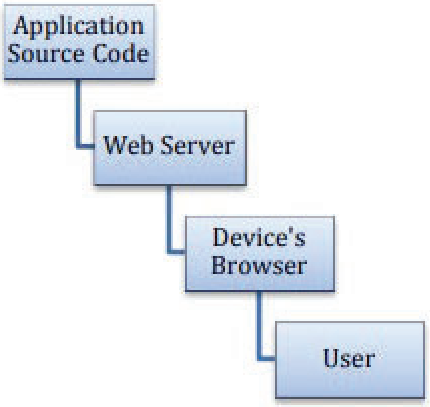


Fig. 2. To illustrate the logical flow of development of a PWA

4 Benchmarking

In order to get valid benchmarking results, we both created same application using two different approaches. Then we both installed our applications on computer connected in the networks with Home Assistant which controls all connected devices. With this step we ensured that both hardware and network would be exactly same for both applications and we started testing. Since both applications are to control one smart-room they consist mainly of buttons and sliders. We tested overall time it takes to load, how long it takes to render our most complex screen, how long it takes to render our navigation and lastly how long does it take to send a HTTP POST request over the network. For native app, 2 libraries were used. For measuring the time it takes to load the application a react-native-startup-time library was used which when added into the code it simply outputs the overall loading time to console. For the rest of measurements on mobile app a library called Reactotron was used. Reactotron has two parts. One is a program that displays our benchmarks and second is a library which is added to source code and we manually insert benchmarking starts, steps and stops.

For benchmarking Progressive web application Google have powerful analysis tool called Lighthouse which is implemented in Google Chrome. The main page fully loads after 1.51 seconds but in background service worker initiate several network request to download files which are stored in cache in case we go offline. So the page was fully loaded after 1.87s.

Although Google Lighthouse does not offer RAM measuring, we found this information in Task manager. Value of used RAM memory was oscillating around 235MB. Using http post request tooks only 0,847ms. PWA directory which include all files needed to run an application have only 84kB.

Table 1. Benchmarking results of native and progressive web application

Benchmarking	Native application	Progressive web application
Time since start up	6,79s	1,87s
Time to send POST request	3,53s	1,34s
RAM used	175MB	235MB
Size of directory	31.7kB	34,7kB
Size of installed app	39,32MB	2,76kB

5 Conclusion

After running the benchmarks none developing approach seems to shine over another. PWA is slightly lighter than regular mobile app but mobile app is more flexible. When it comes to choosing we have agreed that the most defining factor should be number of people using the application. For big companies or generally bigger group of people we suggest Progressive web app since it does not require downloading and is always ready to work for anyone with url to website and authentication, of course that is if there is some authentication set up on website. For personal projects or small defined groups of people whose numbers won’t change rapidly we strongly recommend classical mobile application. It is slightly more powerful, a little more flexible but it requires a download. It also has lower risk of being compromised because mobile apps just send and pull date through internet and handle all data locally, while PWA is in its entirety set up online.

References

1. Adetunji, Oluwatofunmi & Ajaegbu, Chigozirim & Nzechukwu, Otuneme. (2020). Dawning of Progressive Web Applications (PWA): Edging Out the Pitfalls of Traditional Mobile Development. American Scientific Research Journal for Engineering, Technology, and Sciences. 68. 85-99.
2. J. Sedkowska, ‘How does the user experience of a progressive web application compare to native application?: A case study on user’s attitude in context of social media.’, Dissertation, 2020.
3. Fransson, Rebecca, and Alexandre Driaguine. Comparing progressive web applications with native android applications: an evaluation of performance when it comes to response time. (2017).

This page is intentionally left blank.

An Approach to Smart Parking System

Jozef Genšor[1] and Radoslav Vargic [1]

[1] Slovak University of Technology in Bratislava, Faculty of Electrical Engineering and Information Technology, Applied Informatics

Abstract. In this contribution, we present an approach to smart parking system. The proposed system uses a camera location above the parking place. system for reading related disorder detection. Multiple methods as components were proposed and evaluated. The results favorize the simpler method, based on the traditional approach based on the texture analysis.

Keywords: Smart parking, Background subtraction, Edge detection

1 Introduction

From surveys from 2019, every second household in Slovakia has at least one car. If we look at the situation in our capital, less than a third of the population has a reserved parking space, whether at work or in residential areas. As for parking spaces, it has been proven that each person spends an average of 8 minutes a day searching for a parking space. This daily routine of ours is not only very frustrating for our mental health but also for the environment. In this contribution we focus on the issue of parking spaces. For video dataset capture we selected the typical supermarket parking lot. We applied motion detection methods and evaluated and processed the results. An example of an existing system for detecting parking spaces by camera recording based on image processing technique [4].

1.1 Overview of Smart parking systems

Various methods are currently used to monitor the number of free parking spaces. The most common method is to count free parking spaces [6]. This method works on the principle of calculating the difference between the maximum capacity and the current number of cars that entered or left the parking lot. This method is the most simple, but without specific information about which of the parking spaces is free. In practice, to find out information on the occupancy status for a specific parking space there are used additional sensors. The two most used traditional sensors are ultrasonic sensor and inductive sensor. Ultrasonic sensor is the most common, especially in underground garages. It is also widely used in the automotive industry as a parking sensor. This sensor detects the distance of the object from the sensor. It works on the principle of high frequency sound waves [1]. A sensor located above the parking space emits waves that are reflected from the road when the parking space is empty. The evaluation electronics of the sensor calculates the time interval between the sent and received signal. When the car is parked, this time interval changes to determine if an object is there or not. An example of smart parking system using ultrasonic control sensor [7]. The advantage of this sensor is the evaluation accuracy. The disadvantage is the complicated installation, as each parking space means a dream whose purchase price is relatively high. In the inductive sensor is the active element a coil. An alternating current flows through the coil, creating a magnetic field [2]. If an object made of an electrically conductive material is in its vicinity, the magnetic field of the coil is disturbed. The inductive sensor is usually mounted directly in the roadway of the parking space. The city of Santander in Spain also experimented on the implementation of a smart parking solution, with parking lots with built-in inductive sensors [8]. The advantage is insensitivity to wear, resistance to short circuits, price. The disadvantage is the complicated installation of the already built parking lot. Another method for monitoring the condition of car parks and parking spaces is camera detection. An essential part of this method is a camera that continuously scans the car park and the operating software.

2 Reference dataset

For a suitable parking lot, we consider those that have parking spaces situated as large as possible in relation to the camera's shooting angle. An example of suitably and inappropriately seated parking spaces (see Fig. 3). The reason is the situation when, in horizontally situated parking spaces, a higher parked car would occupy a considerable part of the parking space above it. This would lead to distorting results, or even to the impossibility of detecting the occupancy status for non-peripheral parking spaces. Such a situation should be solved by placing the camera as perpendicular as possible above the parking spaces. In practice, however, this would mean that the camera would have to be placed on a high object around the parking lot with the ability to optically zoom in large multiples.

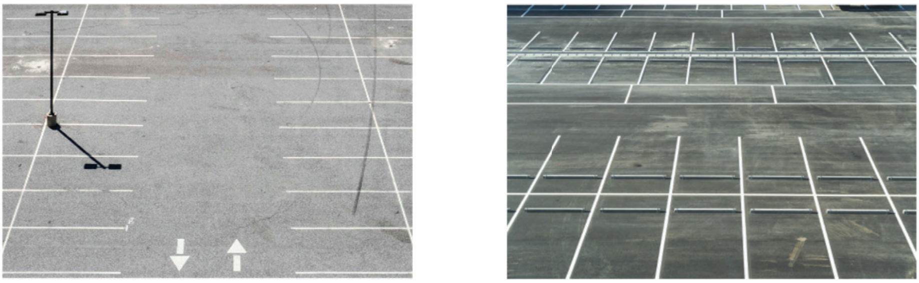


Fig. 3. Example of inappropriate (left) and appropriate (right) layout of the parking places in the camera view.

In the selected layout on Fig. 3 right we empirically experienced that reasonable number of cars that can be effectively monitored is about 15 – the more distant parking places do overlap. In practice, it would be necessary to place more cameras, which would be placed on an elevated object if the parking lot is filmed or placed. In the dataset , any departure or arrival of a car other than the initial condition is considered to be a change. The definition of parking spaces consists in defining shapes with four coordinates. Coordinates are the height and width of a given point depending on the resolution of the video. Each shape of the parking space is drawn in each frame of the video. Since the coordinates are fixed, it is important that the captured image does not move. The example situation is depicted on Fig. 4. For each parking space there is constructed a mask as rectangular region of interest. The important were also choice of spaces between parking places to avoid interferences between them.

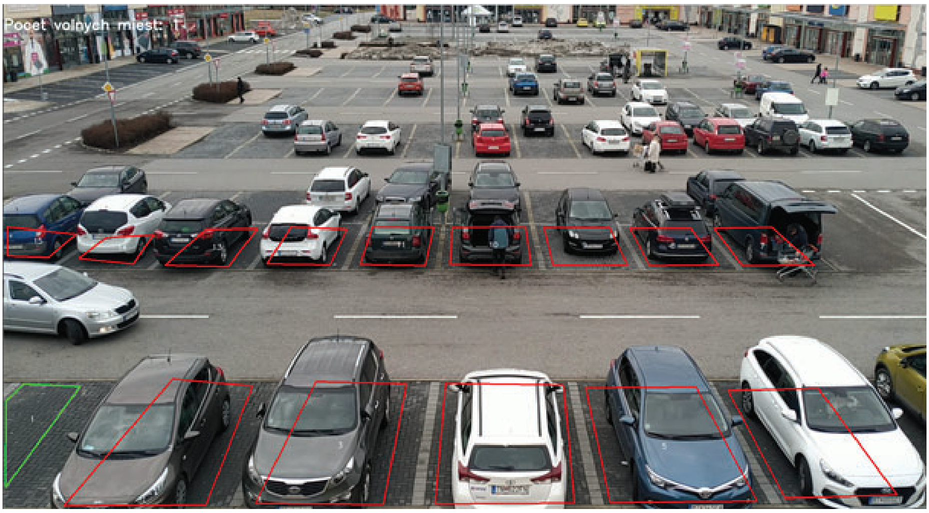


Fig. 4. Marking of monitored parking spaces.

3 Proposed methods

We tried two independent methods. First – method A tries to guess – based on texture analysis whether the parking lot is free or occupied by car. The method is based on number of edges [5], present on specific parking place. We start by converting the image to grayscale. This is followed by gaussian blur, then edge detection (Laplacian), then erosion. This helps us to distinguish between free parking place (few edges) and occupied place (lot of edges) for which we can set a reasonable threshold. An example is shown on Fig. 5. Method as is combined with method B, which is background subtraction with continuously updated background model with selectivity [3]. To get stable object tracing we use gaussian blur and threshold. Thus, we can detect the object movement. To interpret the movements, we split each parking place into three zones as depicted on Fig. 6. Then we simply measure the number of moving pixels in the zone and employ the thresholding models with gaussian averaging and hysteresis.

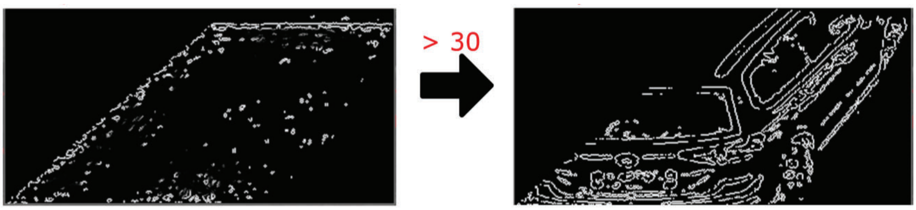


Fig. 5. Example of free (left) and occupied (right) masked parking place. The images can be easily separated by checking average pixel value, lefx=15, right=40, the threshold is set to 30

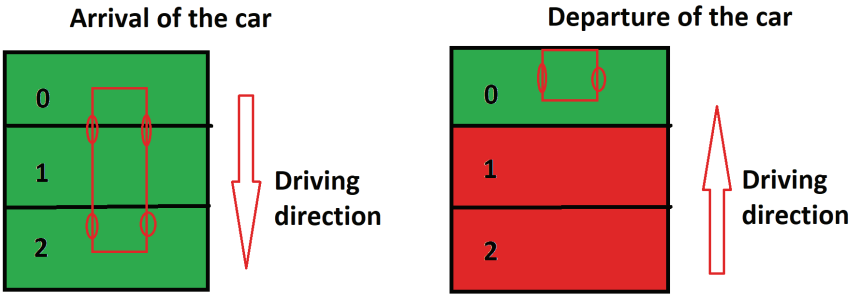


Fig. 6. Split of the parking place to 3 zones to robustly detect arrival or departure of wehicle.

4 Results

The method A offers robust results for the evaluated dataset. Despite its simplicity, the detected success rate is was 100%. The more promising method B we found more problematic. When the initial background is not properly initialized when there are no cars, then the system learns wrong background (the inverse one). So for heavily loaded parking places, even with parkings over night this can this seems to be significant problem and additional logic shall be impoyed to allow proper (re) initialization.

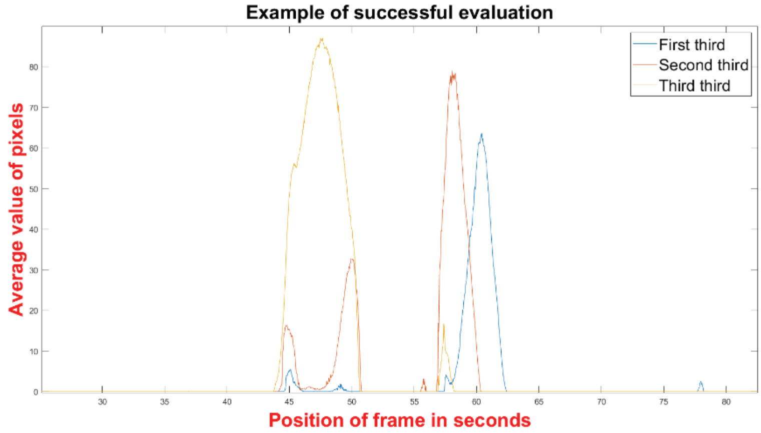


Fig. 7. Example of car movement in method B – relative amount of moving pixels in the all thirds of the parking place (so the movement is in direction third third → second-third → first third)

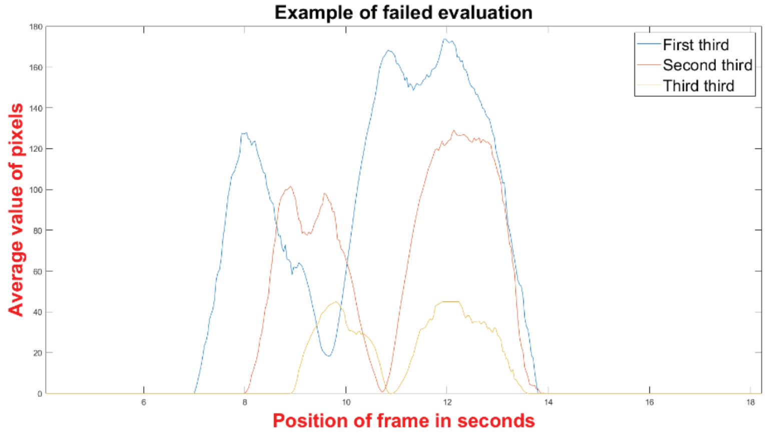


Fig. 8. Example of car movement in method B – wrong and overlapping detection

In our dataset, due to the above mentioned problem, the succesrate for method B was only about 40%. An examples of the feature values used for detection by threshold are depicted on the Fig. 7 (positive case) and Fig. 8 (negative case).

5 Conclusion

In the paper we presented simple but effective methods for smart parking solution based on video analytics. We provide simple, robust and fast method (method A) for parking place occupancy evaluation. The more promising method B based on background subtraction we considered as less stable under given conditions. We did not employ neural network du to simplicity reasons and tried to avoid sufficient efficiency using traditional methods which was successful.

Acknowledgment

Research described in the paper was financially supported by the Excellent creative team project VirTel.

References

1. Ng, Fares: Ultrasonic Sensors, 2020, DOI: 10.13140/RG.2.2.33638.78404
2. Lvliang Liu: Analysis of Operation Characteristics of Inductance Sensor, IOP Conference Series: Materials Science and Engineering, 2018, DOI: 10.1088/1757-899x/452/4/042114
3. Elgammal, Ahmed: Background subtraction : theory and practice, Morgan & Claypool Publishers, 2015, ISBN: 978-1-62705-440-9
4. Hilal Al-Kharusi and Ibrahim Al-Bahadly, Intelligent Parking Management System Based on Image Processing, World Journal of Engineering and Technology, 2014, DOI: 10.4236/wjet.2014.22006
5. AMER, Ghassan Mahmoud Husien a ABUSHAALA, Ahmed Mohamed. Edge detection methods. In: 2015 2nd World Symposium on Web Applications and Networking (WSWAN). IEEE, 2015. DOI: 10.1109/wswan.2015.7210349.
6. Ali Abd Al-Zahra, Murtatha Falah, Zain Hussam, Design and Implementation of Smart Car Parking System, June 2016 [Online]. Available: https://www.researchgate.net/publication/320356747_Design_and_Implementation_of_Smart_Car_Parking_System
7. Yousif Allbadi and Jinan N. Shehab and Musaab M. Jasim , The Smart Parking System Using Ultrasonic Control Sensors, IOP Conference Series: Materials Science and Engineering, 2021, DOI: 10.1088/1757-899x/1076/1/012064
8. Pablo Sotres and Carmen Lopez de la Torre and Luis Sanchez and Seung Myeong Jeong and Jaeho Kim, Smart City Services Over a Global Interoperable Internet-of-Things System: The Smart Parking Case, 2018 Global Internet of Things Summit, DOI: 10.1109/giots.2018.8534546

Comparing Canonical Versions of Covariance Matrix Adaptation Evolution Strategies

Bílik Dominik[1], Martiška Adam[1], Otruba Jakub[1] and Juraj Kačur[1]

[1] Slovak University of Technology, Faculty of Electrical Engineering and Information Technology, Ilkovičova 3, Bratislava, Slovakia

Abstract. Evolution strategies are very widely applicable method of optimization. This paper describes a system for finding a minimum of a multidimensional discontinuous function and monitoring this fitness function across generations. The system utilizes both $(\mu/\mu_I, \lambda)$-CMA-ES and $(\mu/\mu_I + \lambda)$-CMA-ES versions of ES. The paper then compares results and behaviour across the whole process of these two versions.

Keywords: Evolution Strategies, Optimization , CMA-ES.

1 Introduction

This article will discuss possible implementation of a system, that finds the minimum of a multidimensional continuous function using evolutionary strategies such as ES (μ + λ) and ES (μ, λ). Our approach is based on a derandomized ES with covariance matrix adaptation (CMA-ES).

1.1 Evolution Strategy

Evolution strategies (ES) are a sub-class of nature-inspired direct search methods belonging to the class of Evolutionary Algorithms which use mutation, recombination and selection applied to a population of individuals containing candidate solutions in order to evolve iteratively better and better solutions. So evolution algorithm is basically based on the principle of biological evolution, that is why we use recombination, which represents the selection of a new mean value for distribution. In principle, it is a stochastic search algorithm that minimizes a nonlinear objective (fitness) function. The search steps are done by stochastic variation, the so-called mutation of points found so far. The best of these points (offspring and parents, based on used strategy) is chosen to be continued for the next generation. Choice is being made based on a fitness (objective) function. Mutation is usually carried out by adding a realization of a normally distributed random vector. It is easy to imagine, that the parameters of the normal distribution play an essential role for the performance of the search. Thus, new populations are constantly being created. Each population has so-called parents, from whom offspring are created, from which the best are selectively selected in order to improve the next population.

1.2 Covariance Matrix Adaptation

Or CMA-ES (Covariance Matrix Adaptation Evolution Strategy) is a particular kind of evolution strategy for numerical optimization. Using CMA-ES the shape of mutation distribution is generated according to a covariance matrix C, which is adapted during evolution. Thus, the mutations can adapt to the local shape of fitness landscape and convergence to the optimum can be increased considerably. It uses special statistics cumulated over the generations to control strategy-specific parameters (the covariance matrix C and the step size σ). Covariance matrix is a square, symmetric matrix giving covariance (measure of the joint variability of two random variables) between each pair of elements of a given random vector, thus it's main diagonal contains variances. In this example, the size of covariance matrix is dependent on number of dimensions used. The primary feature of the CMA-ES is its realiability in adapting an arbitrarily oriented scaling of the search space in small populations.

2 Algorithm

2.1 Initialization

Before implementing CMA-ES we have to take into account that we have two options for implementing this strategy ES ($\mu/\mu_I, \lambda$), where parent population is created only from offspring population and ES ($\mu/\mu_I + \lambda$), where the new parent population includes also best parents from previous generation. We distinguish between these strategies with operand variable that we define in the beginning of the algorithm. We also define following initializing variables: number of offspring λ, number of parents μ, standard deviation (step size) σ, minimal standard deviation that needs to be reached in order to stop our algorithm σ_{min}, vector of size 1 x d, where d is number of dimensions y, covariance identity matrix of size $C = I$, number of maximum generations g_{max} and our fitness function, that we want to optimize $f(x)$.

At first we want to initiate first parent population creating each individual parent P as a structure that contains: randomly selected multidimensional input parameters for our fitness function y, normally distributed random vector N, weight of each individual that connects the recombinants of two consecutive generations and represents the tendency of evolution in the search space w and result of our fitness function evaluated with input of y $F(y)$

The main loop consists of three main parts:

1. sampling of new solutions
2. re-ordering of the sampled solutions based on their fitness
3. update of the internal state variables based on the re-ordered samples

For this main loop we defined 2 conditions, at least one of them need to be met to stop the loop:

- Number of generations reaches its maximum value predefined in initial parameters
- σ for the current generation of parents is equal or lower than our σ_{min}

2.2 Creating offspring population

Our goal is to generate λ offspring population that has same structure as our initial parent structure. Values for each offspring are generated as follows:

$$w \; = \sigma\sqrt{C}N(0,1) \tag{1}$$

Where *N(0,1)* is new normally distributed random vector, *C* is adapted covariance matrix for current generation, σ is step size for current generation and *w* is offspring weight based on previously mentioned variables.

$$y = y + w \tag{2}$$

Where *w* is weight of current offspring generated in previous step and *y* is input column vector.

$$F = f(y) \tag{3}$$

Where *F* is function output value of our selected fitness function that accepts *y* as an input vector.

2.3 Sorting offspring population – creating parent population

After calculating these values for each individual offspring we created offspring generation, which needs to be sorted in order to get the set of best offspring – new parent generation. Sorting is either performed on only offspring generation, or on generation composed of offspring and parents from previous generation depending on chosen operand. Regardless of the number of input individuals, we want to create generation of μ best parents. In order to observe behavior of our evolution strategy we are also searching for best individual (the one that fits our acceptance criteria the most) and mean calculated from best set of sorted offspring.

2.4 Updating internal state variables

In the last step we need to update our state variables based on our parent generation. We are calculating mean of our parent weights and mean of the normally distributed random vector *N* for our chosen parents. These values are used for updating our input vector *y*, adapting covariance matrix *C* for next generation and calculating standard deviation σ for our current generation. The standard deviation σ accounts for the level of exploration: the larger σ the bigger search space we can sample our offspring population. It controls the overall scale of the distribution, often known as step size. Very last step is to check our terminating conditions: whether our current generation is lower than our maximum number of generations g_{max} or if our new calculated σ is still higher than our minimal standard deviation σ_{min}. If both conditions are met we continue with our algorithm until we find optimized result (function minimum in our case)

3 Results

3.1 Test functions

We decided to use 2 different functions as our fitness function, first of which was a simple ellipsoid function given by:

$$f = \sum_{i=1}^{d} 10^{6\frac{i-1}{d-1}} {x_i}^2 \tag{4}$$

To test how the system behaves with more complicated functions, we decided to use "Ackley" function, commonly used for testing algorithms looking for optimum of a function. "Ackley function is given by:

$$f = 20(1 - \exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} {x_i}^2}\right)) - \exp\left(\sqrt{\frac{1}{d}\sum_{i=1}^{d} \cos\,(2\pi x_i)}\right) + exp(1) \tag{5}$$

Both of these functions have their optimum equal to 0 in f(0,0,…)

3.2 Methodology

For easier display of features of parents in every generation, we did all the testing with *d = 2,* this way we could observe how the system behaves. For number of parents *μ* and number of offspring *λ* in each generation, we chose: *μ = 100* and *λ = 500.*

Based on changes on the changes to the algorithm described in (Hansen, Müller and Koumoutsakos 2003), which aimed at optimizing it for lower dimension sizes and higher population sizes, we used following strategy parameters:

$$c_c = c_\sigma = \frac{4}{d+4}$$
$$c_{cov} = \alpha_{cov}\frac{2}{(d+\sqrt{2})^2} + (1-\alpha_{cov})\min\,(1, \frac{2\mu-1}{(n+2)^2+\mu})$$
$$\alpha_{cov} = \frac{1}{\mu} \tag{6}$$

We also set a maximal number of generations to 30 for better comparison.

3.3 Best fitness

In the first test we observed minimal functional value in each generation for both canonical versions of our ES. Example shown is using Ackley function.

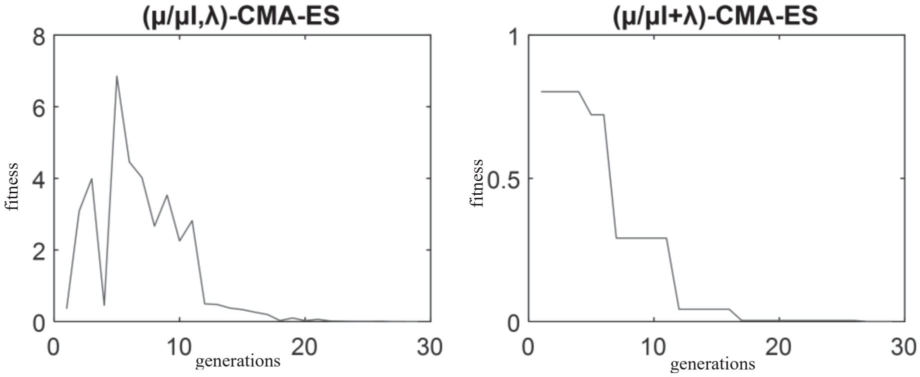


Fig. 1 comparison of best individual fitness across generations

Note that we opted to put these graphs side by side, as the scale would make $(\mu/\mu_I + \lambda)$ -CMA-ES hardly readable.

We can see in this comparison, that for $(\mu/\mu_I + \lambda)$ -CMA-ES, the graph is monotonously decreasing, meaning that even if we stop the algorithm before it has reached the minimum, we will see some sort of improvement. However this value can be a local minimum. This is caused by the fact that the best scoring individual is always retained if there wasn't any better individual in between the offspring. We can also see that in $(\mu/\mu_I, \lambda)$ -CMA-ES, the value raises massively, this is caused by the initial divergence, which we will observe better in later tests.

3.4 Mean fitness

Second test was very similar to the first one, only we didn't observe the best individual, we focused on the mean functional value across all newly chosen parent population.

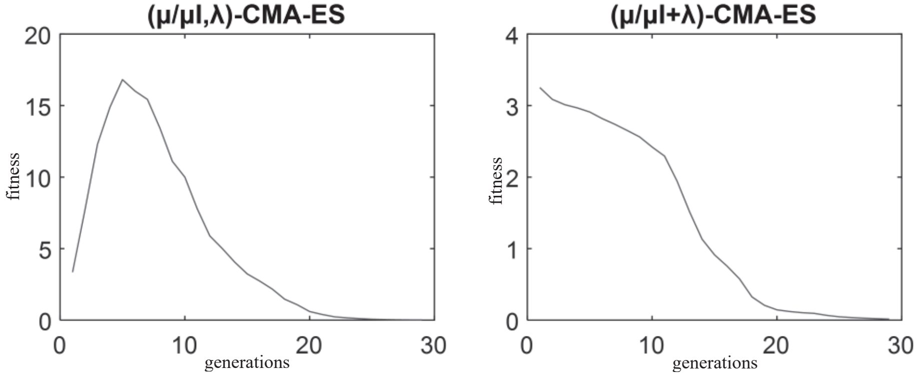


Fig. 2 comparison of mean fitness in parent population across generations

Much like in the previous test, we can see that because we retain the part of the population which isn't being improved upon, the fitness of $(\mu/\mu_I + \lambda)$ -CMA-ES population is only improving. Depending on what margin we use, it might appear that $(\mu/\mu_I + \lambda)$ -CMA-ES reaches optimum faster then $(\mu/\mu_I, \lambda)$ -CMA-ES.

3.5 Pupulation location

Lastly we looked at "location" of parent population across generations, or rather their features inputting into the fitness function.

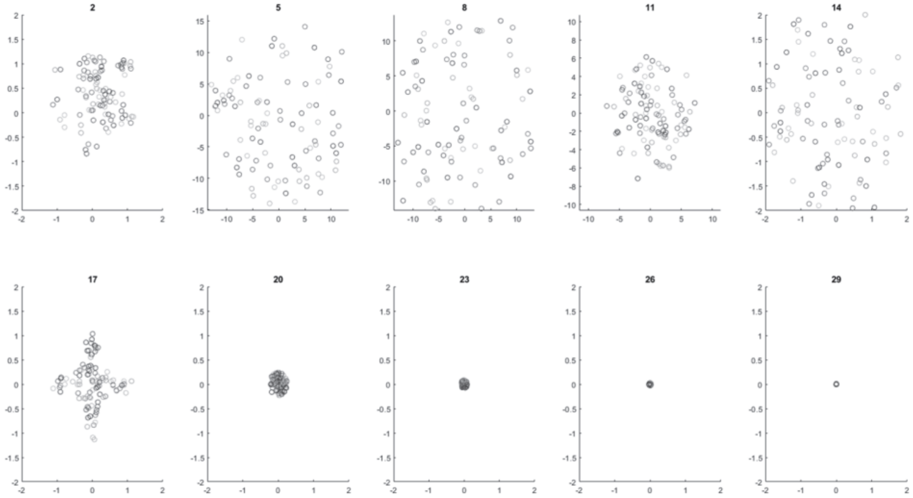


Fig. 3 parent features across generations in $(\mu/\mu_I, \lambda)$ -CMA-ES

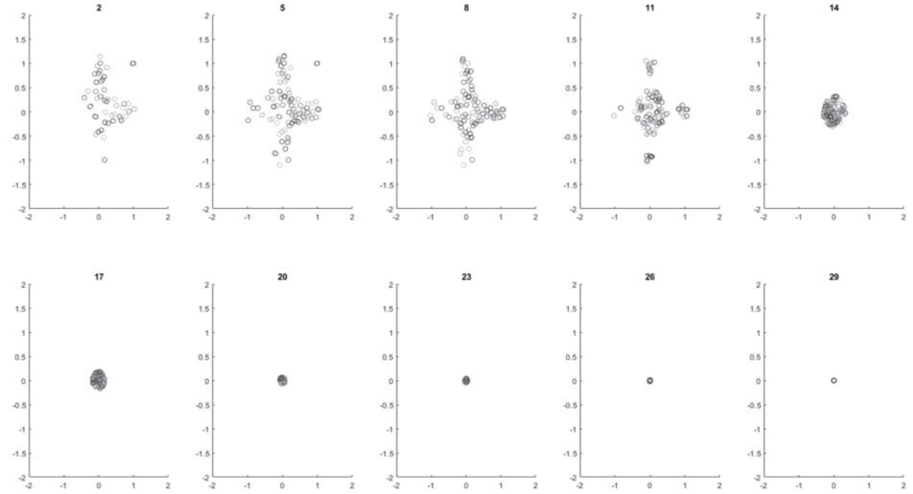


Fig. 4 parent features across generations in $(\mu/\mu_I + \lambda)$ -CMA-ES

Couple of notes here, in the first figure, 5th, 8th and 11th generations are massively zoomed out, to see the spread of parents. Also note, that the first generation is not displayed here, since all the parents from the first generation had their features set to 1.

When looking at progression of $(\mu/\mu_I, \lambda)$ -CMA-ES, we can clearly see that the parents features massively diverge from the optimum in the first generations. In $(\mu/\mu_I + \lambda)$ -CMA-ES we can clearly see the parents kept from previous generation. We can also observe positions of local minimums in 11th generation of $(\mu/\mu_I + \lambda)$ -CMA-ES, where the smaller clusters are located. these aren't a problem, since the new generation isn't calculated from the parents located at these points, rather from a recombinant of all the parents.

3.6 Conclusion

From the results, we can clearly see that $(\mu/\mu_I + \lambda)$ -CMA-ES version of the algorithm reaches the optimum faster with the parameters set right. In our testing however, we noticed, that if the parameters aren't set correctly, $(\mu/\mu_I + \lambda)$ -CMA-ES can get stuck before the optimum with σ already reaching zero. This could probably be averted by changing the way σ is updated specifically when "+" is used. Another downside of this version is, that it takes longer to compute, since the choosing of the new parent population is made from $\mu+\lambda$ individuals rather then just λ.

Over all, the number of generations when both canonical versions reached the minimum were comparable. when looking at low amount of generations, $(\mu/\mu_I + \lambda)$ -CMA-ES outperforms the other version by far, since it doesn't diverge before it converges to the minimum.

Acknowledgment.

This publication was created thanks to support under the Operational Program Integrated Infrastructure for the project: International Center of Excellence for Research on Intelligent and Secure Information and Communication Technologies and Systems - II. stage, ITMS code: 313021W404, co-financed by the European Regional Development Fund

References

1. Hansen, N., Ostermeier, A.: Completely Derandomized Self-Adaptation in Evolution Strategies., Evolutionary Computation *(2001) 9 (2): 159–195.*
2. Hansen, N., Müller, S., Koumoutsakos, P.: Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)., *Evolutionary Computation* (2003) 11 (1): 1–18.
3. Hans-Georg Beyer (2007) Evolution strategies. Scholarpedia, 2(8):1965.

This page is intentionally left blank.

Application in Augmented Reality

Lívia Bzdilová and Gregor Rozinaj

Faculty of Electrical Engineering and Information Technology, Ilkovičova 3, 812 19 Bratislava,
Slovak Republic
redzur@stuba.sk

Abstract. Nowadays, the need for Information and Communication technology is more important than ever before. A lot of fields were interrupted due to pandemic - such as education, work, culture and many others. The only way, how we can communicate, work from home, is just because of these technologies. One of the fields, that has been affected, is Sport. In this article, we will describe the process of creating and designing AR application with workout topic. Augmented reality is the type of reality that superimposes digital data and images on the physical world. The application is designed so, that anyone can practice their training from the comfort of home. While we are practicing the workout, we can walk around the avatar to see how to do the exercise properly. To achieve the goal, we worked in Unity platform, Microsoft Visual Studio Community, which is necessary for the creation of scripts controlling the management of individual components of the application and modules - Android Build Support and Vuforia Augmented Reality Support. This configuration is used for our entire project. The most important part of this work is the use of AR smart glasses, that add the extra information - such as 3D images, video into the user's real life.

Keywords: Augmented Reality, Unity, Vuforia.

1 Problem analysis

Although we may not even realize it many times, Augmented reality, like Virtual reality, is used on a daily basis today. These are concepts that belong to the achievement of modern times and communication technologies that affect our lives, whether we like it or not.

In this article, we deal with the approach to the topic, ie what augmented reality means, where it is used. Our topic deals with the augmented reality in education, but of course this kind of reality is also used in many other areas. I also described the difference between augmented reality and other kinds of reality, ie virtual and mixed. I also mentioned the types of augmented reality we know.

The next part describes the actual implementation of application, which relates to physical activity-exercise, which is at this time modern technology, respectively long sitting at the computer, production and advantage. This section also describes the hardware used to create applications, describes the instructions, design and functionality of application.

1.1 Difference between VR and AR

As regards AR, compact definitions have been proposed by many scientists. For example, in 1997, Ronald T. Azuma defined AR as a collection of applications that verify the following three properties:
1) a combination of the real and the virtual;
2) real-time interaction;
3) integration of the real and the virtual(e.g. recalibration, obstruction, brightness).[1]

Even though they share algorithms and technologies, VR and AR can be clearly distinguished from each other. The main difference is that in VR the tasks executed remain virtual, whereas in AR they are real. For example, the virtual aircraft that you piloted never really took off and thus never produced CO2 in the real world, but the electrician using AR may cut through a gypsum board partition to install a real switch that can turn on or off a real light.

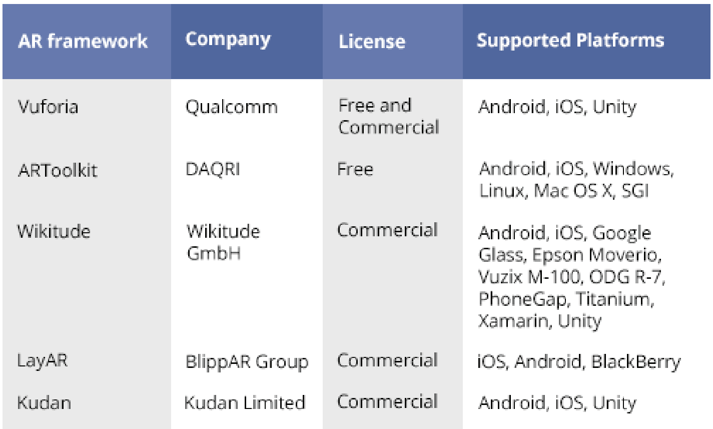
1.2 AR applications development

Why develop AR applications? There are several important reasons:
Driving assistance: originally intended to help fighter jet pilots by displaying crucial information on the cockpit screen so that they would not need to look away from the sky to look at dials or displays (which can/could have been be crucial in combat), AR gradually opened up the option of assisted driving to other vehicles (civil aircraft, cars, bikes) including navigation information such as GPS.
Tourism: by enhancing the capabilities of the audio-guides available to visitors of monuments and museums3, certain sites offer applications that combine images and sound.
Professional gesture assistance: in order to guide certain professional users in their activities, AR can allow additional information to be overlaid onto their vision of the real environment. This information may not be visible in the real environment, as it is often “buried”. Thus, a surgeon may operate with greater certainty, by visualizing the blood vessels or anatomical structures that are invisible to them, or a worker participating in constructing an aeroplane may visually superimpose a drilling diagram directly onto the fuselage, without having to take measurements themselves, which leads them to gain speed, precision and reliability.
Games: while it was popularized by Pokémon Go in 2016, AR made inroads into this field a long time ago, through the use of augmented versions of games such as Morpion, PacMan or Quake. It is clear that this sector will see a lot more development based on this technology, which will make it possible to combine the real environment and fictional adventures. [2]

1.3 Augmented Reality SDK

There are several libraries that are needed to create an Augmented reality application.

Table 1 : Augmented Reality SDK comparsion

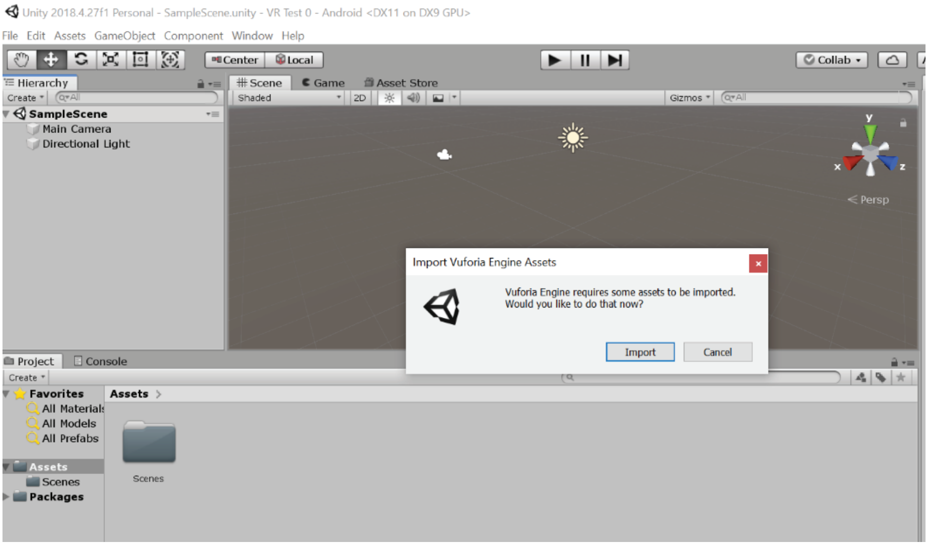


AR framework	Company	License	Supported Platforms
Vuforia	Qualcomm	Free and Commercial	Android, iOS, Unity
ARToolkit	DAQRI	Free	Android, iOS, Windows, Linux, Mac OS X, SGI
Wikitude	Wikitude GmbH	Commercial	Android, iOS, Google Glass, Epson Moverio, Vuzix M-100, ODG R-7, PhoneGap, Titanium, Xamarin, Unity
LayAR	BlippAR Group	Commercial	iOS, Android, BlackBerry
Kudan	Kudan Limited	Commercial	Android, iOS, Unity

2 Solution implementation

2.1 New Project

After installing Unity and Microsoft Visual Studio, we can create a new project. In Unity Hub, choose the version of the Unity Hub Editor 2018.4.28f1 and the type of 3D project. A sample scene is automatically created with the Main Camera and Directional Light elements added. However, for our AR application to work properly, we will need to replace the Main Camera element with an AR camera. To add this, select GameObject -> Vuforia Engine -> AR Camera in the main top menu. Unity will automatically notify us that the import of some assets is required. We will approve the import and Vuforia Engine will ensure the import of the necessary components, including the AR Camera.



Import Vuforia Engine Assets

Vuforia Engine requires some assets to be imported.
Would you like to do that now?

Fig. 1 : Components import

2.2 Animations creation

The creation of all animations took place in the Blender program, where we imported our Avatar from mixamo.com together with the animations, which we further modified. When downloading the avatar, we chose Format FBX For Unity (.fbx) and Original Pose (.Fbx). In Blender, we imported the avatar in the menu File> Import> FBX (.fbx)> Import FBX. We could work with animations on the embedded avatar. We monitored the individual movements in the Timeline section, where we could also choose to display keyframes or seconds. Here we set all the animations to the avatar, which we then disassembled in Unity as needed. When exporting, we first selected our object and in the File> Export> .FBX menu we selected the Selected objects option in the Limit to section and deselected the Add Leaf Bones option in the Armature FBX section. We set the location and name and confirmed by clicking the Export button.

2.3 Animator Controller

For each scene in which the training is realized, we have created our own program Animator Controller for the avatar, ie for abdominal training for beginners, abdominal training for advanced etc. We've added all the states it can contain to the Animator, so start, workout, pause, and end. It was not necessary to set any transition conditions between states for all products except your own choice of exercises.

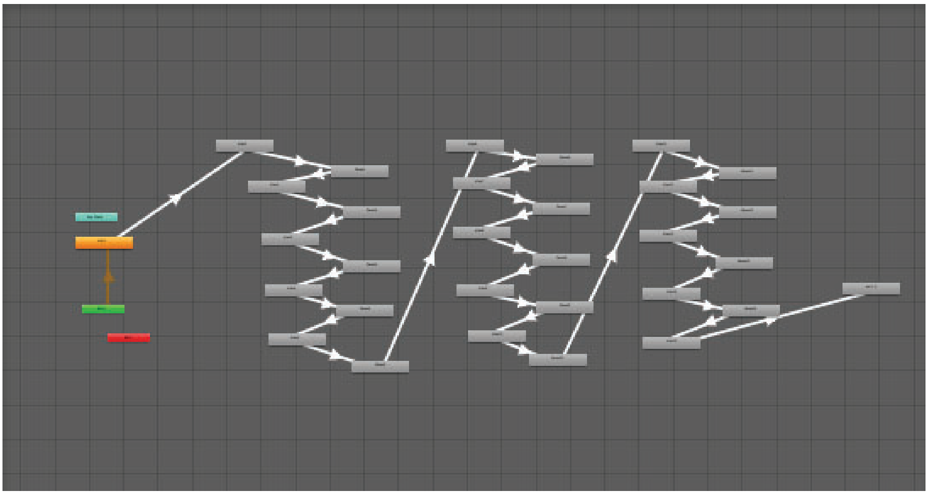


Fig. 2 : Animator for 5 exercises

When creating Own training category, we also set the conditions for individual states, ie for example, if it goes from exercise 1, then it must go immediately to pause 1 and from pause 1 it can go to exercise 2 to 17. So, for example, if we marked exercise 3 at the beginning, so it is necessary to set the conditions that if it goes from start to exercise 1, it automatically continues to pause 1, as we have assigned each exercise a move to pause, from there to exercise 3, but to know that it should go from pause 1 to exercise 3 and not to exercise for example, 10,12,14 so we put exercise3Checked and so he already knows that if we marked exercise 3 at the beginning, then he should go there. This way we had to set the conditions for all 17 exercises, so the Animator Controller looks more complicated than the Controllers for other categories, where there are only 5 exercises.

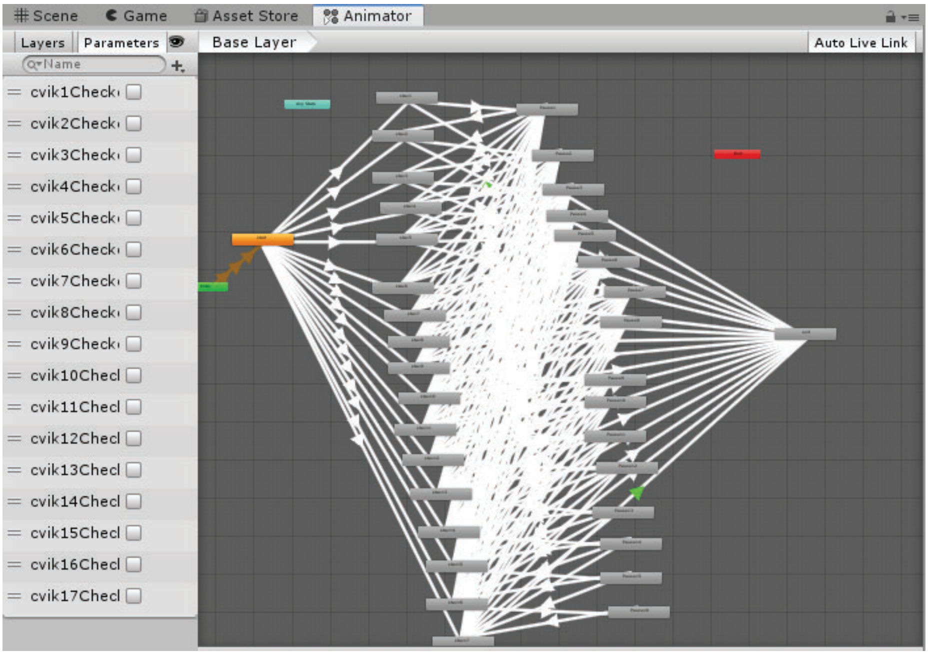
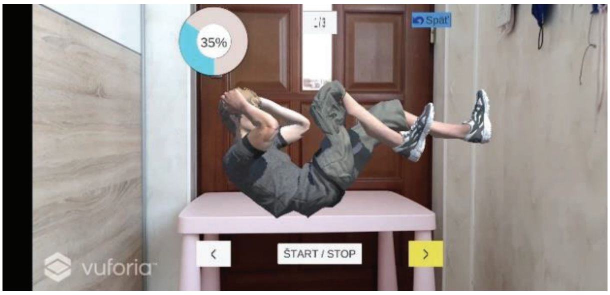
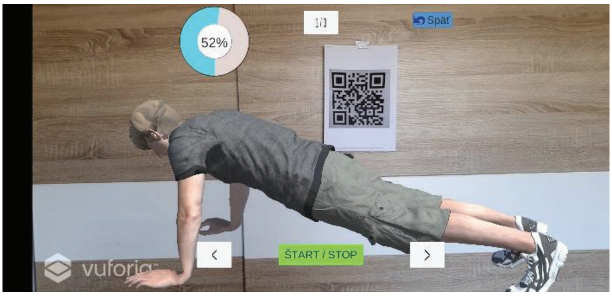


Fig. 3 : Animator for 17 exercises

2.4 Versions of application

Since the application is created on the Android platform, so we wanted to achieve its functionality in both mobile devices and AR glasses, so we used 2 solutions - different for each device. The mobile application works on the principle of finding the plane on which the user places the avatar. The Plane finder element works on the principle of displaying the viewfinder to the place it deems appropriate, and if it suits the user, it will place it there by clicking on this viewfinder. However, this solution did not work for us in AR glasses, which was caused by the fact that Ground Plane Stage is not supported in all devices and another problem would be that AR glasses do not have a mouse offer, and therefore it would not be possible to place it as in a mobile device. Therefore, we were looking for another available solution - and that is to display the avatar on the stage, not based on finding a plane, but based on an object that indicates the avatar display - in our case using the Image target element, which is described in the next section. Of course, the version for AR glasses is also usable for mobile devices. Another difference between the mobile version and the AR glasses version is that in AR glasses, the Exercise Descriptions scenes are solved by audio recordings, which means that they do not have to read text on the small glasses display that they can hear thanks to the built-in speaker. The recording starts automatically after clicking the Exercise Descriptions button. Adding a recording was to insert a clip into GameObject> Audio> AudioSource. This way we added audio files to each scene that required it.





On the first picture, we can see what the version of the application for Android mobile devices looks like, where an avatar is placed on the desktop using the Ground plane stage. On the second picture, we see the version of the application for AR glasses Vuzix M300XL, where we place the avatar using the Image target - QR code.

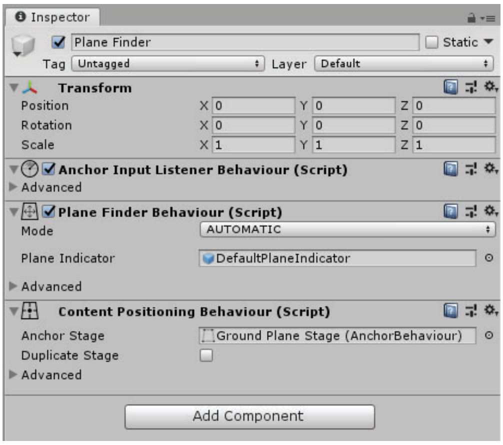
2.5 Image Target

We also had to add an Image Target to the scene because AR glasses do not have a mouse, so placing an avatar is not possible as in a mobile phone, where the user simply finds the area on which the avatar wants to see and clicks to place it. Therefore, we solved this issue through the Image target, which is actually the addition of an object to which the avatar will be linked. This means that if we run the application in AR glasses and look through the camera at a pre-placed object, the camera will recognize it and display the avatar at that point, which will start training. The avatar is tied to this item and the user can view it from any angle. We have decided that our Image target will be a QR code, generated based on the text. We created the given QR code online on the QR code generator, where it was enough to enter the text and choose the frame, color, shape of the QR code. We then downloaded it and added it to the project. In the application in AR glasses, it works by the user printing out this QR code and placing it according to where it suits him best, so that the avatar is displayed.

Fig. 4 : QR code

2.6 Ground Plane Stage

Ground Plane is supported for Android, iOS and UWP. It is only compatible with devices supported by platforms (ARKit / ARCore) or devices that have been specially calibrated by the Vuforia Engine. The latest device coverage is available online.
We have placed this element in the Unity project in the menu GameObject> Vuforia Engine> Ground Plane> Ground Plane Stage. This serves as a parent element to which we have assigned a "child" to our 3D object. Ground Plane Stage has a visual label in the Unity editor, where it is a grid of 100 x 100 cm, which is used to see the real placement of a 3D object on the plane itself. Then we place the Plane Finder via GameObject> Vuforia Engine> Ground Plane> Plane Finder. Here we can set the Anchor Input Listener Behavior - listens to user input (for example, by clicking on the device screen), Plane Finder Behavior - tries to find a suitable plane where he would be able to place a 3D object in the real world. To make the required features work properly, we've moved the Ground Plane Stage to the Anchor Stage field in the Content Positioning Behavior section.



Inspector

☑ Plane Finder □ Static

Tag: Untagged Layer: Default

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

☑ Anchor Input Listener Behaviour (Script)

▶ Advanced

☑ Plane Finder Behaviour (Script)

Mode: AUTOMATIC

Plane Indicator: DefaultPlaneIndicator

▶ Advanced

Content Positioning Behaviour (Script)

Anchor Stage: Ground Plane Stage (AnchorBehaviour)

Duplicate Stage: □

▶ Advanced

Fig. 5: Plane Finder

2.7 AR Smart glasses Vuzix M300XL

In order for the application created for Android to work properly not only on a mobile device, but especially on AR glasses, it was necessary to get acquainted with this device and functionality.

First, we had to charge the AR glasses, connect the main part to the computer with a USB cable, and in order to display it correctly on the computer, not just as an Android device, we had to mark USB mode in the Storage and USB section and select from four options File transfer. We needed this option to be able to move our application to the glasses. Since AR glasses support Android 6.0, we had to select this version in the Unity project itself in File> Build Settings> Player Settings> Player> Other settings under Identification to set the Minimum API level to Android 6.0 "Marshmallow" API level 23.

First we created a new project in Unity and for the correct functioning and display of the application in AR glasses we added the latest package Vuforia Engine AR to the Unity project. We downloaded the Vuforia Engine package in .zip format and added it manually via Package Manager. To add a package, we use the Add Package from Disk option in Unity, where we selected the downloaded package and opened the .json file. In this way we imported the Vuforia Engine with us, which we could then see added to the list of packages.

In the Build settings itself, in the File section, we chose the Android platform. The next step in creating the AR application was to replace the main camera with an AR camera. In the GameObject section, we chose Vuforia> AR Camera, which created the AR camera object in the scene.

Fig. 6 : Vuzix M300XL

2.8 Building application

Before building the application itself, several requirements need to be met. In the Edit menu> Project Settings> Player> Other Settings, it was necessary to mark the x86 architecture, necessary for compatibility with Vuzix M300 XL glasses in the Configuration> Target Architectures section. In Edit menu> Project Settings> Quality, we have set the default quality level for Android to Very Low.

We connected the glasses to the computer and moved the application to the AR Glasses Store. Launching the application from AR glasses starts in the menu Settings> Others and here we can find the application under a saved name. We made sure we set up the installation from unknown sources. We confirmed this in Settings> Security> Unknown sources.

3 Evaluation of results

Part of the assignment was to study the issue of Augmented reality and its use. In the first part, we described what the term Augmented Reality actually means, as not everyone knows it. Furthermore, the article describes where augmented reality is used, even though we may not be aware of it. First part of the article lists and approaches the SDK libraries that are needed to create augmented reality.

The second part deals with creating an AR application for Android. To create such an application, several tools were needed, which are described in the section Implementation of the solution. It describes the whole process of creating - from downloading the necessary tools before creating the application to moving the finished application to the device. For this type of reality, we needed to provide a device for which the application will work and on which we could test it. In our case we used AR glasses Vuzix M300. The functionality of these AR glasses and work with them is included in the article.

The theme of the application, where we use Augmented reality, is physical activity - exercise, which is becoming less and less nowadays.

The basis of the created application lies in trainings, where we see an avatar in the real world, which we place according to our own requirements and we can move around it, and thus the exercise becomes much more precise.

It should be mentioned that this application works in 2 versions - for mobile devices and AR glasses. In the glasses we display the avatar using an image on which the avatar will be anchored. On the contrary, we place the avatar on the mobile device based on the search for the plane where we want to place it.

The application itself consists of several categories that focus on a different part of the body. The user chooses, category, level if he wants to practice only one game, or he can also compose his Own training. In this category, he chooses only the exercises he wants to practice and does not have them fixed. The user also has the opportunity to see the individual exercises, what they are focused on and how they affect the body.

References

1. AZUMA R.T. : “A survey of augmented reality”, Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355–385, August 1997.
2. J.CH. Pomerol : Virtual Reality and Augmented Reality, Myths and Realities, edited by B. Arnaldi, P. Guitton, G. Moreau, ISBN 978-1-78630-105-5 , Online: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119341031.fmatter

This page is intentionally left blank.

An Approach to P300-Based BCI System

Jelena Kirić[1] and Radoslav Vargic[1]

[1] FEI STU in Bratislava, Ilkovičova 3, 81219 Bratislava, Slovakia
xkiric@stuba.sk

Abstract. The usage of brain-computer interface increases more and more over the years. It is used to control external devices, wheelchairs, robots, even for gaming. One of the applications of the EEG based BCI is for spelling device, in the first place intended for people disabled by amyotrophic lateral sclerosis. The aim of this research was to create the P300-based BCI system for spelling in Matlab environment. For the practical part of the research, the dataset of EEG signals of 5 subjects each spelling 5-character word was used. Proposed system uses features of P300 response, its physical appearance and the time interval needed for its appearance, for classifying the flashes in target and non-target groups. In the contribution we present several modifications of the basic algorithm and compare them with each other, and to the reference method.

Keywords: Brain-computer interface, EEG, P300 speller.

1 Introduction

A brain-computer interface is a system used to convert the brain activity signals into artificial output. BCI can replace, restore, supplement, enhance and improve the natural output of the central nervous system (CNS).

BCI systems can be separated into two categories: invasive, which requires the surgical procedure to implant the electrodes under the scalp of the user; and non-invasive, usually EEG-based, where the electrodes are placed on the scalp, mostly using an EEG cap. With help of BCI, using EEG activity, people can control external devices, robots, computers, etc.

The P300-based BCIs use the feature that unlikely event induces the P300 event-related potential (ERP) component in the signal. This BCI is typically used for spelling. P300 ERP appears in the EEG signal approximately 300ms after an attended stimulus.

In this study we have proposed the method for detecting the P300 response and then classifying the stimulus as target or non-target in speller BCI. The method is based on using the physical appearance of P300 response in EEG data to detect it.

This study is still in progress.

2 Dataset

The data used for this study were downloaded from BNCI Horizon 2020 [1]. These data were acquired by C. Guger et al. for research published in the article “How many people are able to control a P300-based brain–computer interface (BCI)?” [2]. The dataset consists of EEG signals of 5 subjects performing the visual P300 spelling task, and the label data.

The data were recorded while subjects were sitting in front of a laptop computer. On the screen was a 6x6 matrix of letters and numbers. The subjects were asked to spell the 5-character word using the BCI speller. For the training run subjects were spelling the word WATER, and for the testing they were spelling the word LUCAS, one letter at a time. Subjects were requested to focus on the letter in the matrix they were prompted to spell while entire rows and columns were alternately flashing.

For each subject data were saved in a structure containing two runs: one for training a classifier and another one for testing the classifier. These runs were saved in a form of a matrix of 11 rows, so called channels. First channel was a reference channel. The following 8 channels represented the 8 EEG channels i.e., electrodes. 10th channel contained the ID of flashed row/column. The columns were numbered from 1 to 6, 1 being the very left column; rows were numbered from 7 to 12, 7 being the very top row of the matrix. Channel number 11 contained the target information. It was set to 1 if a target row or column flashed and 0 if a nontarget was flashed (see Fig. 1).

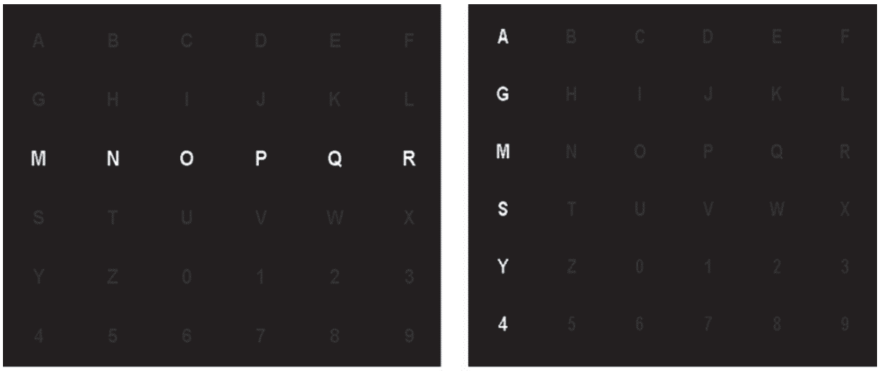


Fig. 1. The character matrix flashing the row with ID 9 (left) and column with ID 1 (right).

The EEG of the 8 channels were captured with 256 Hz sampling frequency. Then the data were converted to double precision, bandpass filtered between 0.5 and 30 Hz and down-sampled to 64 Hz. [2]

3 Proposed method

The method for data classification that we had proposed in this study is based on the physical appearance of P300 response in EEG signal. The method consists of setting the threshold on EEG signal and searching the EEG signal in borders of certain time interval after each stimulus. If the amplitude of the signal in that certain time interval crosses above the threshold, that stimulus would be classified as target, i.e., the expected stimulus.

The P300 response of each subject has his own unique appearance, ergo the classifier needed to be calibrated on training data separately for each subject, and then tested on testing data from the BNCI Horizon 2020 dataset.

This study is still in progress. It is planned for further study to use other physical features of the P300 along with its amplitude and time interval of the appearance, in order to find the best combination of the parameters for detection of the P300 response in EEG signal.

3.1 Signal preprocessing

To make the signal processing and classification easier, every signal of one run was fragmented into five trials, each representing an attempt to spell one letter, i.e., one run consisted of flashing a matrix rows and columns until the whole word was spelled. One trial consisted of flashing a matrix rows and columns until one letter was spelled (see Fig. 2).

The EEG signals of each subject distinctly differed in average amplitude value. To be able to use the same set of thresholds on every signal, we had to normalize the signals.

Our normalization algorithm calculated the average value of the samples in every trial of the signal that is being normalized (in the parts of a signal between two trials, before the first trial and after the last one, usually occur a lot of anomalies, so these samples were not considered (see Fig. 2)) and divided every sample of the whole signal.

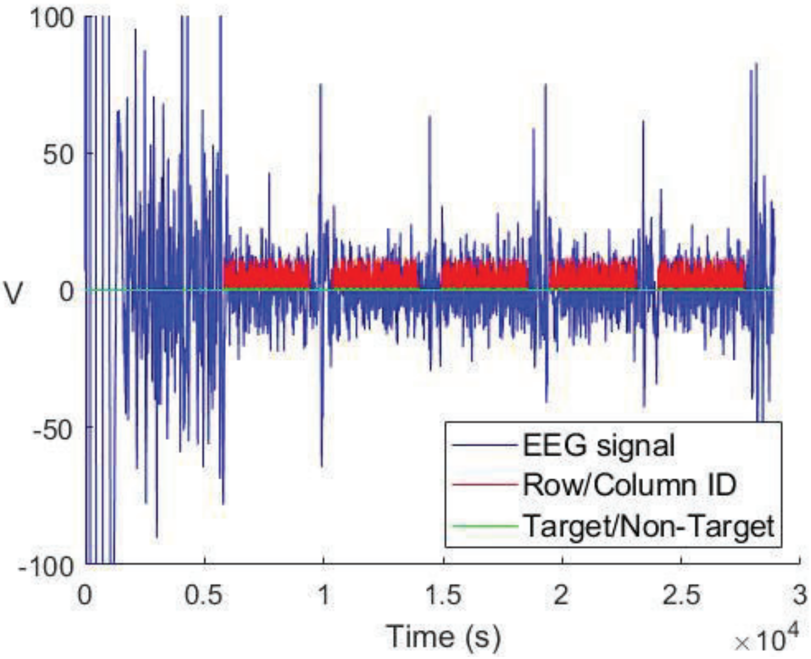


Fig. 2. Plot of one signal from one channel. Red samples represent the ID of the row/column that flashes at that moment. Each cluster of red samples is one trial. Also, the anomalies manifested by large amplitudes can be seen outside the trials.

3.2 Threshold manipulation

The set of thresholds from 0 to 5.5 with step of 0.25 was implemented on every signal from every electrode of each subject. After implementing each threshold, our classifier searched the extraction of the signal recorded after every stimulus. The extractions were defined by time interval parameter.

If the classifier had found a sample that had crossed the threshold in that extraction, that stimulus would have been classified as target, if it had not found any samples above the threshold, the stimulus would have been classified as non-target.

For each threshold implemented, the confusion matrixes were then made and used for calculating the accuracy of classifier. The threshold that was used in classifying that had the best accuracy was identified as the best for that electrode and that subject.

This manipulation with threshold was repeated with different time interval parameters which is explained in subsection 3.3, also it was repeated for each electrode channel to find which electrode gives the best results in detecting the P300 response.

Confusion matrix. To construct the confusion matrix, we had to create a true group and predicted group after every threshold manipulation. To create a true group, we have used the row/column ID channel, and target/non-target channel. The predicted group was created by our classifier in the way mentioned above.

The accuracy was calculated using the equation 1.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Deployment of these classes in confusion matrix can be seen in Fig. 3.

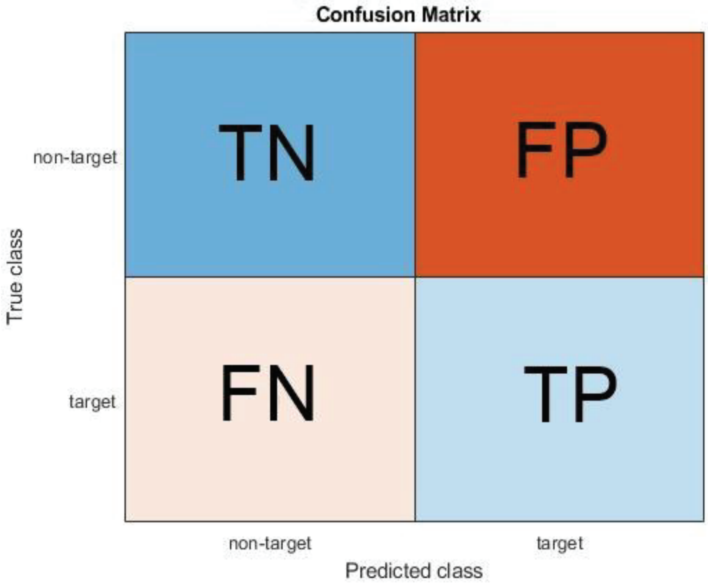


Fig. 3. Class deployment in confusion matrix, where TN is true negative, FP is false positive, FN is false negative, and TP is true positive.

In every trial the rows and columns flashed in total 180 times. 150 flashes were non-target, i.e., negatives, and 30 were target, i.e., positives. This disbalance in classes of confusion matrix would lead to distorted accuracy. For example, if we set the threshold so high that no sample would cross it, every stimulus would be classified as non-target (see Fig. 4). In this case the accuracy would be 83.3%, but no stimulus would be classified as target, therefore this classifier would be impractical despite the high accuracy. To avoid this occurrence, we had to balance the classes by multiplying the smaller class by 5.

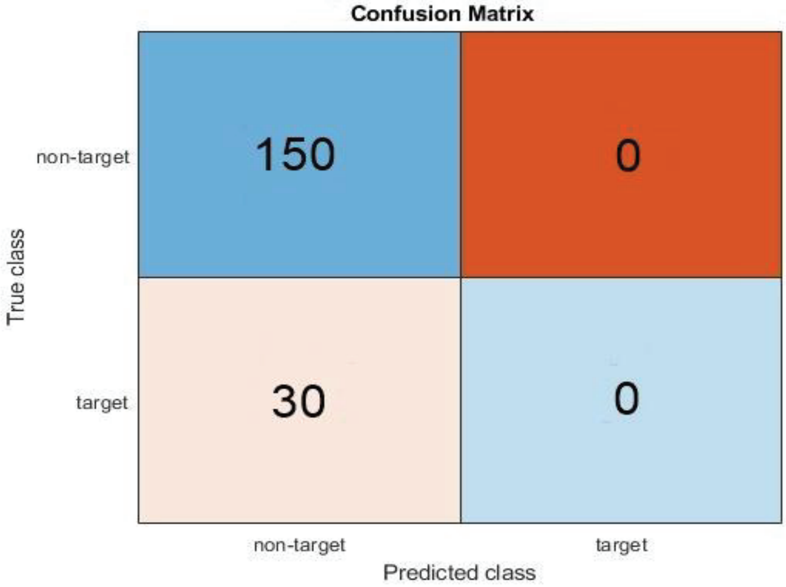


Fig. 4. Confusion matrix in case of threshold set too high.

3.3 Time interval manipulation

Since the P300 response occurs approximately 300 ms after the stimulus our classifier searched for the responses in time intervals around 300 ms after each stimulus. We have manipulated with 4 time intervals to find which one reaches the highest accuracy. The intervals that we have tested are:

- from 250 ms to 350 ms after the stimulus,
- from 200 ms to 400 ms after the stimulus,
- from 150 ms to 450 ms after the stimulus,
- from 100 ms to 500 ms after the stimulus.

3.4 Testing runs

Finally, when all the parameters that have reached the highest accuracies for each subject were collected on training data from the dataset, we have tested them on the testing data.

We have created a 6x6 matrix of zeros which was the score matrix and corresponded to character matrix on Fig. 1. Every time our classifier found a potential P300 response after a row/column flashed, that row/column was classified as target, and entire row or column scored 1 which was added to corresponding row/column of our matrix of scores. Index of the cell with the highest score pointed to the letter in character matrix that our classifier detected as the target letter.

Oftentimes multiple letters have the same highest score as shown in Fig. 5. In this case the correct letter that should be spelled was A, which score is in the top left corner, but the same score has the letter E.

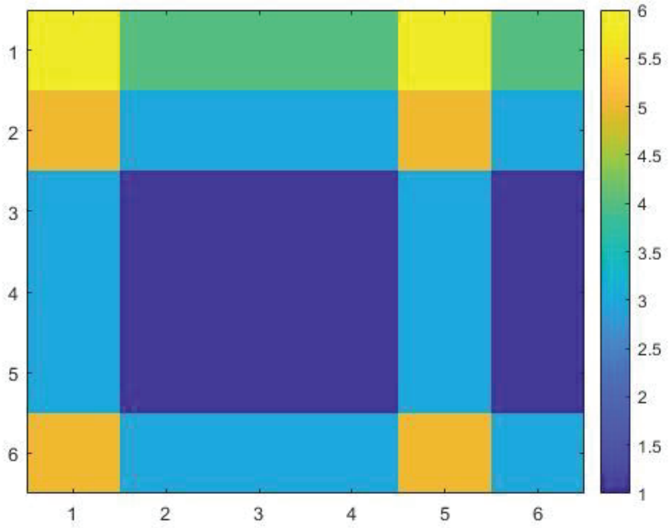


Fig. 5. Heat map of score matrix where two letters have the same highest score.

4 Evaluation

The best thresholds, and time intervals for creating our classifier for each of 5 subjects are listed in Table 1.

subject	threshold	time interval [ms]
1	2.5	150 - 450
2	2	250 - 350
3	2.25	100 - 500
4	2	250 - 350
5	1.75	250 - 350

Table 1. Parameters for classifier that reached the highest accuracies.

Two out of 5 subjects (40%) had the same best combination of threshold 2 and time interval between 250 and 350 ms. This time interval is the best for 60% of subjects.

The reference method proposed by C. Guger et al. reached the 72.8% of all subjects (81 subjects) that managed to spell the word with 100% accuracy, and only 1.2% did not managed to spell any of the letter correctly [2]. We have worked with only 5 of these 81 subjects and we do not know what accuracy they have reached with only these 5 subjects.

With our method we have managed to select correctly 12% of letters (60% of subjects had the spelling accuracy of 20%). It is crucial to mention that this study is still in progress, this method is going to be expanded with other parameters in order to find the best combination with threshold and time interval for detecting the P300 response which is expected to lead to higher accuracies.

5 Conclusion

The aim of this study was to propose the method for detecting the P300 response using its physical appearance traits and then classifying the stimulus as target or non-target in speller BCI. The method consisted of setting the threshold on EEG signal and searching the EEG signal in borders of certain time interval after each stimulus. The set of thresholds from 0 to 5.5 with step of 0.25 was implemented on every signal from every electrode of each subject. It was concluded that the best thresholds are around 2. The 60% of subjects reached best accuracies when the P300 response was searched in time interval from 250 ms to 350 ms. The data used for this study were acquired by C. Guger et al. [2]. C. Guger et al. tested their method on 81 subjects, we have tested our method on only 5 of 81 subjects. We do not know what accuracy they have reached with only these 5 subjects. With our method we have managed to select correctly 12% of letters. The plan for further study is to expend this method using other shape traits of P300, find the best combination of the parameters and use them in detection and classification.

6 Acknowledgment

The research in this paper has been supported by VEGA 1/0440/19, and the Excellent creative team project VirTel.

References

1. BNCI Horizon 2020, http://bnci-horizon-2020.eu, last accessed 2021/05/08.
2. Guger, C., Daban, S., Sellers, E., Holzner, C., Krausz, G., Carabalona, R., Gramatica, F., Edlinger, G.: How many people are able to control a P300-based brain–computer interface (BCI)?. Neuroscience Letters (2009).

A System for Local Multimedia Distribution

Oliver Palou[1], Michal Majdan[1], Radoslav Vargic[1]

[1]Slovak Technical university, Faculty of Electrical Engineering and Information Technology,
Ilkovicova 2, Bratislava 841 01
xpalou@stuba.sk

Abstract. In this contribution we present a novel system for local multimedia distribution. It allows distribution of streamed and pre-loaded multimedia. Distribution employs adaptive protocols as HLS and MPEG DASH with specific modifications. The system uniquely combines the media and informs about system state. The system is compared to similar existing systems.

Keywords: Multimedia distribution, Adaptive bitrate streaming, Adaptation

1 Introduction

A distributed multimedia system is an integrated communication and information system that enables the management, delivery, and presentation of synchronized multimedia information with quality-of-service guarantees. Recently, system for online multimedia distribution takes even more importance as the education goes online and is crucial to have online access to the study materials. In this contribution we closely examine widely used content management systems. We point out some shortcomings that based on our analysis we found essential or not efficient. In remaining part of section 1 we focus on widely used systems and their work with multimedia. In section 2 we discuss requirement for good content management system and in 3 chapter we will describe our technical work, chapter 4 is focusing on our testing result.

1.1 CMS

Content management system (CMS) is software that helps create and manage content on many type of client device with user-friendly interface, rather than needing to work directly with the code. When we are talking about CMS most people think about system that helps create and manage website (WordPress, Shopify, Joomla …) but there is another subcategory of software which help distributing media to widescreen devices in public places, this software is designed to just show content and have minimal or zero back way interaction with viewers. There is many type of software designed for this purpose but their function is bound to specific hardware requirements as operating system or brand. Their offer some basic features as playing video or showing some live information. In the rest of this section we summarize the representative solutions. Samsung MagicInfo – piece of software designed for smart TV from company Samsung Co. Ltd.. They offer many backend databases with various information (finance, news, flight info…). Technician can use this information optional or add own server. Also, there is implemented experimental artificial intelligence to collect data from microphone or cameras to improve targeted advertising[1]. SuperSign CMS – from company LG Electronics Inc, bound to products from the same name company. Their strong innovation is easy configuration on video walls, that are combine from small screens [2]. NEC Digital Signage software – need for implementation hardware accessory what is mount to smart TV and then offer decent functionality. Company bet on common features and don't inventing something new, just focusing to improve current state [3]. Sharp SDSS – from company Sharp choose other way, they design software for computers which are connected to screens. Software is still in development and offers many standard features of CMS [4].

1.2 Live Streaming Platforms

We have noticed that existing CMS system do not support live streaming content. In the next chapter we will go ever some platforms that focus on live streaming, but do not focus on the usage of television as end devices. There are many popular streaming platforms, that allow user to share content with other user. The platforms use similar technologies to handle multimedia. YouTube - is an online video platform owned by Google. The platform allows user to upload and live stream video content. Google embraces new technologies and developed video code VP9 and QUIC protocol [5]. Twitch - video live streaming service operated by Twitch Interactive, a subsidiary of Amazon. Twitch focuses on video game live streaming and broadcasting esports competitions. YouTube encodes media with VP9 and H.264 video codes. YouTube provided the content using combination of DASH and HLS. Media is distributed using HTTP/3 which uses the QUIC. The reason for mixing the protocols and codecs is that Apple devices do not support VP9 nor DASH [5]. Twitch encodes media in H.264 and provide HLS. Twitch distributes the content using HTTP/1.1. In the following section we will go over how these platforms, how they handle the ingress as “First Mile” and distribution of media as “Last Mile”.

First Mile. First mile is the traffic from the source to the server. There are two types of content, live and video on demand and both have their own requirements.
Live content requires low latency. There are multiple protocols, but RTMP is the most used protocol for live streaming. RTMP provided latency of 1-2 seconds. Popular software like OBS, allows users to stream using RTMP. Requirement for video on demand ingress are not demanding as live content and is usually handled by API.

Last Mile. Last mile is the traffic from the server to end user. The current standard are HTTP based protocols, as they are well suited to reach big audiences. HTTP Live Streaming and Dynamic Adaptive Streaming over HTTP are the most used HTTP based protocols. HTTP streaming also provide Adaptive Bitrate Streaming. Adaptive bit rate allows the viewer to dynamically receive the highest quality video that is based on the device capabilities and fluctuating network connection.

1.3 Storage considerations

Storage availability is a big concern, when storing multiple versions of the same content. Common media application format (CMAF) is an emerging standard intended to simplify delivery of HTTP-based streaming media. The advantage of using CMAF is the use of a single format of chunks. HLS and DASH share segments. The segments are split using DASH specification. Video and audio steams are split up into separate segments. This allows to match multiple different video bitrate with a single audio bitrate. This greatly reduces the storage requirements for storing a single video. [5] [6] [7]

2 Proposed solution

Purpose of content management system is delivering content to viewer. Content can be video, text information or live video. In our analyze of other systems we notice that every system relies on stabile internet connection and don't consider any internet outage. We try to design system that count on unstable connection by downloading content, when it is possible, to internal storage and play content form it. As there are many operating systems and we want to cover width range of screens we decide create our system on Android operating system that is used in some smart televisions and with hardware accessory we can used it on any device without dependency on operating system or non-smart screens. We are planning use this system in small or medium companies (schools, universities, …) that have maximum 100 screen units. Also, we are focusing not to limit the project only to television screens, but to also have a fully functional web interface for consumption and managing multimedia.

2.1 Server

Our propose for the server is a lightweight application integrating all necessary functionality into a simple deployable solution. To cover most types of end device and future proof we are planning to support HLS, DASH and mp4. Plan is to use HTTP/2 as it provides benefits over HTTP/1.1. HTTP/2.0 protocol decreases latency and allow to multiplex multiple requests over a single TCP connection. This is particularly useful as the end devices request the playlist and then a segment. We will not use HTTP/3, as support for it is less than a year and RFC is still in Internal Draft status as is not advised to use.

Transcode. We have analyzed different video codecs and quality of a video with resolution 1920x1080 using Video Multimethod Assessment Fusion. which is developed by Netflix for analyzing video quality. We have plotted the resulting data on Fig 1. and determined for our desired quality that the codec H.265 and VP9 provide better quality at lower bit rates than H.264. [8] [9]

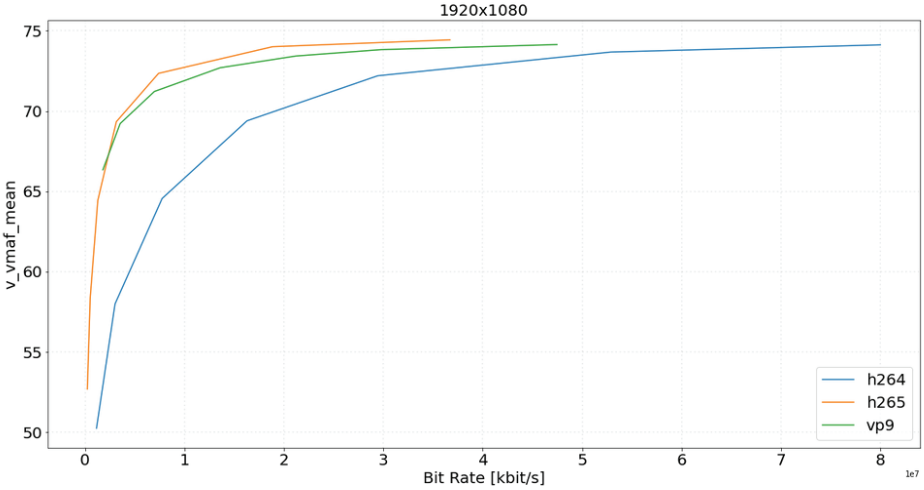


Fig. 1. VMAF quality graph containing video quality by bitrate using different video codecs measured on server.

We have also analyzed the transcoding time requirement. The time to transcode video with video codec H.264 is significantly lower than H.265 and VP9. The lower latency of delivering the content outweighs the benefits of lower bandwidth and lower storage requirements. The limitation of device support by choosing VP9 would increase the requirements for storage and transcoding. Therefore, we plan to encode live streams using video codec H.264. and video on demand content with H.265 video codec. [5]

Storage. As we would like to provide HLS and DASH, the concern of storage requirement rose. We have analyzed the benefit of using CMAF in our contribution. When HLS and DASH are sharing segments, the requirement for storage is lower by 50% [5].

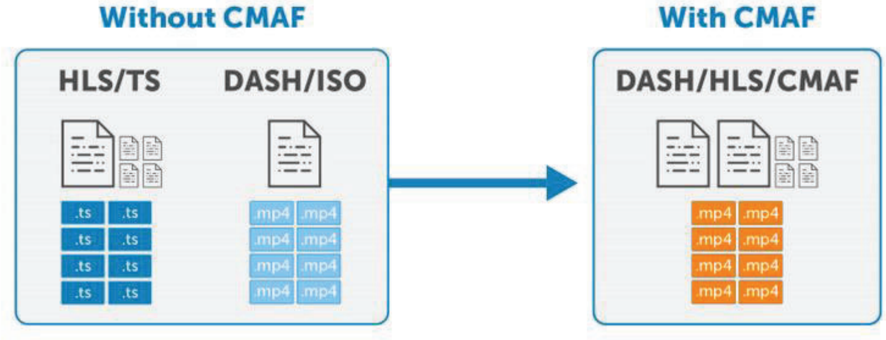


Fig. 2. Comparison of divided segments and CMAF shared segments

We have analyzed the potential of sharing audio segments between different video resolution. For analyzing purpose, we kept the same video quality across different resolution using Bit Per Pixel (1) value 0.05.

$$(\text{bitrate} * 1000) / (\text{width} * \text{height} * \text{fps}) = \text{BPP} \tag{1}$$

Further we have analyzed the percent proportion of audio in 10-minute video with 30 frames per second. On table below we can see the size of the different videos, audio bitrates and the audio percent of the total size.

Table 1. Storage calculation for different video resolutions

Resolution	Video Size	Audio Bitrate	Audio size	Audio
1920x1080	0.14 GB	144k	6.48 MB	4.42%
1280x720	62.208 MB	128k	5.76 MB	8.47%
854x480	27.6696 MB	96k	4.32 MB	13.50%
640x360	15.552 MB	48k	2.16 MB	12.20%

By sharing audio segments between video resolutions, we can lower the storage for audio by 8.292% and the total storage requirement by 6.737%.

3 Realization

The realization consists of server application with monitoring and big screen client (see Fig.3). The more detailed description of the components is provided in the next subsections.

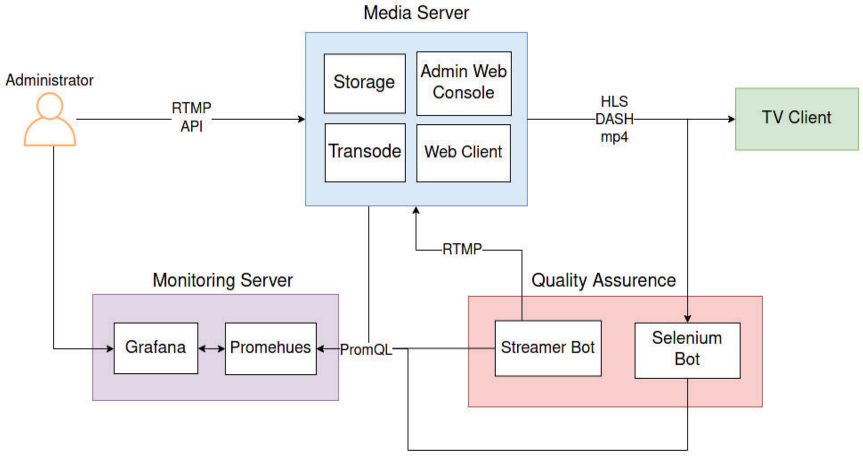


Fig. 3. High level overview of components in proposed system.

3.1 Server

Currently the server application is hosted on a virtual machine that has 1 virtual CPU, 2 GB Rams assigned. The application has small overhead distributing multimedia and depending on the configuration of transcoding it varies. The codebase is easily manageable as it is monolithic. The whole codebase spans across 150 source code files and totaling under 15000 lines of code.

The backend is written in Node.js framework, which is a JavaScript runtime environment in V8 engine. The first mile is following the current standard of RTMP as ingress. The RTMP server is able to accept video codecs H.264 and audio codec AAC. For the "Last mile" delivery, HTTP based protocols were chosen as they can be deliver using a regular web server. The server can serve video content in HLS, DASH and mp4. Mp4 support was integrated for devices that do not support ABR or are designed to cache the content and play it locally. We encode the live content using H.264 and VOD using H.265. Distribution of HTTP based protocol is handled by the web server supporting HTTP/2.0 with HTTP/1.1 fallback. For storing data MongoDB database was chosen, which is a non-relational database. Library mongoose is used, which allows structured schemes for data.

The admin console is written in React.js Framework. Administrator can manage content. We have also implemented a knowledge base, that contains documented source code and information about used technologies in the project. Monitoring of the server application is implemented with a custom metrics exporter and the data is formatted in Prometheus Query Language. Monitoring software Prometheus is used to collect the data, then data is visualized using Grafana. We have implemented alerting of events, like outages or any unexpected errors. These alerts are sent to the administrator using webhooks. Quality assurance monitoring for live stream is implemented with a program written in Python and Selenium framework. The application visits a test live stream in web client and collects metrics like page and video load time and is checking if the video has any playback issues. The test stream is streamed by an application written in Node.js to the server. Both applications have custom metrics endpoints, that Prometheus is collecting data.

3.2 Client

System is primary oriented for presentation and advertising. We focused to deliver multimedia content to watcher via widescreen displays, which are deployed in public places like hallways. For low-cost version of our project, we designed whole system on existing hardware as smart television which are already installed. Application for smart televisions is designed for android operating system. Choosing android is best option because we eliminate differences on many operating system which was restrictive. Designing application for any possible system will bring more problems as solutions, so we agree that android is our choice. When we want deploy system on device what has different operating system we use low-cost hardware extension as android TV boxes, which will be connected via display port to our screen. System consists of few smaller features as playing videos from server, ability to play ABR streams as HLS and Mpeg-Dash or displaying text information in one line on the top of screen.

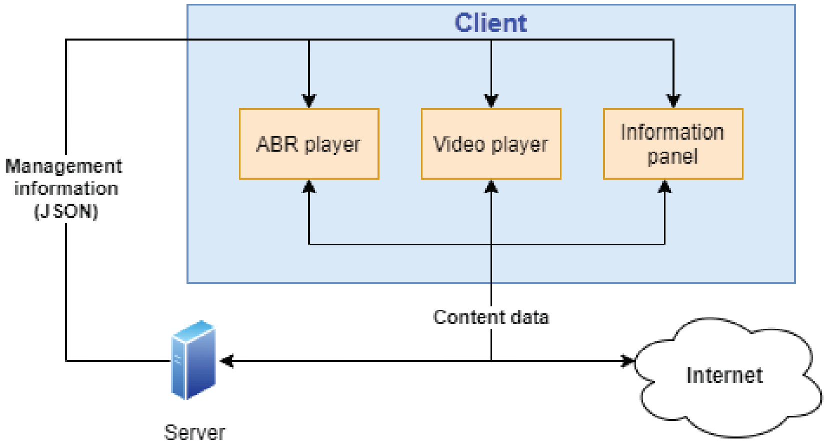


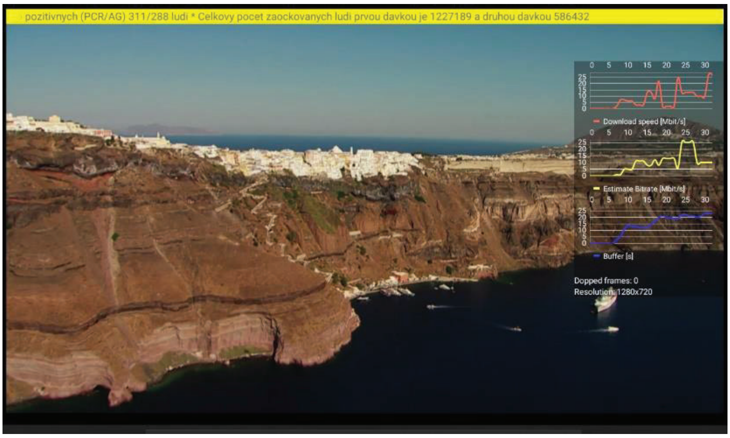
Fig. 4. High level overview of components in client

Video player. We determined our goal to give watcher best QoS (Quality of Services) as it is possible by suppress unstable internet connection. For playing video from server is unnecessary load for internet connection we choose to download all videos to internal storage once and then playing it form this storage. By this step we are no longer relying on reliability of connection. Client periodically updating list of videos and when new video is available it will be download.

ABR player. In live stream is not possible download whole video at once, so we chose adaptive bit rate streaming which give us ability adapt on existing connection.
QoS in ABR streams have mainly 2 big indicators:

- Resolution, when higher means better QoS
- How many times video switch to higher or lower resolution, smaller number means better QoS

For improving QoS on stream we do some changes in player algorithm, special in policy what choose buffering resolution. When stream is playing, player load few seconds of stream in buffer and in right time it is displaying them on screen. This cause little delay in every stream and watcher didn't notice anything, but for player this feature is critical, because he can eliminate some packet drops on connection. Level of buffered seconds indicate if connection is good or poor. If buffer contain more seconds as he need right now, that means for him that he can start loading video of better quality with bigger size.



pozitivnych (PCR/AG) 311/288 ludi * Celkovy pocet zaockovanych ludi prvou davkou je 1227189 a druhou davkou 586432

Dopped frames: 0
Resolution: 1280x720

Fig. 5. Example sceen displaying the player during playing video stream

We add statistic, in graph, for our stream, where viewer can monitor actual status of stream and see behavior of stream, to better understand why stream lower or increase quality.

In our project we edit buffering rule by changing thresholds when video will change loading resolution from server. Because of absence of interaction from viewer we significantly improve time to take when player change quality to higher and did not change other parameters as number of switches to other quality or video freeze.

We analyze algorithm for buffer loading and change limit for minimal level of buffered seconds in buffer needed to increase quality and maximal buffered second to decrease stream quality. We create 3 groups of setting as can be seen form Table 2.

Table 2. Used groups of settings

	Minimal level for increase quality [s]	Maximal level for decrease quality [s]
Group 1	**5**	**12,5**
Group 2	**10**	**25**
Group 3	**20**	**50**

Then we test this setting on different internet connection, where we simulate internet shaping and packet loss on 15%. For out testing we used emulator for connection parameters (SoftPerfect Connection Emulator), and it helps us simulate different parameters of line (shaping, latency, packet loss, duplication, etc.). Shaping policy is buffer based. When traffic exceed line shape, packets above will be stored in queue and send when it is possible. This queue is limited to 10kbit of packet, so when happened situation that queue is full, everything above will be dropped. In our first phase of testing we simulate also latency on line because this parameter have no big impact on final solution of our experiment [10].

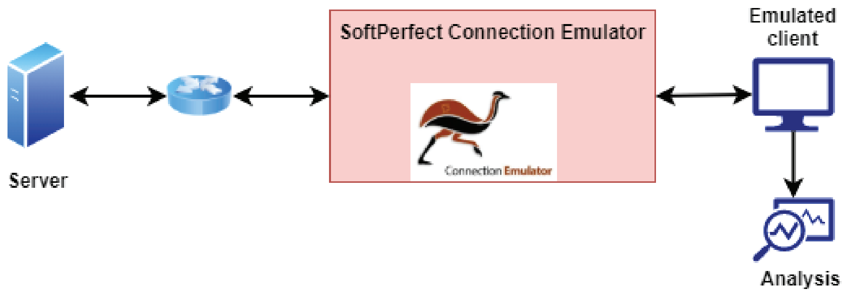


Fig. 6. Overview of simulation and testing

4 Evaluation (Results)

After testing, we notice on results when we use lower values of thresholds we can achieve same internet resilience as stream with default settings (Group 2) but higher quality of stream will be played earlier than in others settings. We combinate many internet parameters to simulate good and bad internet conditions and in every scenario we have same result. When we averaged all result, player can achieve highest possible resolution in 7,66 seconds earlier then in others. We can say that this is success of our work when we can elevate quality of experience for viewer.

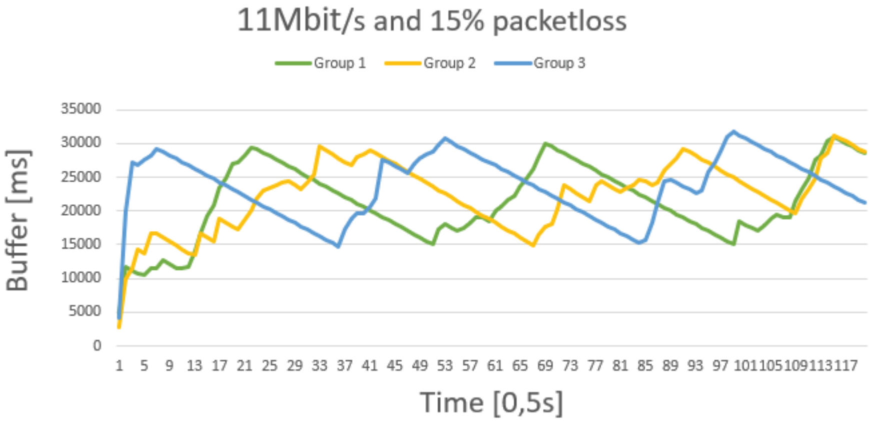


Fig. 7. Graph of buffer level state

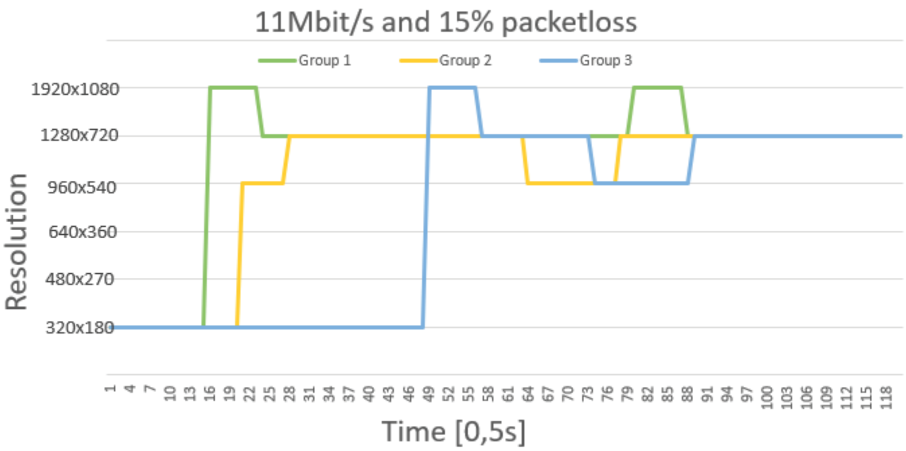


Fig. 8. Graph of played resolution in time

5 Conclusion

We design system that offer basic features as any other commercial CMS with some innovative technologies as downloading video to internal storage or modified ABR stream. We can just discuss if this innovation is new step forwards in evolution CMS or step back because many much lagers companies and institutions did they own research and this type of delivering multimedia can be for them unsatisfactory. From our perspective, own research and testing we can say that system can be start using in small and medium companies as tool for distributing multimedia and distance learning.

Acknowledgment

Research described in the paper was financially supported by the 2020-1-CZ01-KA226-VET-094346 DiT4LL project and the Excellent creative team project VirTel.

References

1. Samsung, Electronics, "displaysolutions.samsung.com," [Online]. Available: http://displaysolutions.samsung.com/docs/pages/viewpage.action?pageId=2064846&preview=/2064846/7376093/MagicInfo_Server_UM_Eng_Rev.1.2_160718.pdf. [Accessed 2021].
2. LG Electronics Inc., "lg.com," [Online]. Available: https://www.lg.com/us/business/display-solutions/supersign-w-lite/downloads/LG_SuperSign_CMS_v_2_9_Eng.pdf. [Accessed 2021].

3. NEC Display Solutions, "sharpnecdisplays.eu," 2018. [Online]. Available: https://www.sharpnecdisplays.eu/p/download/v/5803ab139e14e305f3f5414e6542c492/cp/Products/Shared/Brochures/Brochures_Options/SignagePlayer-Guide/NEC_Buyers_Guide_May2018.pdf?fn=Digital-Signage-Player-Guide.pdf. [Accessed 2021].
4. Sharp, "global.sharp," 12 1 2021. [Online]. Available: https://global.sharp/products/img/professional-monitors/software/SDSS_CTLG_S.pdf.
5. "Google Developers," [Online]. Available: https://developers.google.com/media/vp9/settings/vod/. [Accessed May 2021].
6. E. M. Bishop, "QUIC - Internet-Draft Version 34," 2 February 2021. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-quic-http.
7. "Wowza," [Online]. Available: https://www.wowza.com/blog/what-is-cmaf. [Accessed May 2021].
8. "FFmpeg," [Online]. Available: https://trac.ffmpeg.org/wiki/. [Accessed May 2021].
9. "VMAF," Netflix, [Online]. Available: http://github.com/Netflix/vmaf.
10. "Connection emulator," SoftPerfect network management solutions, [Online]. Available: https://www.softperfect.com/products/connectionemulator/. [Accessed May 2021].

This page is intentionally left blank.

HALO Hologram Software

Denis Džačko[1] Samuel Polakovič[2] Filip Suchán[3] Erik Kaľavský[4] ,
Gregor Rozinaj[5]

[1]Faculty of Electrical Engineering and Information Technologies,
Institute of Multimedia Information and Communication Technologies,
Ilkovicova 3, 812 19 Bratislava, Slovakia

kalavsky.erik6@gmail.com

Abstract. Today's communication systems offer limited possibilities of image processing and subsequent its presentation. Most systems offer an image that is presented only in 2D. With a 2D image, the third dimension disappears, which is the depth of the image. In this article, we deal with a system with the ability to transform an image from 2D to 3D while maintaining in-depth information. This creates a 3D impression and improves the quality of communication, because we create the impression that the person we are communicating with is in the room with us, even if this is not true. Such a system requires real-time hardware and software implementation. In our case, we deal with both hardware implementation and software solution in real time.

Keywords: Atmel, connection, LED strip, HALL-effect sensor, small rotate engine, USB, Graphic user interface, stream, IP camera

1 Introduction

Today, information technology [1] is increasingly used in various fields and on various occasions. These technologies are especially used between people for audio or video communication. This communication is largely restrictive and insufficient, due to the transmission of only the audio signal during telephone calls or the transmission of only the 2D image during video transmission.

In video communications, the depth of the image is lost, giving the impression that the communication is taking place with a moving 2D image on the screen. In order to achieve the perception that the person with whom we communicate through information devices is next to us, in-depth information is one of the most important parameters. To preserve this information, various algorithms are created [2] how to preserve this information and then present it in order to create an image with a 3D effect.

In this work we will focus on how to create such an image, then process it, distribute it over the Internet, and additionally processing such an image where the result will then be presented in 3D.

2 BASIC PROPOSAL

1. Our hardware design is based on the Halo hologram concept [3]. Halogram is a tool for displaying an image that appears as a 3D image. Such an image gives us the

impression that the person with whom he communicates via video communication is next to us, even though he is not in the room with us.

In addition to the hardware design, software image processing is required to make the created image as realistic as possible.

The design consists of two main parts and these are the software implementation and the hardware implementation. Figure 1 shows the basic design concept.

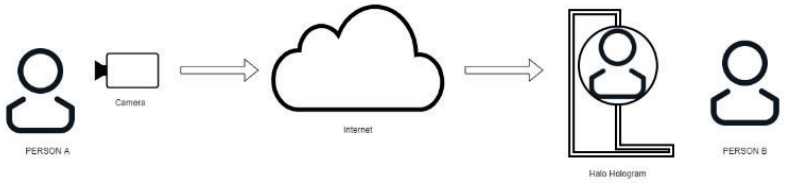


Figure 1 Basic scheme

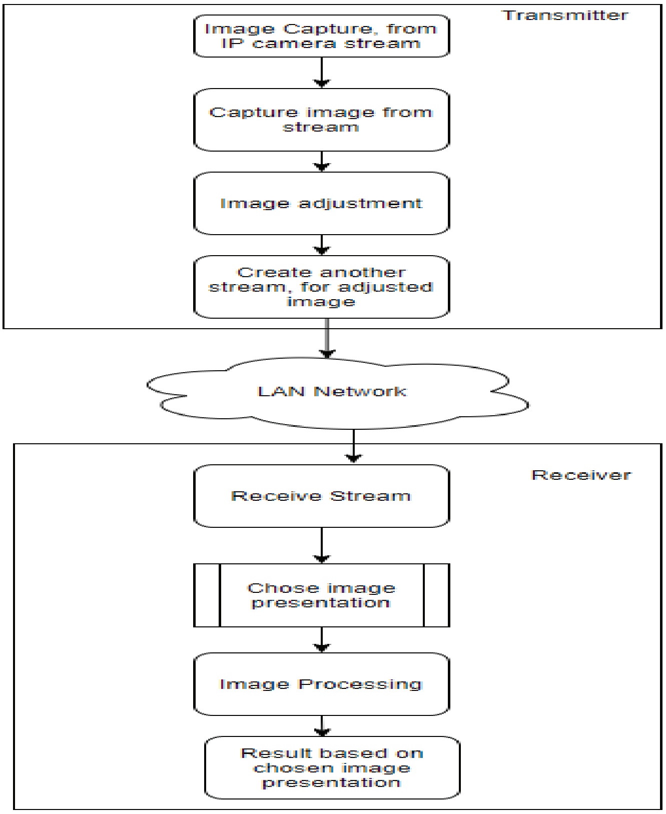

Transmitter
Image Capture, from IP camera stream
Capture image from stream
Image adjustment
Create another stream, for adjusted image
LAN Network
Receiver
Receive Stream
Chose image presentation
Image Processing
Result based on chosen image presentation

Figure 2Elementary blocks of realization

We will analyze the individual blocks in Figure 2 in more detail and approach the principle of operation.

3 Transmitter

In this section, we discuss the methods by which an image can be captured, subsequently edited, and sent over a LAN to a receiver. This part is software implemented in the C ++ programming language, which has been extended with a library for OpenCv image processing and also with a library for G-Streamer streaming.

1. Image capture

For image capture, we chose an IP camera that can be easily modified and customized as required. The main requirements are the size of the scanned image and the frame rate per second. Another advantage of IP cameras is that it creates its own stream of transmission that is easy to capture and then work with the captured screens.

We are also able to use a WEB camera, which is part of every computer, instead of an IP camera.

2. Capture the image stream

In this part we capture individual image frames, whether they are from an IP camera or a WEB camera in real time.

3. Image adjustment

In this block, the individual captured image frames are resized to the required size so that they can be subsequently sent over the network

.4.Create a new stream with the edited image

After capturing and editing each frame in real time, we will create a new separate stream of such an image. The new stream is created via TCP confection on a server with a dedicated free port. In our case, we work in one closed LAN network, so the IP address of the local computer serves as a server.

Such a stream can also be captured by third-party applications after entering the correct URL of the line on which the stream is presented. The newly created stream has a delay of approximately 2-3 seconds

4 Reciever

The main functionality of the software solution is in the receiver because, here the choice is made onwhich device the given image will be presented.

4.1 Receiving the stream from the transmitter

In this section it is necessary to enter the required parameters for reception and this is the IP address of the server and also the port on which the stream will be received. Subsequently, a URL address is generated based on the selected parameters for receiving the stream from the transmitter.

4.2 Display selection

This part serves in what form the image should be displayed. In our software implementation, we have more options than the resulting image can be displayed. Since some parts are still in development, we decided to point out two bases that are implemented.

4.3 Pyramid image display

In this view, the image will be divided into four quadrants as shown in Figure 3, with the same image being displayed in each quadrant.

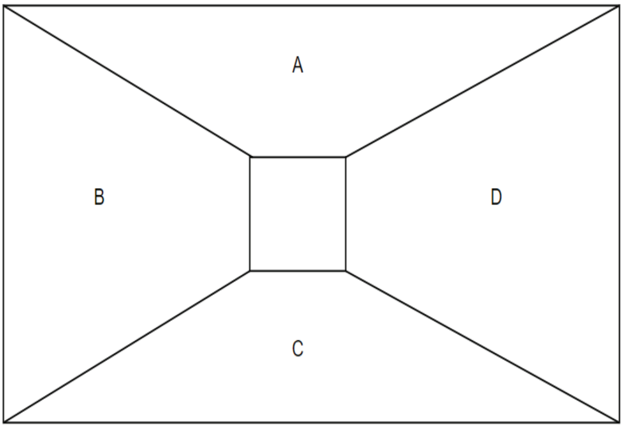


Figure 3split of screen A,B,C,D quadrant

Subsequently, in the middle of the placement, the pyramid that will reflect the image will be created, adding that the image will float inside. The image on each side will be the same. The location of such a pyramid is shown in Figure 4.

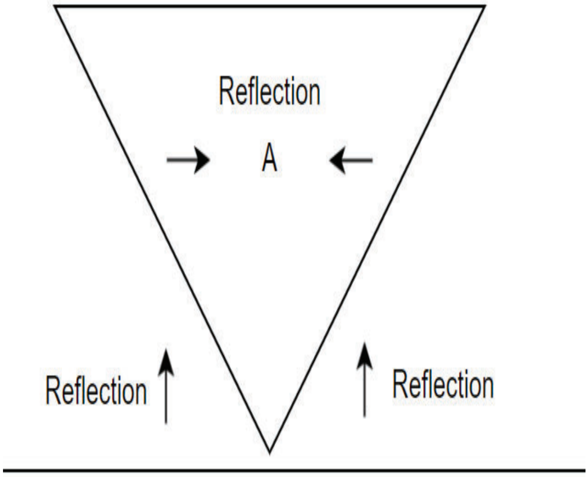


Figure 4Display mirrored image

4.4 Display on Halo hologram

The display on the globe model will be described in more detail in Chapter V Hardware Implementation

4.5 Image processing

After selecting the display, real-time image processing then takes place. In this section, each farm received is adjusted according to the selected display option. Each display option requires specific image adjustments.

4.6 Image display based on the selected method

After receiving and processing the image, the resulting image is then presented on a display device. In the resulting image, the depth of the image is preserved and a 3D impression is created.

5 Software implementation of the Hardware part

To display the image in the best possible quality, we built a circle that is seated in the base. The display of a given image is created by a rapid rotation of a circle around its own axis. On the circle are the location of the LED strip that creates the image. The hardware implementation can be divided into three parts, which are the transfer of data from the receiver to the microcontroller, editing the received data and then sending them to the display unit, displaying the data on the built circle. These parts are shown in Figure 5.

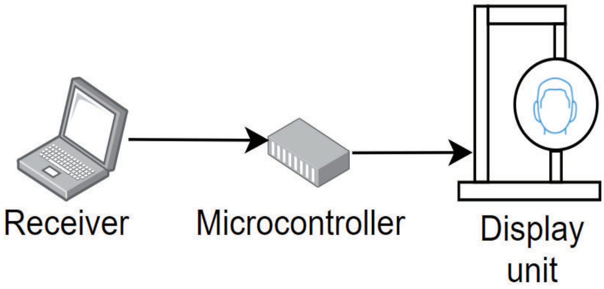


Figure 5Elementarry scheme of hardware realization

5.1 Data transfer from the receiver to the microcontroller

After image processing on the receiver side, one frame is then sent via the serial USB connector to the motherboard of the Atmel UC3-A3 Xplain microcontroller.

After establishing a connection between the microcontroller and the receiver, each frame of the received image is then sent to the microcontroller in blocks. One block is a 120x65 nut.

For one such picture frame it is necessary to send 3 picture blocks. This is because the resulting image contains all 3 RGB color components and each component is sent separately. The transmission speed is 125000 Kbaud, which is not enough speed for the given image to be received in a sufficiently short time for its further processing. We could not overcome this speed.

5.2 Editing the received data and then sending it to the display unit

This part is the most important part of the hardware implementation, even though it is performed on the microcontroller itself. Due to the control of the display unit itself. After receiving the input data, a 120x65x3 matrix is created, which is the resulting image matrix, which will then be projected on the display unit.

The resulting image matrix is then sent via the SPI interface. Sending speed is 30Mhz / 1s. The resulting array is sent in blocks of 4 bytes. A 32-bit start packet is sent to establish a connection with the LED strip. Subsequently, the matrix is sent one pixel at a time. Since the size of the sent block is 32 bits, the first 8 bits indicate the basic properties so that the first 3 bits are controllable, 5 bits indicate the intensity at which the LED on the LED strip should light up. The intensity range of the LED strip is from 0 - 31, in our solution we use the maximum intensity of 2 because we use a very bright LED strip. And the remaining 24 bits represent information about the pixel itself in RGB.

The main feature of the display unit is the rotation of the circle. Timers are used for the correct rotation of the circle, which will ensure synchronization based on the frequency of rotation.

The display itself is by means of a slice counter strip which has a timer set to 120 seconds. Subsequently, the resulting image matrix is read from the beginning to the end and also at the same time from the end to the beginning. This is so that the resulting image is on both sides of the display unit.

5.3 Display unit

The display unit is constructed of commonly available material. This display unit is shown in Figure 6. In which the image is projected.

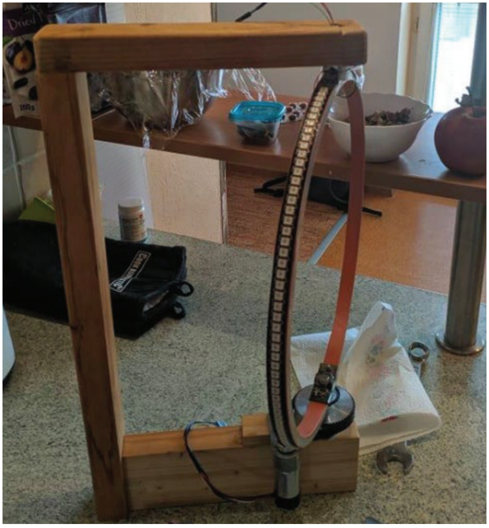


Figure 6Display Unit

Due to the limited transmission speed between the receiver and the microcontroller, the resulting image is insufficient and incomplete. Therefore, the main functionality was tested on a predefined still image matrix which was then sent to the display unit.

6 System improvement options

To achieve better results, it is necessary to increase the baud rate for sending data between the receiver and the microcontroller. The proposed speed is 3000000 Kbaudov / 1s. At this speed, the resulting image should be high quality without imperfections.

Another possibility to achieve better results is to use another microcontroller such as RasberryPi or possibly Arduino. These microcontrollers are able to achieve higher speeds and also have a higher memory capacity.

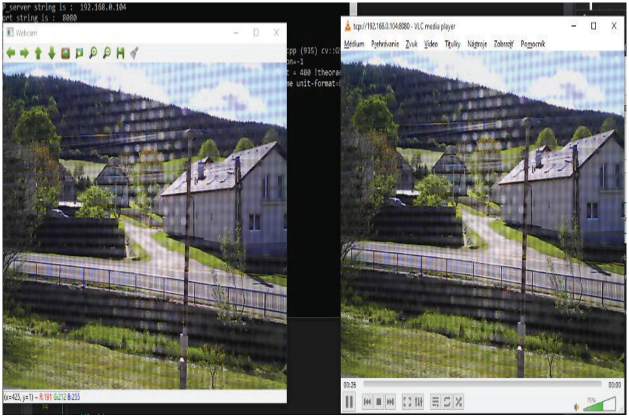
It is also possible to improve the system from a network point of view by adding a publicly available server and communication will not be limited to the LA network.

7 Result

The system can obtain a stream from an IP camera and then, after editing and selected options, present the image on display units.

7.1 Receive and create stream

As can be seen in Figure 7, we obtained the stream from the IP camera and then distributed it over the LAN and displayed it in a third-party application.



```
P_server string is :  192.168.0.104
ort string is :  8080
cpp (935) cv::G
on=-1
t = 480 !theora
me unit-format=
```

Figure 7Obtain a stream from the IP camera on the right, and then create a new stream on the left

7.2 Displayed in pyramide

After selecting the pyramid display option, one frame is divided into 4 identical quadrants as shown in Figure 8.

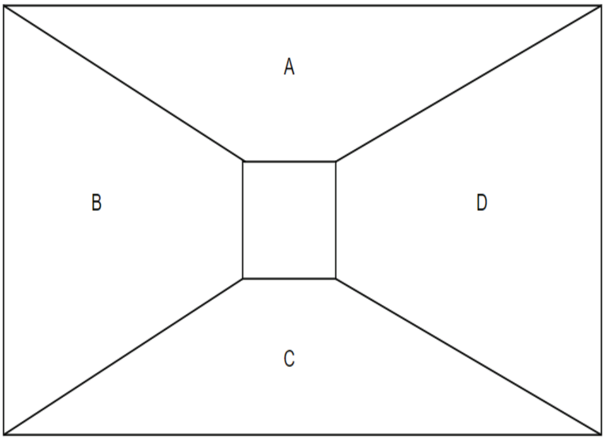


Figure 8Image obtained from an IP camera and then ready for pyramid display

The image thus prepared is then streamed and can be captured via a third-party application on the mobile. Such a representation is shown in Figure 9, with each page presenting the same image. If necessary, it is possible to present a different image on each page

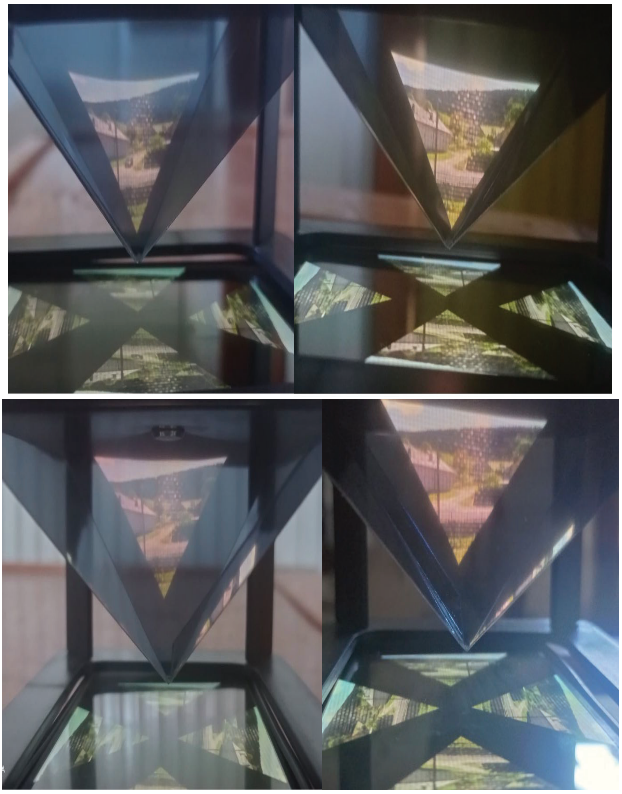
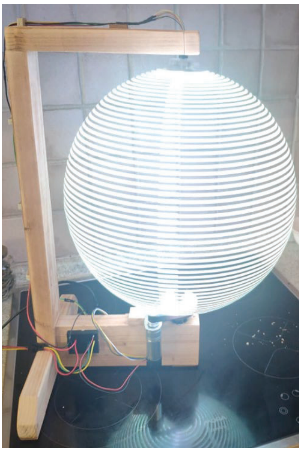


Figure 9age in holographical pyramide from all sides

7.2 Display on rotating circle

In this section, we were the first to test the functionality of this display unit. In Figure 10 and Figure 11 it is possible to see the testing of the functionality on one color and subsequently on two colors. This confirms that the display unit is working properly..



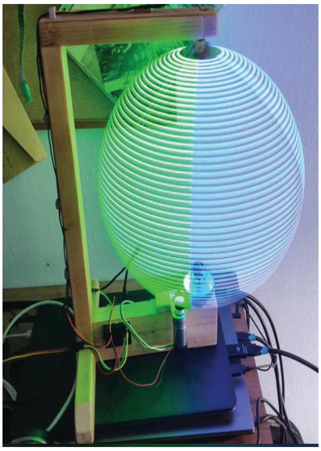


Figure 10. Right figure testing one color on display unit, left figure testing with 2 colors.

After verifying the functionality, we tested a predefined static image, which was then sent to a rotating circle. The main problem in this part was timing because we used Halo sensors that were not enough and created delays. Due to the delay, the image was spread over the entire circle as can be seen in Figure 11.

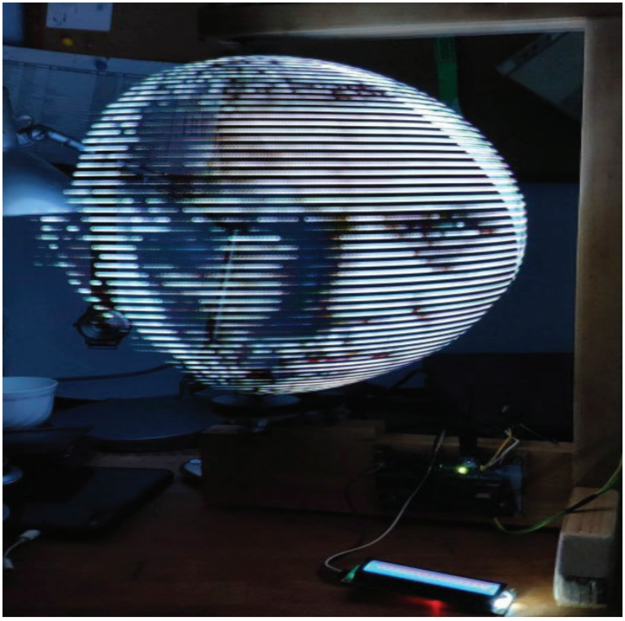


Figure 11 Face display during time synchronization with Halo sensors.

Until a stabilized image is achieved, we then tried to project the human head, which should be the main result of this system. This result can be seen in Figure 13 and also in Figure 14.

Subsequently, we used timers as functions instead of Halo Sensors. In this way, we were able to stabilize the desired image and direct it directly to the display circle.

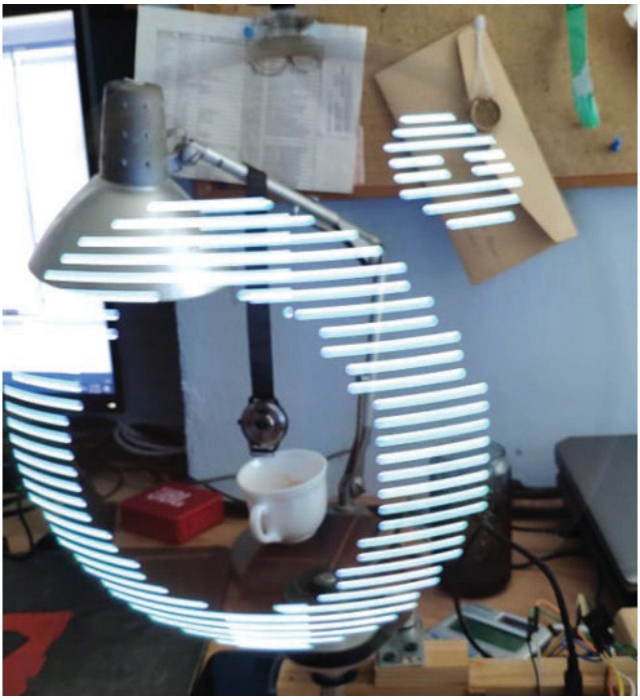


Figure 12Stabilized image display

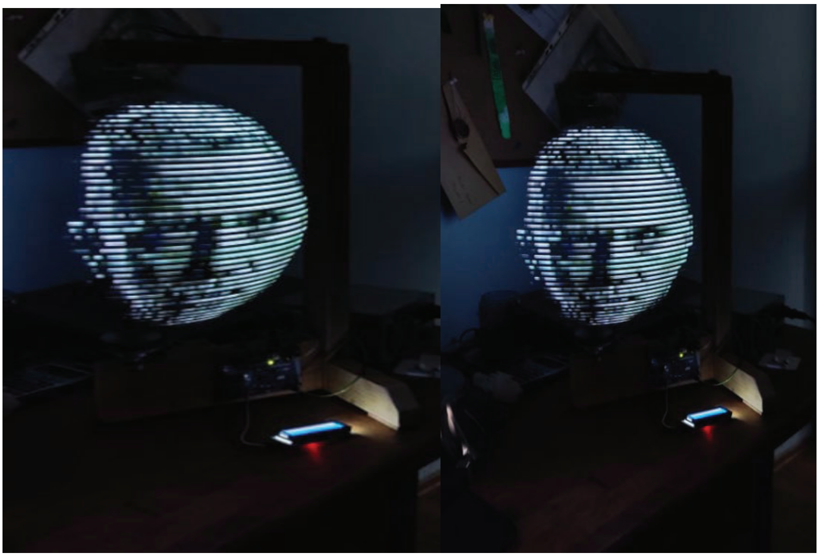


Figure 13Final Display of face

cameras, communication within one VLAN network. We also managed to achieve results in the field of imaging, such that we can display an image from memory for our Halo hologram. Of course, in conclusion, we would like to add that even though we have achieved many tangible results in our project, there is room for further development and improvement. As some, we would like to mention image processing from scanning devices such as Azure Kinect, Data transfer not only in one network but globally. Next, it is necessary to solve the problem of loading speed of sent data for display on the hologram. The greatest room for improvement occurs in the display where it is necessary to further work on the display of not only the stationary image but the real time video image itself to create quality communication technology of the future.

Acknowledgment

This publication was created thanks to support under the Operational Program Integrated Infrastructure for the project: International Center of Excellence for Research on Intelligent and Secure Information and Communication Technologies and Systems - II. stage, ITMS code: 313021W404, co-financed by the European Regional Development Fund

References

[1] EGBU, Charles O.; BOTTERILL, Katherine. Information technologies for knowledge management: their usage and effectiveness. Journal of Information Technology in Construction (ITcon), 2003, 7.8: 125-137.

[2] FEHN, Christoph. A 3D-TV approach using depth-image-based rendering (DIBR). In: Proc. of VIIP. 2003.

[3] SUCHÁN, Denis Džačko1 Samuel Polakovič2 Filip; KAĽAVSKÝ, Erik; ROZINAJ, Gregor. HALO HOLOGRAM HARDWARE. 2021.

[5] BARANNIK, Vladimir, et al. The video stream encoding method in infocommunication systems. In: 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET). IEEE, 2018. p. 538-541.

HALO Hologram Hardware

Denis Džačko, Samuel Polakovič, Filip Suchán, Erik Kaľavský, Gregor Rozinaj

Faculty of Electrical Engineering and Information Technologies,
Institute of Multimedia Information and Communication Technologies,
Ilkovicova 3, 812 19 Bratislava, Slovakia

kalavsky.erik6@gmail.com

Abstract. Our task, which we focused on, is to design and create a device that would be able to display the image of the called party in the room with the appropriate software and give us the impression that the called party is in the room with us for better communication.

Keywords: Atmel, wood, Components, LED strip, HALL-effect sensor, small rotate engine

1 Introduction

Since the beginning of time, mankind has always cared about social contact and mutual communication. This is also the case today, when we are living in times full of modern technology and the age of social networks. However, times have changed and people are more preoccupied and often do not have time to meet in person. So how to improve mutual communication and the enjoyment of this communication for people? In the current pandemic situation, it is even more complicated and therefore the topic of communication with the use of hologram is much more relevant than ever before.

2 Ideas Of Holograms And Our Choose

Firstly, it is necessary to specify task itself, and what kind of scene we are expecting. Under the term halo hologram, we can imagine a device that will display a person face. This device could be used in organizing of corporate meetings or in the educational process of teaching lectures and conferences. At the beginning it was important to think about the very concept of Halo hologram. There are many possibilities of realizing hologram which we had successively analysed and evaluated.

First concept of the hologram was creating holographic pyramid in which the hologram itself would be projected. Holographic pyramid works on the principle of composing 4 projected images together. This solution is suitable for small meetings, but not for large conferences as it would be necessary to create wide- ranging holographic pyramids for large lecture halls.

The second option would be to create a projection bust. The bust would be placed in front of the black screen/ background and would be equipped with motors for its mobility. An image from projectors mounted in front of the busts at an angle of 60 degrees would be projected on the bust. However, this is not the right solution for creating a hologram, since it is a projection on a 3D model.

The last concept which we had analysed was creating a holographic fan. Under this term we can imagine circle filled with Led light*s* that is driven with a high-speed motor. The rotation of this circle with the motor led to the result of the creation of the image on the surface of the circle. We think this solution is suitable/applicable. Whether for use in the commercial or different sector, and it is possible to create this device from light materials in the various sizes needed for projections.

3 Hardware Realization Schema

Let's proceed to the actual implementation. At the beginning it was necessary to realize, how to construct halo hologram itself. It was essential to set apart its main parts. We can divide them into three parts, which are projectional, functional and processional. The last part is represented by the microcomputer which is processing data received from networks and prepares for display. Functional part, which is responsible for the movement of projection part, is build-up from the construction, in which the projection part is mounted, and also the motor responsible for its rotation. Projectional part consist of the circle fitted with led *lights,* which is rotated at a high speed and displays an image processed by microcomputer. This image is giving impressions of 3D model thanks to all the parts working together, so that in the microcomputer is program adjusted for rotation of the motor, which rotates LED – band, which creates halo hologram image itself.

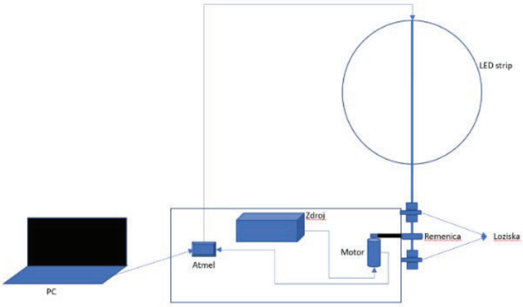


Figure 1: Schema of hologram

4 Building Hardware Realization

After analysing the problem, we started to construct halo hologram itself. Proportion of the circle on which we will install led strip, is going to be 32 cm. This circle is *fitted* with the SK9822 LED lights, which is controlled by SPI. 12V motor which rotates the circle itself is disposing with hall effect sensor, which is important for the image synchronization.

As a source of our hologram, we used old computer source from which we used voltages of 12V and 5V.

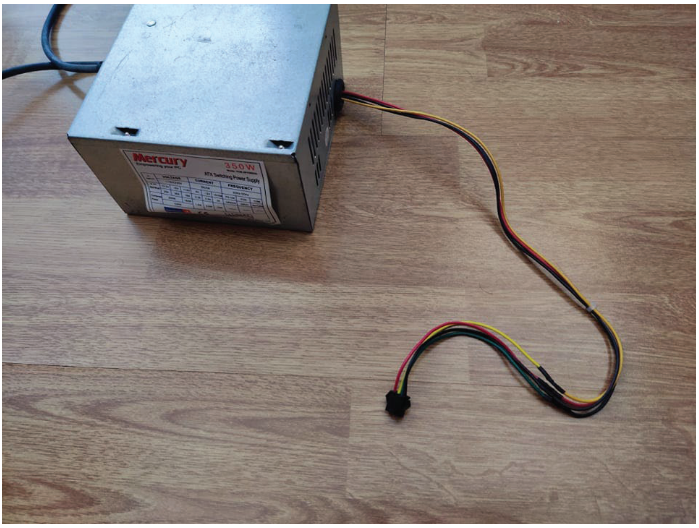


Figure 2: Source

Structure of hologram was made from the wood because of its great availability and workability. The structure was assembled and it’s holding on with the help of wooden pegs and glue. In the lower part is located a hole for a bearing, under which is also located smaller hole for a cables

We used motor, we mentioned before, with a voltage of 12V for the drive. To transfer the driving power from the motor to the display circle, we used machine-made 45 teeth Gear wheel which is connected by the drive belt *to* the motor, on which we also have machine-made gear wheel, with whom we achieved a 3:1 transfer.

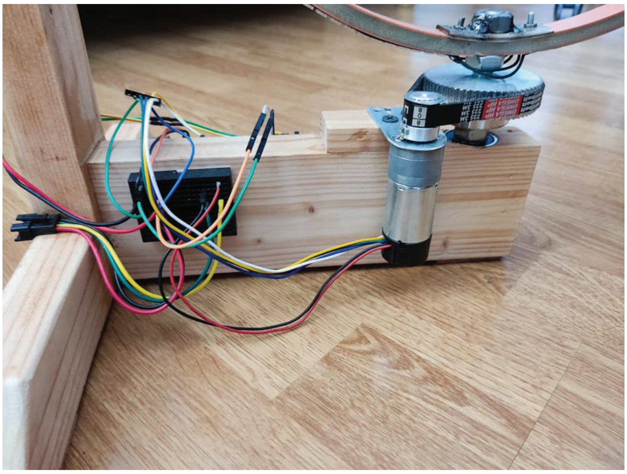


Figure 3: Drive system

The ring itself is made from PVC pipe which is attached to the threaded rod. The top mount is designed for the faster removing. On the circle is fitted Led light which has 65 led lights for a semicircle. The lead from led lights consists of 4 cables but 5V and GND is duplicated so we had 6 cables connected from the above through slip ring.

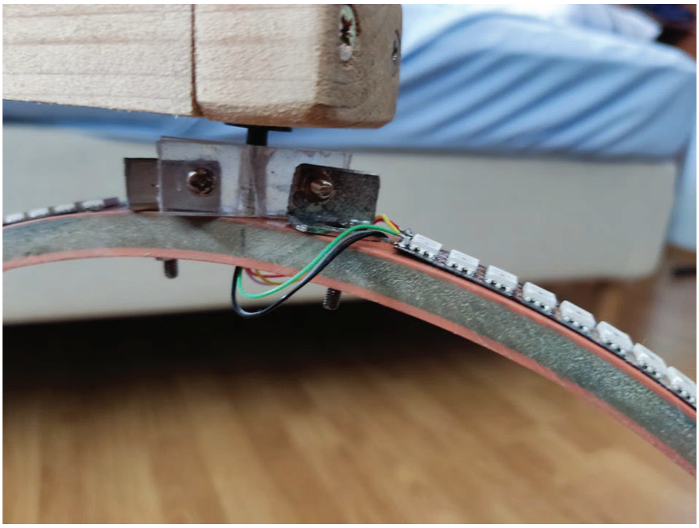


Figure 4: Detail of the upper mounting

We put the connections on the breadboard into separate lines, which we had divided according to their usefulness. On the construction itself is settled microprocessor Atmel, which controls the imaging functions for our hologram. In the end, we did not use the Hall effect sensor, which we considered essential at the beginning, because it did not provide sufficient information in interruptions, because these information were overlapped. We also connect an LCD display to our hologram, to solve the problem with Hall effect sensors. In the end we solve synchronization trough method of timing.

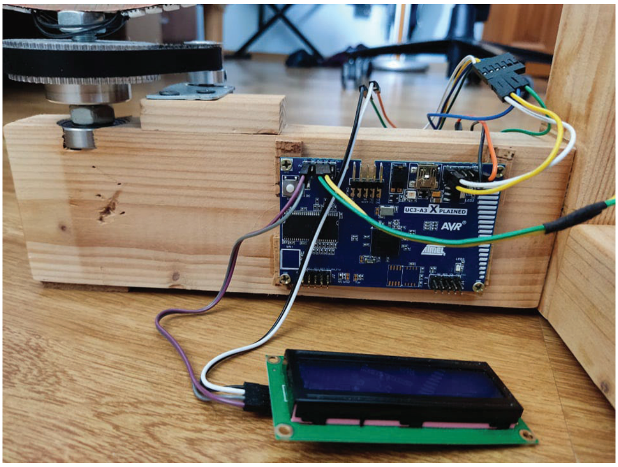


Figure 5: Atmel and LCD display

5 Result

If we were to evaluate the results obtained in this project, we can say, that we managed to construct a Halo hologram, on which we can project the image. Of course, nothing is perfect, and there is always a space for improvements, even with this project. If we were to mention some of them, it would be, for example, an improvement of the engine, because it does not have adjustable speed and also is very noisy. Another example could be the installation of a lights with a higher density of diode lights. We also managed to configure USB on baudrate 128 000, but if was not sufficient for live video transmission, so it is necessary to solve this issue with another solution. For example it could be solved by the improvement of the control microprocessor, which could have built in components like WiFi module. This improvement from Atmel can be, for example Rasberry Pi. But if we are looking at the assignment, we can say that we have fulfilled it, even though the Halo Hologram set by us is considered only as a rough concept and an introduction to the issue itself.

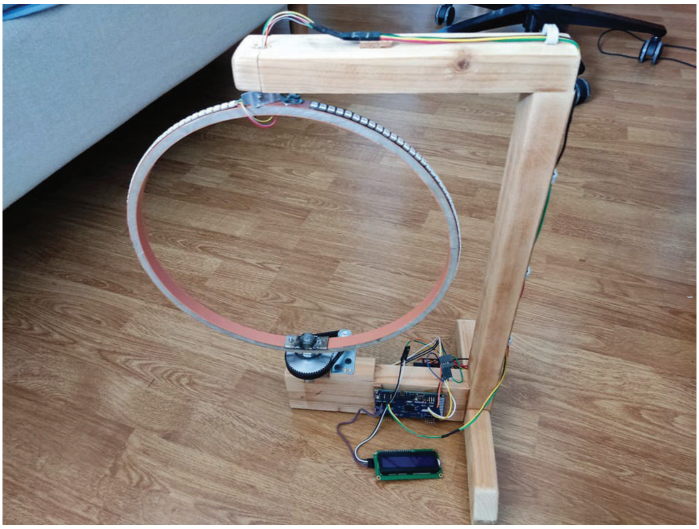


Figure 6: Image of the whole hologram

6 Conclusion

In conclusion, we would like to add that our halo hologram is really only the primitive prototype and there is always a space for improvements and innovations. To mention just a few, it is for example tensioning mechanism between motor and threated rod of the ring.

Further, implementation of hall effect sensors for synchronization of the speed of the display circle, installation of an LED lights with a larger number of diodes, more suitable motor for this type of project. But in the end, we managed to create a rough foundation, a rough foundation stone for the development of a halo hologram. With our knowledge about the initial prototype, we can further build, refine and modernize further development steps at our university.

Acknowledgment

This publication was created thanks to support under the Operational Program Integrated Infrastructure for the project: International Center of Excellence for Research on Intelligent and Secure Information and Communication Technologies and Systems - II. stage, ITMS code: 313021W404, co-financed by the European Regional Development Fund

References

[1] EGBU, Charles O.; BOTTERILL, Katherine. Information technologies for knowledge management: their usage and effectiveness. Journal of Information Technology in Construction (ITcon), 2003, 7.8: 125-137.

[2] FEHN, Christoph. A 3D-TV approach using depth-image-based rendering (DIBR). In: Proc. of VIIP. 2003.

[3] SUCHÁN, Denis Džačko1 Samuel Polakovič2 Filip; KAĽAVSKÝ, Erik; ROZINAJ, Gregor. HALO HOLOGRAM HARDWARE. 2021.

[5] BARANNIK, Vladimir, et al. The video stream encoding method in infocommunication systems. In: 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET). IEEE, 2018. p. 538-541.

This page is intentionally left blank.

This page is intentionally left blank.

Authors' List

Redžúr 2021

redzur.stuba.sk