BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Report of Project No. VI20172020068

Tools and Methods for Video and Image Processing to Improve Effectivity of Rescue and Security Services Operations (VRASSEO)

Generation and Analysis of Face Data

January 2019

Abstract

This report brings description of annotated face image dataset generation. With focus on how to obtain a 3D face image model, and which steps are needed to create desired 2D face image. Result is the working generator which will be used to create input data for face detection and recognition. First steps of the face data analysis are described. Presumed input for this analysis is the image stream. For meaningful result person on the video have to be detected and tracked. Several option are explored and the best is chosen for the following work.

Contents

1 Introduction

Testing is a essential part of each software development. Examples of input data and expected results have to be known before starting this process. Face recognition in non-standard position is not a exception. Difficulty here is to get images in which the right answer is known. In this process that means face with precise data for the pose (and other difficulties like age, light and expression). These APIE (Age, Pose, Illumination and Expression) are the biggest problems for successful face recognition.

In this work the focus is on pose and illumination. It was decided to create a generator which could provide a sufficient number of input images with annotation, so all these input values are known. Base on these images methods for detection and recognition can be tested. After that it can be measure which poses are the biggest problem, where the algorithms fail. Base on that boundaries of existing or new algorithms can be defined. Success rate or reliability of the algorithms could take this information into account.

Another problem which can be solved without this annotated dataset is the tracking of the person. Because without it there is no hope of creating effective algorithms for processing video data.

Section 2 is describing the design, implementation and results of the dataset generator. First acquirement of the 3D face model is discussed. After that necessary steps and procedures to create annotated image is described. Section 3 is coping with detection and tracking of the person in the video. Section 4 conclude and sum up achieved results.

2 Face in the Wild Dataset Generation

One the most important tasks for detecting (and possibly recognizing) face in non-standard positions is to have good training dataset. Dataset with annotated face images. Only with described dataset it is possible to test accuracy of designed algorithms. It is almost impossible to generate this data in real scenario. User would have to turn their head in precise angles within all three dimensions. Dataset is created by using real face but generated pose. First part of this dataset generation is to acquire 3D model of human head.

2.1 Getting 3D Model of Human Head

Several approaches exist for creation of the 3D model of human head. Already in prehistory, sculptors tried to capture their models as accurately as possible. To create a 3D model for a computer application the way the sculptors do, we have to use a software for 3D editing and modelling. However, computation or creation of a precise model costs a lot of time, which makes this way of large dataset preparation unsuitable.

2.1.1 3D Scanners

One interesting approach for creation of the 3D head models include scanning by the 3D capturing devices - 3D scanners. 3D scanners allow us to scan surface of an object and create a 3D mesh faithfully representing the original. Almost all 3D scanners are based on two basic principles - infrared and time-of-flight. Infrared based 3D scanners use an IR transceiver, which contains an IR transmitter for projecting a pattern to the target object and an IR receiver for capturing the projected pattern [12].

Time-of-Flight (ToF) based 3D scanners consist of a receiver and a transmitter, too. The main difference in comparison to the IR technology lies in a different signal evaluation. There are two main methods used to measure the time of flight. The first method uses a pulse modulation (PM) for distance calculation. The transmitter of the sensor stores the time the signal was sent to the scanned object [2] and, after the reflection of the signal from the object, the signal is received by the receiver and the time is stored again. Those two moments provide information about the flight time of the signal used to calculate the distance, which is then used to reconstruct the depth information of the individual signals. This approach requires a high resolution timer that measures delay between the signal emission and reception. The distance is computed directly from the time of flight by Eq. 1 [1]. The main disadvantage of this method is its accuracy due to the high speed of light and timing of the round-trip time is the accuracy of the measurement relatively low [7]. Besides, the measurement must be performer for each individual point of the object, of which we want to know its distance.

$$d = \frac{ToF\ \cdot c}{2} \qquad \text{(Eq. 1)}$$

where c is the speed of light and ToF is the time of flight of the signal.

In contrast, the second ToF method is based on continuous wave modulation. The distance is computed from the phase difference between the sent and received signal. If the sensor uses $s(t) = sin(2f_m\, t)$ signal for transmission, the amount of light *r(t)* reflected by the target is given by Eq. 2 [1].

$$r(t) = R\sin(2\pi\, f_m\, t - \phi) = R\sin(2\pi\, f_m\, (t - \frac{2d}{c})) \qquad \text{(Eq. 2)}$$

where R is the amplitude of the reflected light, f_m is the modulation frequency and ϕ is the phase shift. As in the previous case, it is necessary to calculate the distance d for each point of the object separately according to the following equation [1]:

$$d = \frac{c\,\phi}{4\pi\, f_m} \qquad \text{(Eq. 3)}$$

where ϕ is the phase shift, f_m is the modulation frequency and c is the speed of light.

The disadvantage of this technology is problematic scanning of the glossy surfaces. Both 3D scanner technologies have problems with background light, which can affect the projected pattern [6].

2.1.2 Stereoscopic 3D Model Creation

Another possibility of 3D model creation is use of the 2D images of head for 3D model reconstruction by the stereoscopic algorithms. The main advantage of this approach, in comparison to the manual model creation, is leaving the time-consuming process of the model creation to the computer. The computation demands are very high, because the reconstruction contains several computationally challenging algorithms, yet the time gain is in orders of magnitude.

For each image, the position, in which the image was taken, must be determined. After that, rays going through selected points are projected to the space. If rays running through equivalent points from different images intersect, the 3D position of that point can be computed. To achieve a realistic 3D model many more steps must be performed (e.g. texture mapping or error suppression for the surface smoothing). We tried various open source solutions for this purpose, but the only solution producing high-quality results was application Zephyr made by 3Dflow.

The principles of some of the important algorithms used by Zephyr have been published (e.g. Hierarchical structure-and-motion recovery from uncalibrated images [11], Towards automatic acquisition of high-level 3D models from images [10], Improving the efficiency of hierarchical structure-and-motion [4], Practical autocalibration [5] and Structure-and-Motion Pipeline on a Hierarchical Cluster Tree [3]). Using the tool, we created models from different numbers of images (see Fig. 1a and 1b). To reconstruct the 3D head model, the program needs optimally 50 images. In the Figure 1c and 1d, we can see a 3D mask with texture. Above all, the 3D model created from 39 images is important, because from this number of images features of the face (eyes, shape of the nose, corners of the lips, etc.) become visible.

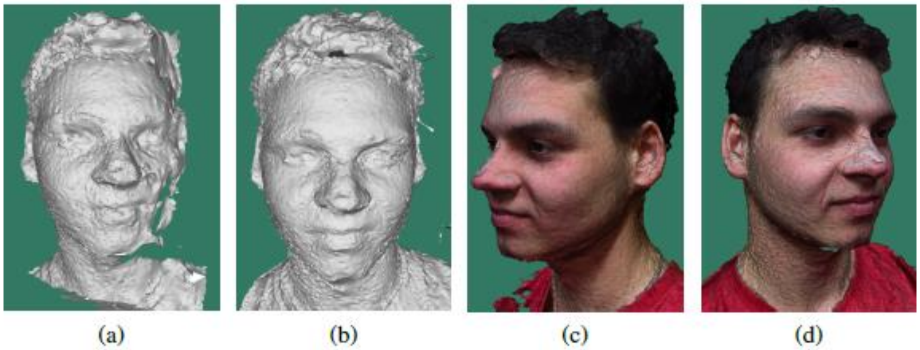

Figure 1: a) Model created from 11 images (input dataset contains 25 images). b) Model created from 39 images (input dataset contains 50 images). c) and d) Models with textures with directional light.

2.2 Dataset Generator

The dataset for classifier training should contain many variations of positive samples and many variations of negative samples [8]. Representation of different types of positive samples in the dataset should be uniform to avoid classifier specialization. If the dataset has to contain many images, it is time consuming to get a snapshot and annotation. If the used 3D models have to represent reality, a lot of pictures can be generated using the generator generously.

One approach to getting models of real objects is scanning. Using such a model can create scenes that are close to reality. For example, pedestrian models can be used to create an assembly scene on a square. For these purposes, a tool called SYDAGenerator (SYntehtic DAtaset Generator) was developed. The program was created to create balanced datasets with heads in different positions. However, it is not limited to other types of models. The software allows to generate a dataset consisting of model projections in various poses and with applied post-processing. With this, the dataset with various images can be created.

2.3 Generation Process

The main goal is to create 2D images based on the 3D model and scene parameters. Output images may contain 3D model with different position, rotation and size. The scene background of output images may be different too. The primary requirement is that the resulting images seem as naturally as possible. The SYDAGenerator uses for 3D model the processing library LibGDX[1]. Therefore, the model has to be converted to g3db file format. Unfortunately, the maximum of model vertices's is limited to 32K. This process can be divided into the following parts:

- 3D model transformation
- Scene settings
- Images generation
- Image post-processing

2.3.1 3D Model Transformation

The scanned 3D models are not directly usable as they are, because of various defects present in the model. Since the most of the defects can be repaired by transformations, SYDAGenerator offers simple tools for transformation (translation, rotation and scale). If more advanced modifications are required, the model needs to be edited by an external application (e.g. Blender[2]).

SYDAGenerator can rotate the 3D object in yaw, pitch and roll during the generation phase. Rotation, however, brings a question as to which point the rotation should be performed.

Once the desired transformations and head center are set, it is possible to save the transformation matrix to JavaScript Object Notation (JSON) file along

[1] https://libgdx.badlogicgames.com

[2] https://www.blender.org

with the model file. After running the program, it is checked whether a transformation matrix file exists. If so, it is used. This is an important feature for an automatic generation.

2.3.2 Scene Settings

The previous step was focused on preparing the 3D model. Scene settings is the next important step in dataset generation. The software simulates capturing images by a camera, therefore the camera properties and illumination have to be set. The 3D model is in the forefront before the background that is created from picture and it is projected into the scene without transformation and deformation.

Another important parameter is the lighting setting. By default, the tool uses to illuminate the scene ambient light only at which the colour temperature can be set. However, if we need directional lighting, the parameter can be set in the configuration file. Moreover, when the model has deformation on the surface, the directional light causes artifacts highlighting. The software allows to set the camera position too. But after our experiments with face detectors, we've come to the conclusion that it is better to set the camera position to straight direction.

2.3.3 Images Generation

The main task of the program is to create datasets and perform data augmentation, so the program allows the transformation of a 3D model during the generating. To train the algorithms for determining the rotation of the head it is necessary to use an annotated dataset. During the generation process, the 3D model can be rotated in yaw (ψ), roll (ϕ) and pitch (θ). For the purpose, the Euler rotation given by Eq. 4 and 7 [9] is used. According to program settings, the combined rotation or rotation in only one plane is performed. Amount of dataset images is determined by step size and range of angles. Additionally, the background can be changed for each iteration of rotation.

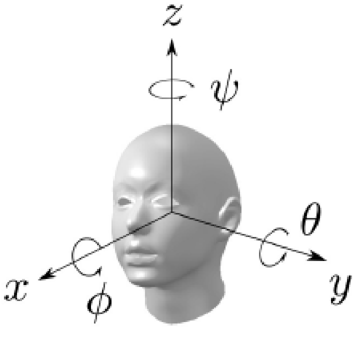

Figure 2: Euler angles [9].

$$R(\phi, \theta, \psi) = R_z(\psi)R_y(\theta)R_x(\phi) \qquad \text{(Eq. 4)}$$

where R is rotation matrix and R_z, R_y, R_x are partial rotation matrices.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \qquad \text{(Eq. 5)}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \qquad \text{(Eq. 6)}$$

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{(Eq. 7)}$$

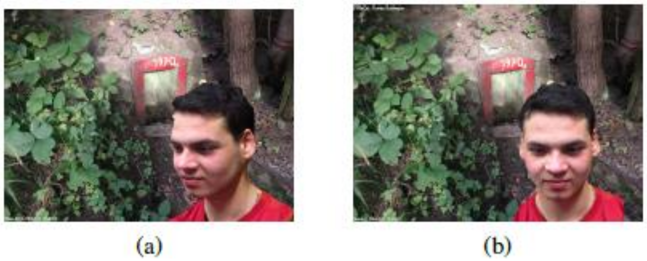

Figure 3: Examples from dataset for testing face detection algorithms. Each image contains a head in a certain rotation.

2.3.4 Images Post-processing

The data contains many undesirable elements resulting from the properties of cameras and environmental influences. Camera characteristics that affect the image include chip resolution, focal length, lens distortion, and more. If we want to test algorithms for detection and identification, we should take these paradigms into account. SYDAGenerator supports image modification with the image processing algorithms. This part is useful for classifier testing.

3 Face Analysis

Important task to work with the face in the wild dataset is to detect the face. When dealing with video data it is necessary to detect a person. Then to track this person so the information that would be gathered are not lost. In parallel to previous step face detection and recognition can be done whenever person is detected. Without this part any other analysis of face data could be theoretically done but it would be very inefficient or more likely unable to perform in video data. It is also possible with some effort to alter designed generator to generate whole siluette of a person. But to start with the tracker it is not needed because there are a lot of video of persons. This section is based on the [13].

3.1 Person Tracker

Goal of this work is to assign each person in the scene a unique ID. This ID should stay with the person for the whole appearance in the scene. First detector is finds person in the scene. After few frames for each detected person an instance of tracker is activated. Then the detector is called again, detected persons which are currently tracked are filtered. After few frames tracker is activated and starts to track new detected person. This process is applied on the whole video sequence. Tracker of course has to end when tracked person leaves the scene.

Detector *YOLO* is used. *YOLO* is state-of-the-art detector which is available in OpenCV. Weights for this detector has to be downloaded and loaded by the application. Several methods were tested as a tracker: kcf, boosting, mil, tld, medianflow, mosse, csrt. Last mentioned *csrt* from OpenCV contrib was chosen, because it had the best result among the others. When a tracked person is more than 20 % out of the scene then its tracking is automatically ended.

Last thing to solve was the update of tracker from detector. As it was described after each detection there have to be filtration whether the person was in the scene before, or if it is a new person. For that NMS (Non-Maximum Suppression) function was used. This function is computing overlap of defined areas and using specific parameters it decides if detected person is the same or not. When new detected person does not have sufficient overlap with others it is marked as new. When there is a sufficient overlap then location of the person is updated. Results of this method could be seen in Fig. 4.

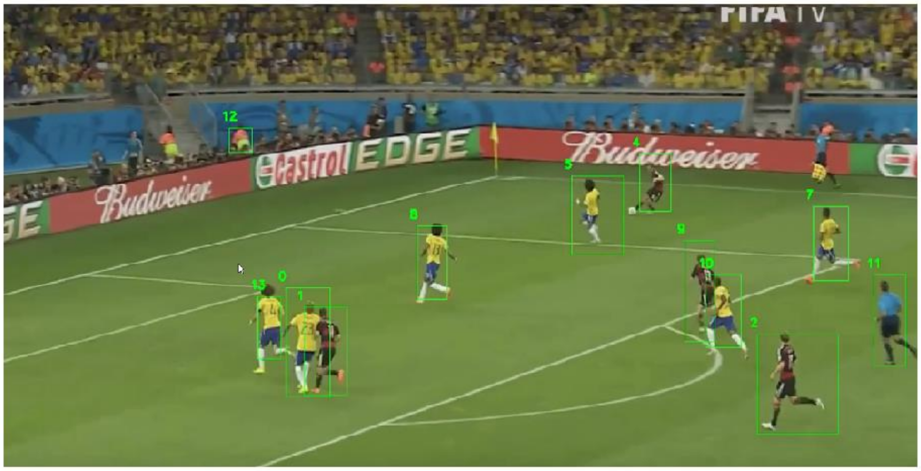

Figure 4: Example of the results from person tracker. Persons are in green rectangles with their respective ID shown as the green numbers.

4 Conclusion

With the rapid development of classification algorithms and neural networks, requirements for quality datasets are emerging. Unlike augmentation methods that only modify dataset images, new images can be created with image generator. Images are based on the 3D models and background composition. In terms of the testing of any face detector, it is essential for each image to have an annotation containing information about the face position and pose. The SYDAGenerator allows to generate datasets designated for training and testing of the classification algorithms. The tool creates an annotation file for each image in the set. In case the 3D models are created by scanning of the real objects, the generated images look realistically. In addition, SYDAGenerator has an option to use a post processing filter to simulate various camera properties. First step of the analysis was done and that is the tracking of the person in the video. So that it is possible to locate the faces.

References

[1] A. Bevilacqua, L. Di Stefano, and P. Azzari. People tracking using a time-of-flight depth sensor. In Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on, pages 89–89. IEEE, 2006.

[2] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt. 3d shape scanning with a time-of-flight camera. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 1173–1180. IEEE, 2010.

[3] M. Farenzena, A. Fusiello, and R. Gherardi. Structure-andmotion pipeline on a hierarchical cluster tree. In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pages 1489–1496. IEEE, 2009.

[4] R. Gherardi, M. Farenzena, and A. Fusiello. Improving the efficiency of hierarchical structure-and-motion. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 1594–1600. IEEE, 2010.

[5] R. Gherardi and A. Fusiello. Practical autocalibration. In European Conference on Computer Vision, pages 790–801. Springer, 2010.

[6] P. Mitas, J. Gorski, J. Wilkowski, P. Czarniak, and P. Podziewski. Unique problems of a structured-light 3d scanner during scanning of various wood species. Annals of Warsaw University of Life Sciences-SGGW. Forestry and Wood Technology, 88, 2014.

[7] E. Rakitina, I. Rakitin, V. Staleva, F. Arnaoutoglou, A. Koutsoudis, and G. Pavlidis. An overview of 3d laser scanning technology. In Proc. of the International Scientific Conference, (Varna, Bulgaria). Citeseer, 2008.

[8] D. Riccio, G. Tortora, M. D. Marsico, and H. Wechsler. Ega ethnicity, gender and age, a pre-annotated face database. In 2012 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS) Proceedings, pages 1–8, Sep. 2012.

[9] C. Segura and J. Hernando. 3d joint speaker position and orientation tracking with particle filters. Sensors, 14(2):2259–2279, 2014.

[10] R. Toldo. Towards automatic acquisition of high-level 3d models from images. Universita degli Studi di Verona, Verona, 2013.

[11] R. Toldo, R. Gherardi, M. Farenzena, and A. Fusiello. Hierarchical structure-and-motion recovery from uncalibrated images. Computer Vision and Image Understanding, 140:127–143, 2015.

[12] Z. Zhang. Microsoft kinect sensor and its effect. IEEE multimedia, 19(2):4–10, 2012.

[13] R. Pazderka. Monitorování pohybu chodců (Pedestrian movement monitoring). Biometrics course project documentation, 2018.