FACIAL EXPRESSIONS ANALYSIS BASED ON A COMPUTER VISION ALGORITHMS

Iurii Krak, Anton Ternov, Vladislav Kuznetsov

Abstract: *In this paper a variety of computer vision algorithms in a context of their application to the problem of study of facial expressions were analyzed. Performance characteristics of computer vision algorithms for tracking markers which were attached to the face in ideal and real conditions were obtained. A list of algorithms' faults that occurred during experiments was given, and ways to reduce an impact of various factors that influenced the experiments were suggested. The results of experiments were used to model facial expressions on a face of a virtual model of a human head.*

Keywords*: facial expression, computer vision, reference points.*

Introduction and problem statement

In studies of facial expressions, researchers often use tools of verbal description that can contain both text descriptions in a form of predicates and different schematic symbols [Krak_1, 2012]. The results of such studies may be different models which describe variety of human mimic conditions [Hanke, 2002, Sutton, 2002], or a variety of components of such expressions [Miller, 2011].

Analyzing the results of these studies, an amount and a quality of the work and the originality of classification of facial expressions should be noted. However, such research is inherent by a drawback: presence of a human factor that occurs when we create a model of facial expressions. Human intellect, of course, in the analysis of complex objects and in amount of samples of images of objects is more powerful than a computational intelligence, but it is inherent in less accuracy, abstraction and simplification. Also worth noting that verbal data which are obtained from such studies are difficult to analyze and compare.

According to the article [Krak_2, 2012], solution to such problems is the use of computational intelligence in such way that it would be possible to obtain a classification of facial expressions and their components.

The problem statement is to develop and to analyze tools of computational intelligence for the next criteria (requirements) satisfaction:

– High speed of computation;

– High level of automation (low level of human impact during processing time);

– Low level of errors caused by primary data.

Methods used for the analysis of the human facial expressions

Speaking about methods used for the classification of facial expressions and their components, it is important to understand that in a process of solving this problem, we intentionally simplify an object of study. Simplification of the object allows us to get the object model and thus solve the problem using available means of computational intelligence.

A model of object in the solution of this problem is a discrete map (projection) of an object in three-dimensional space by a sequence of maps over time (image stream) in two-dimensional space. Model of the object has a lower space dimension than original one, and becomes discrete by time (not taken into account all states, only states at regular intervals).

For such a model of the object, however, a complement is needed in order to have a possibility to move from a two-dimensional space to the object space. This requires at least another projection of the object to obtain coordinates of an object in three-dimensional space. This can be done in two ways: by using the original image directly and calculating surface of the object based on several projections (solving is based on depth maps), or by going to a mathematical abstraction, by reference points (markers) which are used to calculate position of points on a surface of a face.

Marker is an object with known pre-physical properties (color, shape, size), the form of which is invariant to affine transformations such as rotation and tilt. These criteria are satisfied by spherical object whose color is different from properties of the studied object (a human face). Usage of such abstraction is preferred over direct study of a face surface. Because it can be proven by simple geometric transformations that for each element of the object’s model (marker) and for each moment of time based on coordinate increments on projections of two images, real coordinate of a point on a surface of a face can be calculated. Thus, this method allows finding an exact solution for a set of the points, but for all other points, a solution is to be found approximately. A positive aspect of this solving method is lowering computational complexity compared to the solution based only on images.

Considering the solution of this problem by using the abstraction of "tokens" we should find a way to calculate coordinates of a marker in two-dimensional space of frame. The problem of calculating the coordinates can be solved in general by computer vision techniques and algorithms.

Algorithms of computer vision

Algorithms of computer vision is a general name for a class of algorithms used to extract objects from the image, which are set explicitly or implicitly, as well as to get information about these objects. These algorithms include several levels of algorithms that are used at various stages of processing images and primary data [Panin, 2011; Forsyth, 2004; Shapiro, 2006]:

– Image filtering algorithms - extracting specific areas from an image by specific criteria: areas of a certain shade, transitions from one shade to another, etc. Depending on a task, different types of algorithms will be used;

– Algorithms of selecting background and an object on stationary background. These algorithms compare footage sequences and infer the presence and direction of a change. These algorithms include methods of separation of an optical flow and block matching methods of pixels samples on an image with a standard image;

– Algorithms of detection of key features: they select areas of high brightness, corners, lines or circles. Modifications of these algorithms are algorithms for computing descriptors of key features. Based on the information about the image, mathematical operations are performed, which form an object "descriptor". This "descriptor" describes in a special way characteristic areas on an image (areas of singular points);
– Key features comparison algorithms on a video stream: based on similarity criteria, key points are compared on different images. Point's descriptors can be also used instead of the key points;
– Reconstruction algorithms – 3d coordinates are found based on a position of an object on different projections of an object.

Such algorithms can be used for various tasks related to image processing: image compression, pattern recognition and tracking of some objects in the image sequences, etc. A remarkable feature of such problems is the uncertainty associated with the properties of the objects. Objects are rarely flat and can be modified depending on their position in relation to spectator, and information about the properties of an object can be obtained either through training or by sufficiently clear criteria that lay down in the algorithms of comparing key features. Apart from them, there is a class of algorithms that do not belong to the class of computer vision algorithms, but which are used in computer vision problems: - *classification and clustering algorithms*: based on known criteria, obtained from the study of objects or based on information about detected features, separation of features to entire groups (classes) which correspond to specific objects is performed.

A choice of algorithms that suit the best for this task can be done experimentally by verifying a few representatives of the classes of algorithms of computer vision. They must fit to certain criteria to be applicable for solving this task.

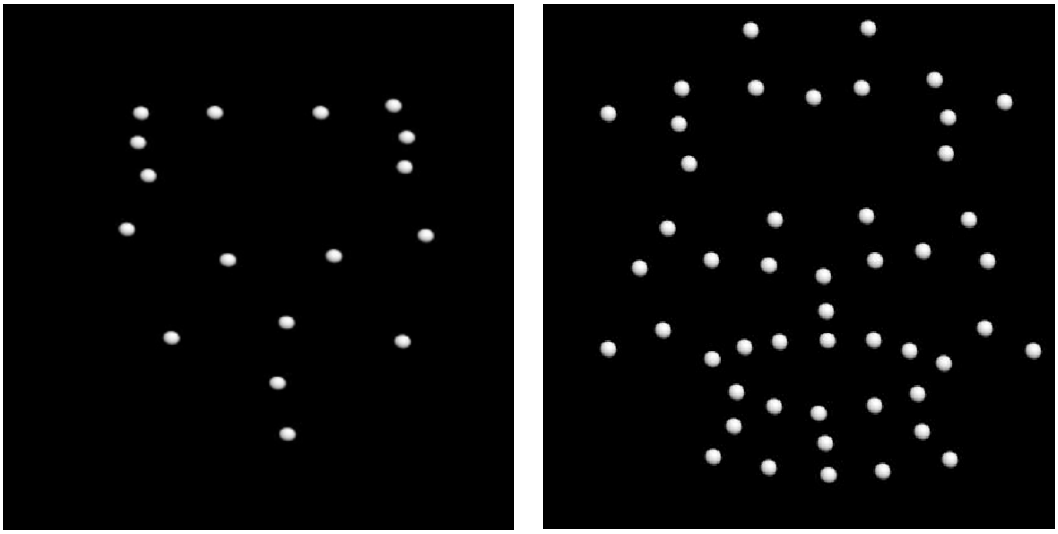
Experimental verification of algorithms. It has been proposed to check computer implementations of these algorithms on real data. The following parameters were analyzed: a) computational speed; b) particular qualities of extraction by different types of algorithms; related to the task of tracking the "marker-like" feature; c) influence of errors in primary data on computation in different algorithms (we did not take in to account errors of primary data during ideal experiment).

Information about the software used in the experiments. Different implementations of computer vision algorithms were tested, in particular: 1) corner detectors Harris, DlineCorner, Susan, Foerstner; 2) key features descriptors: SIFT (Scale-Invariant Feature Transform); 3) optical flow methods: KLT (Kanade-Lukas-Tomasi); 4) block matching methods of pixels' samples in the image with the standard image SAD (Sum of Absolute Differences) and NCC (Normalized Cross Correlation, which contained in the software for video processing: Blender [Blender], ACTS 1.5 [ACTS], Voodoo 1.5 [VisCoda VooDoo], Puchcard VideoTrace [Punchcard VideoTrace], Viscoda VooCAT [VisCoda VooCat], Vicon Boujou [Vicon Boujou], Autodesk Matchmover [Autodesk Matchmover], ImagineerSystems Mocha [ImagineerSystems Mocha], SSontech SynthEyes [SSOntech SynthEyes], Eyeon Fusion [Eyeon Fusion].

Experiments on the study of the algorithms' properties in the ideal conditions

In order to study the performance of algorithms that are present in the software, we created a test video that contained facial expressions without any distortions (including rotation and tilt distortion caused by head movements and some distortions caused by big shutter time in budget camcorders).

A pair of files containing the trajectory of markers on the face was used for these experiments. Files were obtained in professional motion capture recording studio [Easycap facemotion]. First file contained 22 markers (number of markers was reduced in a half), the second one – 48 (original quantity of markers, not taking into account the initial markers on each image). Content of these two files has been converted into a two-dimensional image sequence (video sequence) which represents a frontal view of the three-dimensional scene for each frame appearing on a black background (see Figure 1).



a) b)

Figure 1. Fragments of the first frame of the image sequence: with a reduced number of markers (a); with the original number of markers (b).

Obtained sequences were loaded into a software package in order to get the performance characteristics of individual algorithms used in software implementations. If there was a choice of algorithm (or combination of algorithms) in the software packages, then all the available single algorithms or their combinations were studied (conditional mood).

The experiment was carried out as follows:

– Pre-configured number of tracked features was used – from the minimal 1 (in order to estimate the maximum speed), with several intermediate values of the number of tracked features – 4, 10, 17, 37, 57, 75 (to find out how the estimated performance of the algorithm depends on the number of tracked features) to the maximum value – 75 (total number of features present on the test video sequence);

– Software timer calculated the time required to process the entire video sequence for each run;
– Obtained values were entered into a spreadsheet. Then, the characteristics of the frame rate (FPS), which is used to evaluate the performance of graphic applications, were calculated.

The results of the experiment on the performance at the stage of test run showed that some of the algorithms (or rather their implementations) do not adequately perceive the object "marker". Most of the algorithms which were present in the software packages, such as ACTS, VideoTrace, VooCAT and Voodoo Camera Tracker, detected significantly more (in 3 - 4 times) features that were present in the image. For this reason, we decided to evaluate the performance of the algorithms only in such implementations: 1) in Blender – algorithms: SAD, KLT, Hybrid SAD-KLT, and 2) Autodesk Matchmover – unknown algorithm, and 3) Fusion – unknown algorithm. However, one of the algorithm represented in Autodesk Matchmover showed the best results in quality.

Based on the data obtained in the first experiment, the graphs that show how a processing time depends on the number of features in the image (see Figure 2) were built.

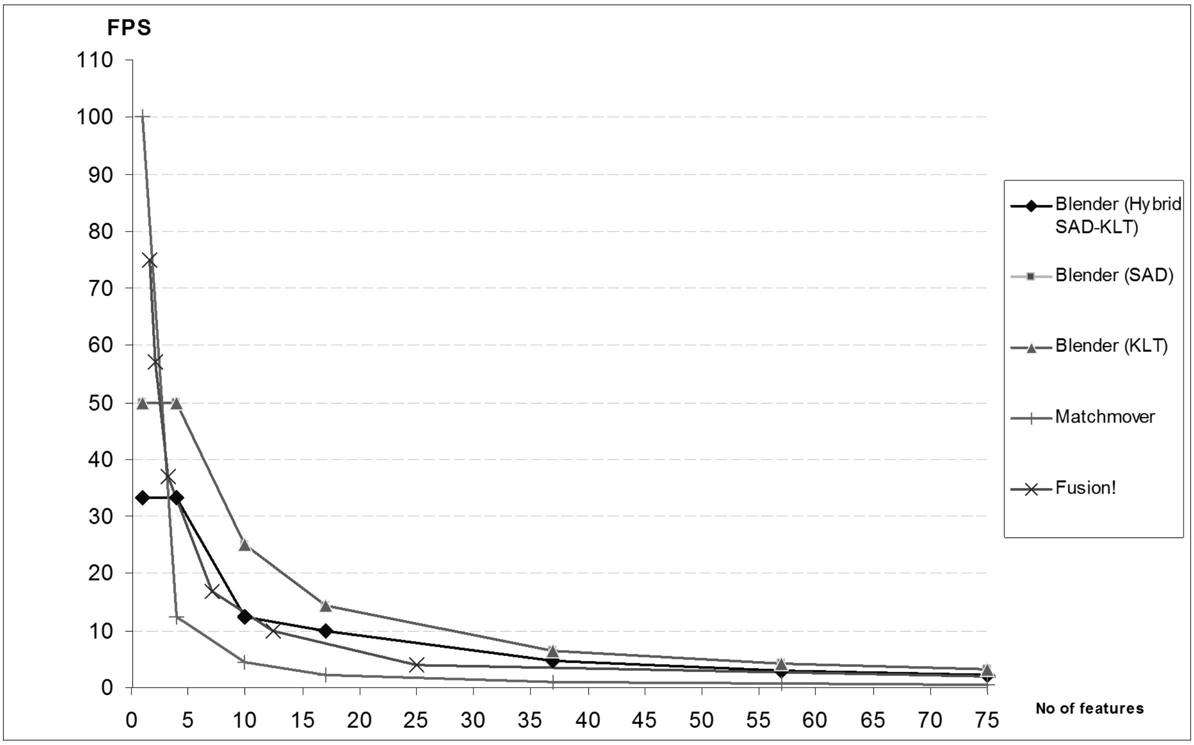


Figure 2. Plots of the processing time depending on the number of features in the image

It can be noted that the implementations of algorithms used in software product Blender showed the best performance characteristics in ideal conditions. However, these graphs do not show the superiority of one algorithm over another, but merely reflect the quality of the algorithms in the specific software package applied for this particular task.

After carrying out the experiment on the calculation of the performance of algorithms' implementations, the errors that might occur during the experiment were analyzed and some suggestions were given:

– Corner detection algorithms (e.g. FAST) find excessive number of points (i.e. circle is perceived as a polygon), and there is no stability (repeatability) in the extraction of features from the same area;

– Feature descriptor algorithms (e.g. SIFT) show better results in conjunction with detectors on which the algorithm is tuned for, otherwise, the algorithms start working with serious errors;

– Algorithms of block matching of pixels samples in the image with the reference image (e.g. NCC) show better results with integral features (such as, e.g., SIFT), than with a set of separate features (e.g., with FAST corner detector);

– Methods of optical flow detection (e.g. KLT) in conjunction with the key features descriptors are faster and more accurate than methods that use a pair of descriptor-descriptor (e.g., SIFT-SIFT).

These findings give a reason to believe that for further experiments on tracking an object "marker" on real data it is desirable to use implementations of algorithms that are better suitable for identifying circular object, such as: 1) circle detection algorithms, 2) algorithms that contain methods of optical flow, and 3) algorithms of block matching of pixels samples in the image with the reference image. This selection was based on a fact that the rest of the algorithms, which were analyzed in this experiment, showed poor results.

Verifying algorithms on real data

For this experiment, a scenario was set up that contained a list of 70 mimic emotional facial expressions and 88 components of these expressions obtained in a study [Krak_2, 2012], and various instructions for carrying out the experiment. Based on this scenario, few videos with different types of markers on the face were taped.

Since to hold such an experiment is very time-consuming (based on the performance of the algorithms presented in Figure 2) it was proposed to conduct the experiment in four stages (Figure 3) for all types of algorithms used in the experiment.

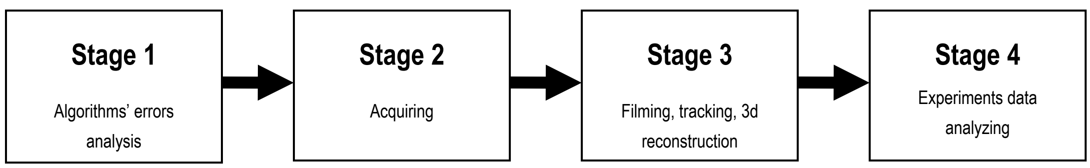


Figure 3. The scheme of the experiment

On the first stage, a study of algorithms' errors of three previously proposed classes was conducted on the real data. Then, a second stage has an objective to analyze the functionality of 3d reconstruction algorithms in order to select the optimal video shooting conditions, which would give adequate results of the 3d reconstruction algorithm. The third stage consisted of: 1) movies shooting based on the shooting conditions obtained on the second stage, 2) analysis of the video clip by computer vision algorithms, which showed the best results on the first stage. The fourth and final stage was an analysis of the experimental results based on the proposed scenario and, thus, evaluations were given to all the experiments conducted in this paper.

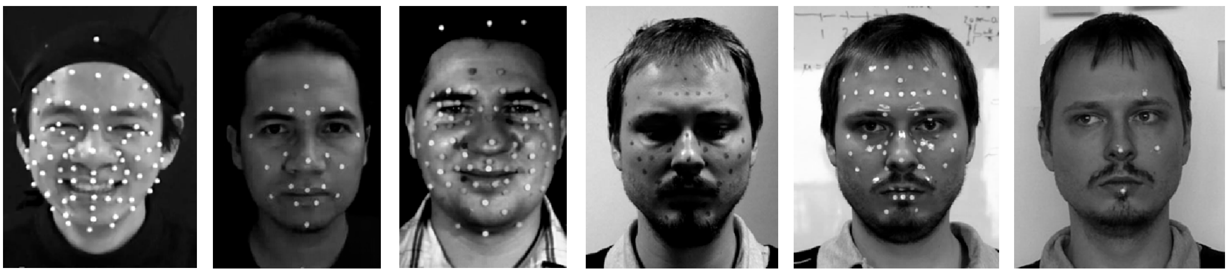
Stage 1. Error analysis of algorithms on real data. It was proposed to make a pilot-test on real data of the several algorithms from different classes of computer vision algorithms that are better applicable to the problem of tracking the object "marker":

- The class of optical flow algorithms (for example, the Kanade-Lukas-Tomasi algorithm);
- Class of methods of block matching of pixels samples on an image with the reference image (for example, the Sum of Absolute Differences algorithm);

Class of circle detectors (for example, the algorithm Hough transform, its implementation in the computer vision algorithms library OpenCV [OpenCV]).

For this experiment, few test recordings with a shorter duration were obtained according to the previously proposed scenario. Acquired videos had the following parameters:

- 70 convex markers of the same color, 1000 frames (Figure 4a);
- 20 convex markers of the same color, 750 frames (Figure 4b);
- 70 convex markers of different colors, 350 frames (Figure 4c);
- 22 markers, painted on skin, 1000 frames (Figure 4d);
- 22 flat markers, 1000 frame (Figure 4e);
- 4 convex markers, 1000 frames (Figure 4f).



a) b) c) d) e) f)

Figure 4. Shots from video sequences used in the experiment

During capturing errors analysis, relation between types of used markers and number of primary data errors was analyzed in all cases. Detailed list of the most common errors associated with primary data was compiled.

Errors that accrued due to the type of used marker:

- Usage of flat markers could cause a loss of tracked features if there appear affine transformations such as "turn on the axis Z" and "turn on the axis Y" (so the markers should be round and preferably convex);

- Markers made of washable paint applied to a face might be merged with background color due to camcorder video compression algorithms;
- White spherical markers comparing to flat markers and markers painted by washable paint gave the smallest mean deviation from previous frames;
- The errors occurred when markers appeared on the edge of face and background.

Errors that accrued due to the poor quality of the video:

- Loss or shift of detected feature due to the transition of marker color to the skin color;
- Blurring due to the usage of compression algorithms with a high level of losses during sharp movements;
- Loss of markers may appear on the blurry images of low resolution (less than 640x480p) or because of a hardware compression;
- Auto-Tuning of the brightness and camera focus distance led to the fact that the hue of a feature was also changed and segmentation buy hue value did not work correctly;
- Due to unscattered light from fluorescent lamps, uneven lighting of a face from all sides is observed, which causes a glare on a surface of a face and on markers, which led to errors;
- The presence of noise on an image creates changes of a feature's border location within certain limits (approximately at the diameter of grain of noise spots) that led to the characteristic fluctuations on a mean value.

During the analysis of these algorithms, interesting feature (i.e. ability) in Hough transform algorithm was noted. Even when there is a trail (transition of object "circle" into an object formed by its path) caused by low quality of obtained video, algorithm defined a trail as a single circle, which was located at the "beginning" or at the "end" of the trail.

These results suggest that in addition to the actual raw data error correction (low quality video), the algorithms require completion of noise and color threshold limiter (use of feature's shade instead of brightness of its individual components) or additionally use the method of image segmentation, which would allow to eliminate fluctuations in the brightness of the image. It is also important to note that it's strongly recommended to increase stability of feature's boundary since it leads to a shift of the feature's center location relatively to the real feature location. To enhance the stability of a feature, the adjusting of the results by curve-fitting methods can be used.

Stage 2. Testing the reconstruction algorithms. For testing reconstruction algorithms, a series of experiments were conducted in order to establish the conditions necessary for the shoot. A built-in algorithm to the software Autodesk Matchmover was used for the reconstruction.

Series of experiments were performed with several configurations of cameras

- 1 camera in the middle, one camera on an angle of 45 degrees to the middle camera;
- 2 camera on an angle of 45 degrees to the frontal plane;

– 1 camera in the middle, 1 camera on an angle of 45 degrees lower in the vertical plane.

Results of the experiments have established the following:

– The best visibility of markers was provided when the facial markers were on the background of the skin and the static key markers, which form the global coordinate system, did not stay on the edge of face and background;

– When shooting a face from side angles (1x45 or 2x45) we did not meet the first conditions, so the cameras with such configuration were not used;

– The best result was shown by the configuration of cameras: one camera in the middle, one camera on an angle of 45 degrees lower in the vertical plane.

Stage 3. Movie recording and analysis of the obtained data. Based on the obtained results on a stage 1 and stage 2, the optimal shooting conditions were chosen, however, none of the algorithms fully satisfied all the requirements. Either it had a high accuracy on the ideal data and low error tolerance of primary data, or it had too great drift of a features. In addition, all of them were inherent in the extremely low level of automation - the results have to be converted to other file formats in order to carry out the reconstruction of 3d.

For this purpose, it was proposed to use a software package Autodesk Matchmover, as it fits to the most of the criteria required to obtain three-dimensional coordinates of objects "marker":

As mentioned in the section "**Experiments on study the properties of the algorithms in the ideal conditions**", computer vision algorithms showed the best results of quality (but they seriously lost in speed as shown on Figure 2);

– This software package allows performing both feature tracking and 3d reconstruction of the markers' orientation in space, taking in to account the camera optical distortion and relative position of each camera;

– It was possible to save the data in an XML format that was suitable for the further analysis of the numerical parameters;

– Data obtained from the program can be conveniently used directly in other graphics software packages installed on a computer (for verification of the data).

Besides the Autodesk Matchmover, during the experiments various types of software (including video and audio editing software) were also used.

The process of obtaining of three-dimensional trajectories was held by a particular scenario (see Figure 5).

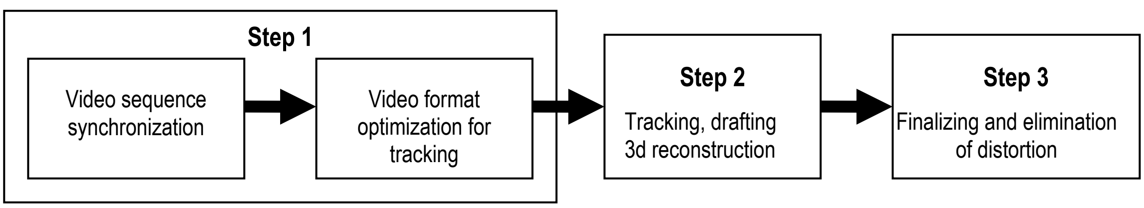


Figure 5. The scenario of a three-dimensional reconstruction

The video obtained from the cameras was separated from audio track. The resulting audio tracks (or rather spectrograms) obtained from each video were analyzed in the program Audacity [Audacity]. Peaks of amplitude (the beginning of a sound) or other features were found on each spectrogram, which proceeded the moment of beginning of useful information on the video. These points on each of the tracks defined the synchronization point of the two videos.

Resulting time values of synchronization points were used during video transformation. Experimentally the best compression algorithm for video was found. It's MJPEG. This format provided algorithms of tracking to work significantly faster comparing to the original video formats MPEG4 or H.264 due to rapid decompression of video in MJPEG (approx. 10 times faster).

Reconstruction is divided into two stages. Initially, the program selects from the entire sequence of images and videos a set of key frames for which the position of each point in space is calculated taking into account a projection of this point on the plane of the frame on each camera. Then obtained results interpolates on the entire range of frames. If some group of markers from the "cloud of markers" changes its position and rotation relatively to the camera, their movements are taken into account in calculating the movements of each marker that belongs to the cloud of markers.

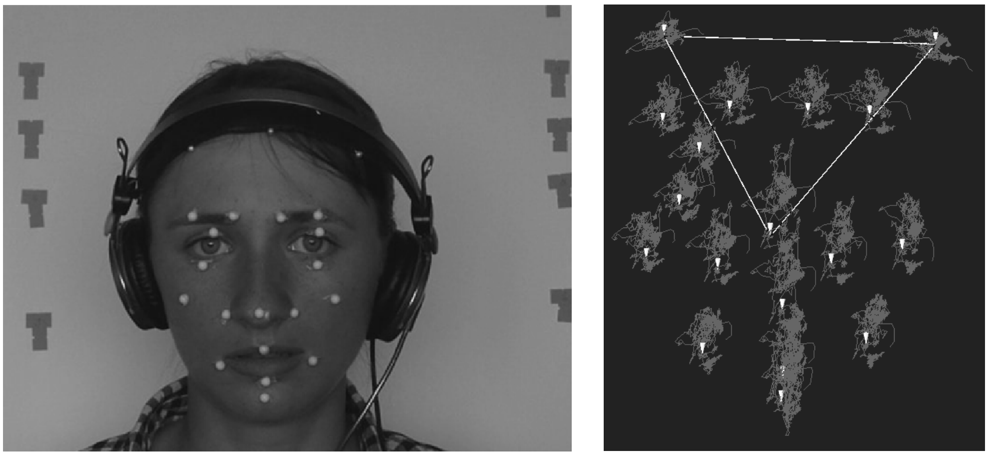
A draft version of the "cloud of markers" has relative position of the markers, which vaguely reminiscent of their location in reality. To obtain more precise location of the markers on a space, calibration of the camera is used (for different versions of the program, this step may be performed before or after the reconstruction of points positions).

Camera calibration is needed for solving such tasks as: 1) clarify the relative positions of cameras in a space, by using the position of static markers in the scene as well (that important), and 2) calculation of the optical distortion of the camera and its settings (even if they are not known in advance). On a stage of calibration, the maximum number of points on a stationary background is calculated and then correction coefficients are calculated for the earlier obtained draft of movements' reconstruction.

On this stage, sufficiently accurate position of each marker (including static) in the space (see Figure 6) is obtained. This can be used for the further studies or to control the facial animation of virtual human head in the one of the computer graphics software packages.

Stage 4. Getting the results of a 3d reconstruction to check the validity of the data. In order to check the quality of 3D reconstruction and to evaluate the reliability of the data, it is necessary to check the data on the test 3d model to view the changes of facial features (jaw, lips, cheeks, nose, eyelids and forehead).

For this purpose, software package Autodesk MotionBuilder [Autodesk Motionbuilder] has been chosen. Obtained data in a cloud of markers were transferred to the resulting parametric face model. Then perception of various facial expressions generated on the face model using the original data was visually rated.

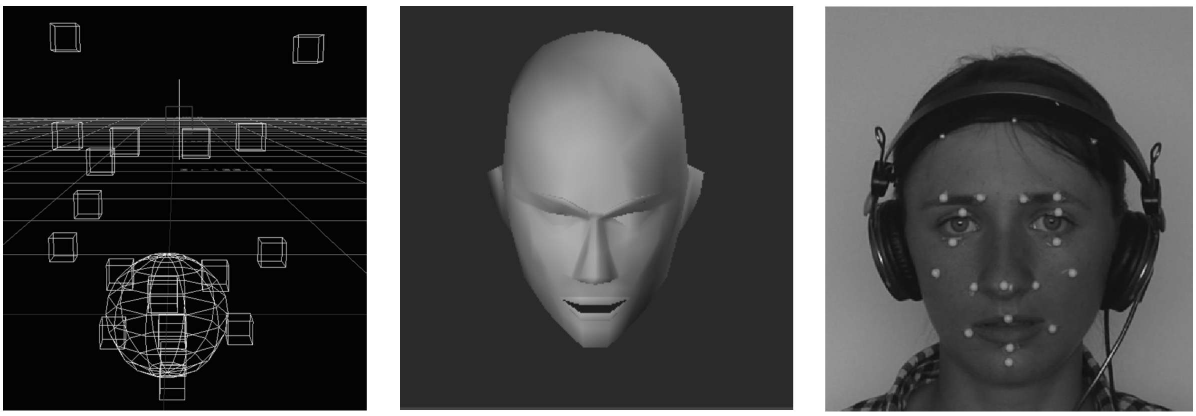


a) b)

Figure 6. Shot from the video sequence (a) and the resulting trajectories of the markers in the software package Autodesk Matchmover (b)

Following steps were made to obtaining synthesized facial expressions. Initial data in the format of three-dimensional trajectories of the points was normalized relatively to the coordinate system. Then, among all the points, several points (three in this case), which did not change the relative position and belonged to a cloud of markers, were found. Based on these points an object “rigid body” was created. The resulting object, particularly its center, was used to create a new reference system (cloud of markers is fixed relatively to the system). The reference system we used to eliminate rotations and tilts of a simplified model of a face. Then, each control point of the simplified model was related to similar point of the marker cloud. These points control the state of facial expressions on a simplified model of a face on each frame of sequence.

These results, when compared with the original video, allowed making a conclusion about a high level of similarity between animated three-dimensional models of facial expressions and facial expressions on the face of a real person captured on a video.



a) b) c)

Figure 7. Comparison of images: cloud of markers a), simplified animated 3d model of a head b) and the original frame of used video c)

Conclusion

The results of this study showed that the specific algorithms should be applied to solve the narrow class of problems. Not all computer vision algorithms are suitable for tracking markers on a human face. Part of algorithms (corner detection algorithms and algorithms based on them) perceives more features than needed and considers them as one. At the same time, a number of algorithms copes with this problem; however, taking in to account the problem of the errors caused by changes of the video settings (brightness variation, noises, etc.), they can be used only with some restrictions.

Computer vision algorithms (including algorithms of 3d reconstruction), used during analysis of the real data and modeling. have shown next features: 1) the reference points method used to study facial expression does allow to calculate accurately changes in some of the points on a face; 2) data obtained by these methods allow to transfer the results to the three dimensional model of a human head and calculate changes for whole points of a face.

Results obtained in this study are considered for further research studies: 1) to simulate facial expressions in sign language by 3d sign language avatar [Krak_3, 2012], and 2) to identify the facial expressions on a known set of facial expressions. It is also proposed to expand the studied set of facial expressions and their components to a larger set in order to cover some of the facial expressions that are specific for Ukrainian sign language.

Bibliography

[ACTS] Web site: http://www.zjucvg.net/acts/acts.html

[Audacity] Web site: http://audacity.sourceforge.net

[Autodesk Matchmover] Web site: http://download.autodesk.com/us/maya/2014help-matchmover/index.html

[Autodesk Motionbuilder] Web site: http://www.autodesk.com/products/motionbuilder/overview

[Blender] Web site: http: //www.blender.org

[Easycap facemotion] Web site: http://www.easycapstudio.com/

[Eyeon Fusion] Web site: http://www.eyeonline.com

[Forsyth, 2004] Forsyth David A., Ponce Jean. Computer Vision: A Modern Approach. – Moscow: Wilyams, 2004 – 928 p. (in Russian)

[Hanke, 2002] Hanke T., Langer G., Metzger C. et al. Interface Definitions, ViSiCAST Deliverable D5-1. – Hamburg, Insitute of German Sign Language and Communication of the Deaf, 2002. – 74 p.

[ImagineerSystems Mocha] Web site: http://www.imagineersystems.com/products/mochapro

[Krak_1, 2012] Krak Iu.V., Ternov A.S., Kuznetsov V.O. System of mimics description for sign language // Artificial Intelligence. – 2012. №4. – P.172-182. (in Ukrainian)

[Krak_2, 2012] Iu. Krak, A. Ternov, V. Kuznetsov, Yu. Barchukova. Creating audiovisual database of nonmanual components for sign language. Eleventh All-Ukrainian International Conference on Signal/Image Processing and Pattern Recognition - UkrObraz'2012. October 15–19, 2012, Kyjiv, Ukraine. Conference Proceeding. – Kyjiv, 2012. – P.7-10. (in Ukrainian)

[Krak_3, 2012] Iu. Krak, M. Lisnyak, A. Ternov. Rozrobka osnovnih instrumentiv kompyuternoi animacii dlya pobudovi sistemi sintezu gestovoi movi. International Conference: Intellectual Systems for Decision Making and Problems of

Computational Intelligence (27–31 may 2012. Kherson, Ukraine). Conference Proceeding. – Knerson:KNTU, 2012. – P.104-105. (in Ukrainian)

[Miller, 2011] Miller F.P., Vandome A.F., McBrewster J. Facial Action Coding System. – Beau Bassin, International Book Marketing Service Ltd, 2011. – 80 p.

[OpenCV] Web site: http://www.opencv.org/

[Panin, 2011] Giorgio Panin and Peter Ratner. Model-based Visual Tracking: the OpenTL Framework. – John Wiley & Sons Inc., New York, 2011 – 318 p.

[Punchcard VideoTrace] Web site: http://punchcard.com.au/

[Shapiro, 2006] Shapiro Linda G., Stockman George C. Computer Vision. – Moscow: BINOM, 2006 – 752 p.(in Russian)

[SSOntech SynthEyes] Web site: http://www.ssontech.com/synovu.htm

[Sutton, 2002] Sutton V. Lessons in SignWriting V. – La Jolla, Center for Sutton Movement Writing, 2002. – 195p.

[Vicon Boujou] Web site: http://www.vicon.com/boujou/

[VisCoda VooCat] Web site: http://www.viscoda.com/

[VisCoda VooDoo] Web site:http://www.viscoda.com/index.php/en/products/non-commercial/voodoo-camera-tracker

Authors' Information

Iurii Krak *– V.M.Glushkov Cybernetics Institute of NASU, senior scientist krak@unicyb.kiev.ua, yuri.krak@gmail.com*

Anton Ternov – *V.M.Glushkov Cybernetics Institute of NASU, scientist, anton.ternov@gmail.com*

Vladislav Kuznetsov – *V.M.Glushkov Cybernetics Institute of NASU, post graduate student, Vlad.Kuznetsov@gmail.com*

address: 40 Glushkov ave., Kiev, Ukraine, 03680