MPEG-4 Synthetic Video in real implementation

V. Kuriakin, T. Firsova, E. Martinova, O. Mindlina, A. Pleskov, K. Rodyushkin, V. Zhislina
Intel Nizhny Novgorod Laboratory

Abstract

MPEG-4 is the international coding standard developed for information transfer via low bit-rate communication channels. This article is the report on an experimental implementation of full-automatic pipeline MPEG-4 Synthetic Video Facial Animation (Simple Profile). This REAL-TIME pipeline includes the automatic detection, encoding, network transfer, and decoding of the facial animation parameters (FAPs) as well as decoder proprietary face model animation and rendering.

To obtain the animation parameters the authors developed the methods of automatic facial region detection combined with the recognition and tracking of the feature points for eyes, eyebrows and mouth (the inputs for further FAPs calculation). For that the algorithms of adaptive color segmentation, correlation analysis, optical flow and multilevel vector analysis were implemented. Model-independent 3D face model animation method based on the FAPs' regions of influence detection upheld by the feature points information was used.

Rendering based on OpenGL is synchronized with audio stream and includes eyelashes and "expression wrinkles" modeling. The animation results detailed in the article are based on realistic video data input sequences.

Keywords: MPEG-4, FAPs, Facial Animation.

1. INTRODUCTION

In the recent years "talking heads" have become increasingly popular in the network community [1], [2], [3]. The realization of the potential of the devices that transmit, receive, and process information is much faster than the transmission channel bandwidth expansion. Low power consumption is also very essential for mobile devices. Video conferencing and video phoning are most perspective applications: in fact, the current MPEG-4 standard supports speech coding and transmission with the bit rate from 2 Kb/s (HVXC standard), and synthetic video coding through FAPs also with the bit rate of 2 Kb/s. These bit rates open great opportunities for "talking heads". Certain specifics in human perception tell us about the plausible success of the new generation of games and teaching programs that use as an interface a "talking head". From other side the rich-media processing is developed, opening the possibilities for interactive video with complex scenes. A recent rating analysis has shown a tendency for a "talking head" website to have a higher ranking among its "headless" peers [4].

This article discusses the synthetic video pipeline implementation that fully conforms to MPEG-4 specifications [5]. According to the standard, a human head is a synthetic visual (3D face) object, and its representation is based on VRML standard [6]. In fact MPEG-4 standard specifies the encoding and decoding of FAFs, and says nothing about face features recognition procedures which should be performed before encoding nor about face animation which should be executed after decoding of FAPs from bit stream. Recognition module transforms the input video sequence into a sequence of animation parameters, the encoder compresses the data; the resulting data stream has to be processed by the decoder to provide FAPS for animation module.

For common success of synthetic video the quality of recognition and animation are essential. Therefore, first, we have developed specific methods and algorithms for automatic facial region localization and feature points recognition and tracking in the input video sequence. This is an understandably challenging problem, given the precision and accurateness required to correctly select corresponding feature points for a specific animation task. Second, we developed 3D face model animation procedures based on FAPs, as well as related techniques for visualization, and multiplexing of natural audio and video streams with synthetic video. Special demo applications were developed to showcase the pipeline.

2. FACE MODEL

Each MPEG-4 decoder should have an own face model animated by a stream of input FAPs. Our 3D face model is based on triangular meshes, like the majority of head models used for animation. We applied texturizing for enhanced realism.

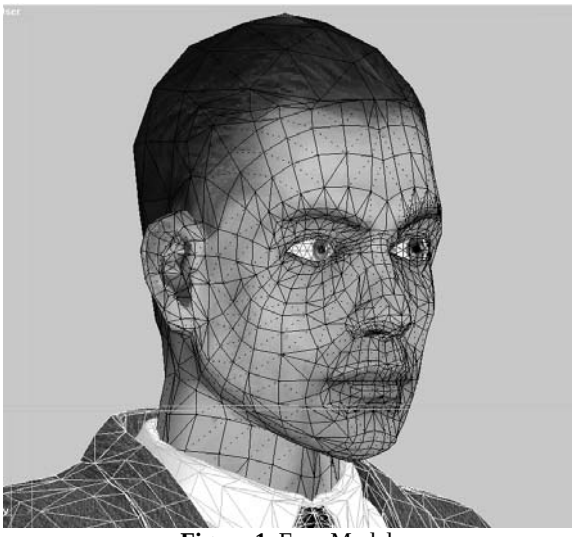

Figure 1. Face Model

The original model was selected from a set of 3D Studio Max built-in models and modified. We added the teeth, tongue, mouth cavity, and shoulders and changed the geometry of eyes and skin to conform to the MPEG-4 specification. Currently the Face Model Scene Graph includes a group of standard-conforming seven objects and an additional objects for shoulders for enhanced realism: 4,000 vertices and 7,000 triangles in total. All model objects are texturized with the total texture map volume of about 2Mb.

3. TEXTURE CREATION

The skin texture was built using a well-known method [21] based on combining two orthogonal photographs: the left view and the front view taken with a PC camera, while the right view is mirrored from the left view.

After all the piecewise lines passing feature points in each photograph were selected the three images are joined together along feature lines built interactively by the application and shown on fig.2. The feature points joined by feature lines are: the apex of the skull, the point on the forehead and the scalp border, the outward point on the eyebrow, the apex between the ear and the face, etc. The front view remains unchanged, while the left and the right views are deformed to match corresponding feature lines on the front view. Unfortunately, no matter how carefully the shooting conditions are controlled one cannot avoid "stitches" in the resulting image. An effective means to remove the stitches is the multiresolution technique: after a number of appropriate deformations the three images are merged together according to pyramidal decomposition based on Gaussian operator (fig. 3).

Figure 2. Initial images

Figure 3. The generated texture image

4. FACE RECOGNITION AND TRACKING

The most complicated task in MPEG-4 Synthetic Video is the original video sequence based FAPs computation. Owing to that the development of our application was concentrated around devising and implementing the algorithms of automatic facial region detection, recognition and tracking of the basic feature points and the borders of the eyes, eyebrows, and mouth used in further computation of the FAPs.

A new face detection method based on histogram adaptive color segmentation was developed to localize the facial region in the input frames. Unlike most other face detection approaches based on color segmentation our method does not require the labor-intensive learning stage as soon as the reference skin color histogram is adapted automatically throughout the input video frame processing. Among the numerous approaches towards basic feature points recognition and tracking one can name various methods that use either filtering [7-8] or optical flow [9-12] or deformable templates [13] or color distribution models [14] or template matching [15-16] as an underlying procedure. But the practice has shown than neither of the above alone can guarantee an adequate precision in localizing feature points.

The systems combining several methods have already appeared subpixel [17], but a recent achievement in facial expression analysis considerably undermines their meaning as soon as it does not require subpixel precision in localizing feature points. Alongside optical flow and filtering procedures intended to increase accurateness in localizing and tracking eyes, eyebrows and mouth our system implements a number of heuristic methods upheld by intensity, color and edge information as well as the a priori knowledge of the objects shape and the multilevel vector analysis.

4.1 Feature Points Recognition

The process of face segmentation, facial feature points recognition and tracking under the system discussed here can be performed in two principal conditions: preprocessing and tracking; each condition associated with a number of recognition and tracking procedures. The flowchart of the method is displayed in Figure 4.

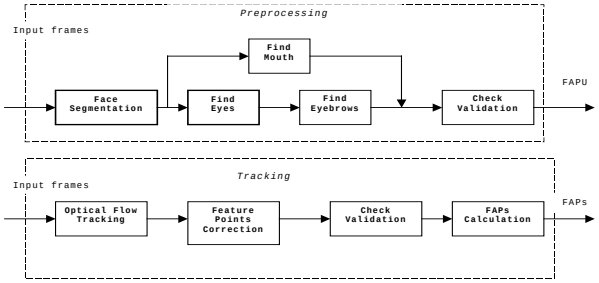

Figure 4. The general block diagram of the FAPs calculation.

4.1.1 Preprocessing

For each video frame input the preprocessing condition solves the following tasks: face segmentation, finding eyes, finding eyebrows, finding mouth, face calibration for further FAP calculation (FAPU, see Figure 5) together with check validation. After the face has been localized in the image the search for each group of feature points is confined to its proper area. On condition that a subsequence of frames returns valid results in feature points recognition the system switches to tracking.

To localize the facial region the adaptive color segmentation algorithm was developed. The algorithm detects the single connected component of pixels that most likely belong to the face (face mask). The data used here are the reference hue-saturation complexion histogram and the one built on the input frame. The distinctive feature of the algorithm is the adaptation of the reference histogram appealing to the previous frame segmentation.

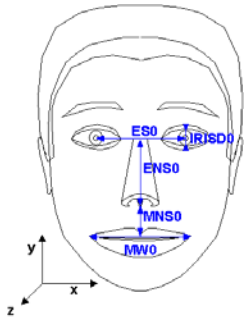

Figure 5. Facial Animation Parameters Units.

Applying convolution to the intensity and gradient fields each masked with the rectangular masks built on the eye model detects the position of the eyes. The detection concerns the centers of the eyes only, while the size is derived automatically, given the distance between the centers.
Image intensity and gradient information is also used to detect the position of the eyebrows. First, the intensity binary fields are scanned for the contours of both of the eyebrows that give the maximum total gradient alongside the contour. When found, each of the contours in turn is scanned for the three characteristic points in accordance with MPEG-4 specified neutral face feature points.
The preprocessing analysis of the binary level contours confined to the assumed mouth position area only gives the location of the corners of the mouth. The exact contour candidates are sifted out for the best fit to the selection criteria that include location, orientation, size and oblongness (squared perimeter to surface ratio) and once a fit is found the computation determines the exact mouth corners coordinates.

4.1.2. Tracking

The tracking also includes frame sequence processing but unlike preprocessing feature points recognition is carried out by different algorithms in the neighborhood of each feature point on the previous frame. Precise FAP calculation that follows also requires a higher (usually subpixel) measure of accurateness. The system implements a full number of algorithms to automatically detect each of the feature point groups while the final choice depends on the image illumination and input quality. Just as in preprocessing once a frame is processed the system runs the validity check for the tracking results. Failing to find the correct location of a feature point the system switches back to preprocessing of the next frame.
The optical flow procedure first runs a draft detection of the feature points in the current frame; for our system that brings up the centers of the left and the right eyes and the two corners of the mouth. To estimate the orthogonal transformation parameters in the image plane (shift, angle, and scale) a 43-point mesh is spread over the neighborhood of the eyes and the corners of the mouth. Pyramidal version of the optical flow [22] will then determine a new current image location for each mesh point. Transformation parameters are calculated so that when applied to the original mesh points they best match (in terms of standard deviation) the points found by optical flow. Given the transformation parameters the system proceeds to the calculation of the new positions of the centers of the eyes and corners of the mouth in the current frame.
The centers of the eyes' coordinates are corrected according to the analysis of the vertical gradient field in the local neighborhood of the respective points calculated at previous step; the size of the neighborhood determined by the accurateness taken on in the optical flow. The vertical gradient field is built as the difference between the max and the min values found for each pixel in a [2x1] window. Next step after the correction is the eye openness estimation based on the horizontal gradient field or the difference between the min and the max values found for each pixel in a [1x3] window.
Feature points for eyebrows are detected within a previously corrected eyes' position area like in preprocessing.
Mouth corners correction consists in precising their actual position in the neighborhood of the respective points found at the previous step. Other procedures used to improve the accurateness of the results include contour analysis for the binary levels of the current frame intensity plane V as well as intensity and saturation analysis. All the procedure output data received hitherto is combined to calculate the final corners of the mouth positions. Upon completion the system estimates the upper and the lower lip bounds while the inner bounds are built on the intensity and saturation pixel values along parabolas resting upon the found corners of the mouth in the current frame. The search for the outer lip bounds is based on the contour analysis aimed to find the best-fit candidates. Tracking results are shown on Figure 6.

Figure 6. Feature points tracking.

4.2. FAPs Calculation

The coordinates together with mouth and eyes parameters found after tracking on the first and the current frames are then used to calculate the basic FAP values under MPEG-4 specifications. Currently the system supports automatic detection of 27 basic FAPs used in lip, eye, and eyebrow animation and imaging the three turnings of the head.
In order to decrease the noise due to coordinate and parameter inaccuracy prior to encoding and mpeg streaming the FAPs are subject to fragmentary linear approximation and Gaussian filtering.

5. FAPS CODING

There are two major ways to encode a FAP stream under MPEG-4:

- Separately code each frame using quantization and arithmetic decoding of FAPs.
- Run coding for a sequence of 16 frames applying one-dimensional discrete cosine transform to the whole set of FAP values using quantization and Huffman coefficient coding.

Figure 7 and Figure 8 show the schemes of FAP decoding.

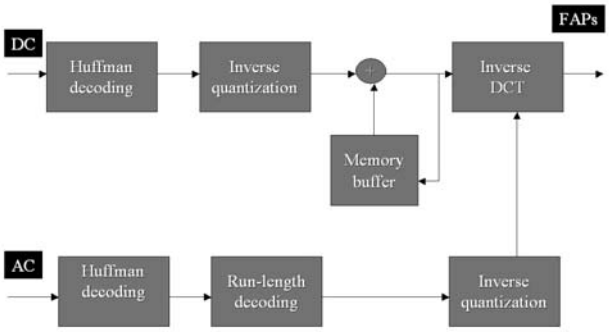

Figure 7. DCT and Huffman decoding

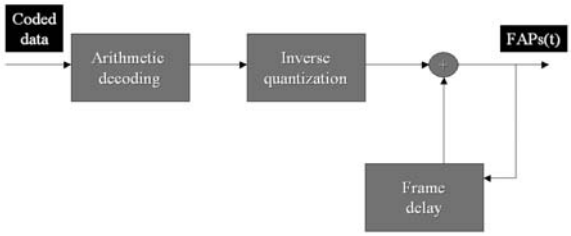

Figure 8. FAPs arithmetic decoding

Both allow two coding modes: intra and predictive. The encoder (decoder) in the intra mode uses only currently coded information while in the predictive it considers previously coded data, too, as a means to predict the current thus actualizing residual coding. Compared to the intra mode the predictive one gives a higher compression ratio.

FAP coding functions used in the system are based on discrete cosine transform. Both intra and predictive modes are available as well as viseme and expressions coding. Viseme and expressions are two high-level animation parameters each coded in its own way. Particularly, expressions can be derived from low-level FAPs and in coding contract just to intensity information. The system decoder provides full support of expression modeling. Altogether under the standard there are 6 mimic expressions (joy, anger, distaste, sadness, fear, and surprise) and 14 visemes.

The demo application illustrates FAP decoder performance, synthetic video stream, sound, and real video synchronizing as well as decoder's ability to reproduce mimic expressions.

The application allows simultaneous viewing of the recognizing and the animating applications. Here, after face motion recognition and FAP stream and speech coding the recognizing server application transfers the data via network to the client application. The client application then decodes the received information and animates the model.

6. ANIMATION

Two ways to animate a 3D polygonal texture model are: one, convey all texture templates to decoder assuming that all local face changes must be imaged by means of texture map only, which blows up the traffic. Two, assume a stationary texture map but an alterable 3D surface with admissible geometric deformations brought about by FAP stream or, in other words, assume the MPEG-4 standard.

Polygonal deformation in turn has a number of approaches that above all gives us pseudomuscular [18], [19] and heuristic models [20]. The pseudomuscular model assumes a simplified description of muscles involved in mimics, as a detailed anatomy would lead to computational complexity, which isn't compatible with real-time performance.

In our heuristic model we assign each FAP a set of face model vertices each with an offset in case of nonzero FAP value.

Developed under Simple Facial Animation Object Profile animation in our application is determined solely by the FAP stream received from the encoder. The face model under MPEG-4 is defined for an emotionally neutral face with all muscles relaxed, mouth closed, and eyes wide open while FAPs determine the offset of the feature points, turning of the head and eyes in reference to the neutral position.

As soon as FAPs apply to different facial sizes and proportions FAP magnitudes are defined as relative units (FAPUs). Each FAPU is a certain part of the distance between the feature points on the neutral face model.

After FAP decoding its magnitude is transformed into the units of the animated model by means of a suitable FAPU. Then the respective feature point is moved to the offset distance from the neutral position.

Our animation software consists of two parts:

- motion modeling (offline),
- framewise FAP interpretation and model deformation.

Each FAP is assigned its region of influence in further motion modeling. The distinctive feature of our face animation algorithm is the model independence as soon as each region's bounds are based on feature points coordinates.

The whole set of FAPs used in motion modeling falls into:

- scene object transformations,
- "sliding" along mesh surface,
- vertex group transformations.

In preprocessing those of the FAPs that are irrelevant to object transformations are each assigned a set of vertices within its region of influence while the system is still offline. Each vertex offset magnitude in reference to the feature point offset within the region is limited by the vertex weight calculated as an exponential weight function.

This information is stored to FaceDefTable, which is an automatic definition of the rules for face model animation.

On-Line animation uses these rules in order to calculate new point's coordinates.

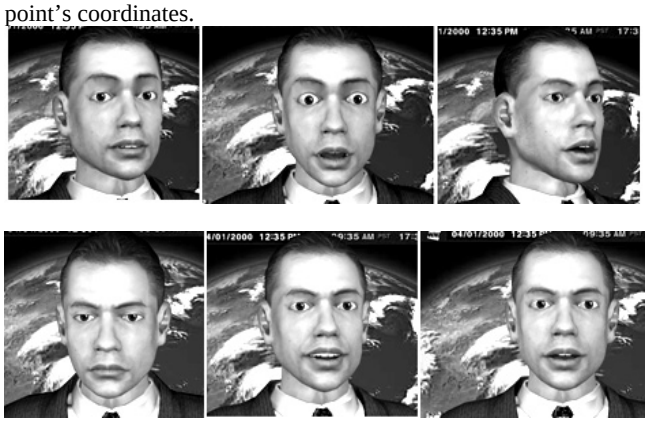

Figure 9. Synthetic video fragments.

7. RENDERING

Rendering is the final stage in the pipeline. In the input each frame is a 3D face model deformed in accordance with its FAPs while the output is a synthetic video sequence showing a texturized

"talking head" projected on the screen and playing a synchronized audio stream. The background can be either a static image or a real video sequence. The output productivity remains unchanged if the real video background is switched to a static one and cuts 25 frames per second for synchronized audio stream visualization and 30 frames per second for consecutive FAP stream visualization sampled on Pentium® III-700, 256M RAM, NVIDIA RivaTNT, Windows 2000.

Rendering is based on 1.1 no extension version of OpenGL, advantageous in its platform and OS independence.

To add to the realism of the model while rendering the system runs 3D eyelash modeling and imaging and adds expression wrinkles. These, for example, if a smile is concerned, run from the nose to the corners of the mouth.

Pre-rendering includes eye contouring, eyelashes generation, assigning expression wrinkles to appropriate lines as well as defining the normal for each point, which is used in further lighting calculations and expression wrinkles modeling.

In actual rendering the eyelashes are texturized with an eyebrow fragment borrowed from the original texture map and rendered with IndexedLinesSet. When rendering the eyelashes a proper function is called for a big and a small model providing the most natural imaging.

Expression wrinkles modeling is based on our own interpretation of the "bump mapping" principle that is to model slight surface irregularity by changing the normal assigned to it fully preserving the geometry. FAP due skin contraction/stretching is transformed to perpendicular direction offset and given the presence of diffuse or specular lighting produces the effect of expression wrinkles.

Figure 9 shows the fragments of synthetic video sequence developed by our application. Figure 10 illustrates the work of the two-window application playing synthetic and real video simultaneously.

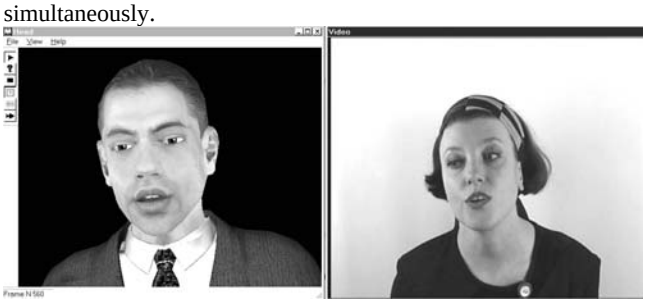

Figure 10. Two-window application mode.

8. CONCLUSION

Full-automatic MPEG-4 facial animation pipeline for polygonal face model was developed.

Its current version provides a number methods for automatic facial region detection, feature points recognition and tracking together with further FAP calculation based on real video sequence.

FAP stream coding is implemented in full conformity with MPEG-4 standard. The team has designed and implemented the model-independent method for automatic computation of the polygonal face model animation. The application for computing and encoding a real video-based FAP sequence into a bit stream has the output performance of 20-22fps while the application designed to decode the FAP stream, animate and render the model has the performance of 25 fps.

We further intend to implement the Calibration Facial Animation Object Profile, automatic texture mapping and fitting, and improve program animation and automatic real video based FAP calculation.

9. REFERENCES

[1] http://www.famous.3d.com/

[2] http://www.cselt.it/ufv/joe.html

[3] http://www-dsp.com.dist.unige.it/~pok/

[4] I. Pandzic, J. Ostermann, D. Millen "*User Evaluation: Synthetic Talking Faces for Interactive Services*", The Visual Computer Journal, Vol. 15, No. 7-8, pp.330-40, Springer Verlag, 1999.

[5] SNHC, "*INFORMATION TECHNOLOGY – GENERIC CODING OF AUDIO-VISUAL OBJECTS Part 2: Visual*", ISO/IEC 14496-2, Final Draft of International Standard, Version of: 13, November 1998, ISO/IEC JTC1/SC29/WG11 N2502a, Atlantic City, October 1998.

[6] The Virtual Reality Modeling Language - http://www.web3d.org/Specifications/VRML97/

[7] H.P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan, "*Multi-Modal System for Locating Heads and Faces*", Proc. 2nd Int. Conf. Automatic Face and Gesture Recognition, IEEE Computer Soc. Press, 1996, pp.88-93.

[8 Yow K. C. Cipolla. R., "*Finding initial estimates of human face location*", Proc.2nd Asian Conference on Computer Vision, Singapore, volume 3, 1995, pp. 514-518.

[9] Black, M. J.,Yacoob, Y., "*Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion*", Proc. 5th International Conf. on Computer Vision, ICCV'95, Boston, MA, June 1995, pp 374-381.

[10] Eisert P., Girod B., "*Analyzing Facial Expressions for Virtual Conferencing*", IEEE Computer Graphics & Applications: Special Issue: Computer Animation for Virtual Humans, September 1998, pp. 70-78.

[11] DeCarlo D. Metaxas D., "*Optical Flow Constraints on Deformable Models with Applications to Face Tracking*", International Journal of Computer Vision, 38(2), July, 2000, pp. 99-127.

[12] Gokturk S. B, Bouguet J., Grzeszczuk R., "*A Data-Driven Model for Monocular Face Tracking*", Proc.8th International Conference on Computer Vision, Vancouver, Canada, July 2001.

[13] A. Yuille, P. Hallinan, D. Cohen, "*Feature Extraction from Faces Using Deformable Templates*", International Journal of Computer Vision, V.8:2, 1992, pp.99-111.

[14] Horprasert Y., Yacoob Y., Davis L., "*An Anthropometrics Shape Model for Estimating Head Orientation from a Monocular Image Sequence*", Proc. 3rd International Workshop on Visual Form, 1997, pp.247-256.

[15] L. Bala, K. Talmi, J.Liu, "*Automatic Detection and Tracking of Faces and Facial Features in Video Sequences*", Picture Coding Symposium, September 1997.

[16] Lievin M., Luthon F., "*Lip Features Automatic Extraction*", Proceedings of the IEEE Conf. on Image Processing, ICIP'98, Chicago, USA, vol. 3, oct. 1998, pp. 168-172.

[17] Tian Ying-Li, Kanade, T., Cohn, J. F, "*Recognizing action units for facial expression analysis*", Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume: 23 Issue: 2, February 2001, pp. 97 –115.

[18] Y. Lee, D. Terzopoulos, K. Waters, "*Realistic Modeling for Facial Animation*", Computer Graphics (Proc. SIGGRAPH'95), pp.55-62, 1995.

[19] Won-Sook Lee, Nadia Magnenat-Thalmann,"*From Real Faces To Virtual Faces: Problems and Solutions*" Proc. 3IA'98, Limoges (FRANCE), 1998, pp.5-19.

[20] F. Lavagetto, R. Pockaj, "*The Facial Animation Engine: towards a high-level interface for the design of MPEG-4 compliant animated faces*" IEEE Trans. On Circuits and Systems for video Technology, Vol. 9, no. 2, March 1999.

[21] Won-Sook Lee, Nadia Magnenat-Thalmann,"*Head Modeling from pictures and Morphing in 3D with Image Metamorphosis based on triangulation*", Proc. CAPTECH'98 (Modeling and Motion Capture Techniques for Virtual Environments), Geneva, pp. 254-267, 1998.

[22] B.D. Lucas, T. Kanade, "*An iterative image registration technique with an application to stereovision*". Proc. 7th Int. Conf. On Art. Intell. , 1981, pp.674-679.

About the author

V. Kuriakin – Valery.Kuriakin@intel.com
T. Firsova – Tatiana.Firsova@intel.com
E. Martinova – Elena.Martinova@intel.com
O. Mindlina – Olga.Mindlina@intel.com
A. Pleskov – Alexander.Pleskov@intel.com
K. Rodyushkin – Konstantin.Rodyushkin@intel.com
V. Zhislina – Victoria.Zhislina@intel.com