Proceedings of ALGORITMY 2009
pp. 372–381

A NOVEL VIDEO COMPRESSION SCHEME BASED ON KINETIC DELAUNAY TRIANGULATION*

TOMÁŠ VOMÁČKA† AND PETR PUNCMAN‡

Abstract. A need for different ways of video representation comes from both the entertainment and the research areas. Our method uses a kinetic Delaunay triangulation (KDT) which is used as an interpolation basis for resulting video frames. KDT is a time-dependent data structure intended to retain the Delaunay property despite the movement of the underlying points. For managing the triangulation, we use the method of continuous legalization of the triangulation by computing the times of topological events (previously described by Gavrilova, Roos, Rokne and others). By handling these events we are able to keep the triangulation Delaunay for its whole lifetime.

We propose a new method that utilizes triangulation construction by collecting samples from selected intra-coded frames, triangulation moving by precomputed optical flow vectors gained by block matching algorithm in inter-coded frames and decoding data from samples by means of barycentric interpolation and feature based warping.

Key words. Video compression, motion tracking, kinetic Delaunay triangulation

AMS subject classifications. 68U5, 68U10, 94A08

1. Introduction. The way of the digital video representation plays a crucial role when determining the intended quality, compression ratio or even the target application of such a video record. The most recent techniques in this area often use not only a reduction of the spatial redundancy but they also try to reduce the temporal one by using a wide variety of techniques, including both the loss and lossless compression of the video frames, conversion to the frequency domain and back and others. The frames of a video record may be seen as a set of unconnected images or the similarity of the consecutive frames may be exploited by some methods. Our work investigates a possibility of using the kinetic data structures, namely the kinetic Delaunay triangulation, in connection with a method for a spatial redundancy reduction. These structures retain some of their special properties despite the point movement and thus enable us, when used together with techniques for motion compensation, to track relevant points in order to compensate differences between consequent frames and thus maximize quality at low bitrates.

This paper is organized as follows: Chapter 2 describes the state of the art in both the areas of video compression and representation and kinetic Delaunay triangulation. More information on the topic of the video coding by using the kinetic Delaunay triangulation is given in Chapter 3. The kinetic Delaunay triangulation itself is described in Chapter 4. The results of the experiments with our method are presented in Chapter 5 and the whole work is concluded in Chapter 6.

*This work was supported by project No. KJB10147071.

†Faculty of Applied Sciences, University of West Bohemia, Univerzitní 22, 306 14 Pilsen, Czech Republic(tvomacka@kiv.zcu.cz).

‡Petr.Puncman@seznam.cz

2. State of the Art.

2.1. Video Compression. The most widely used present algorithms for lossy video compression concentrate either on the intra-coded frame modifications in frequency domain such as the discrete cosine transform (MPEG1-2, DV, MJPEG, H261-4) and the discrete wavelet transform (MJPEG 2000, Intel Indeo 5) or on the vector quantization (Cinepak, Sorenson Video). Inter-coding, if there is any, is often handled by the block matching algorithm (BMA). It presents the simplest way of obtaining the motion vectors of the corresponding macroblocks (16×16 pixels group of four 8×8 pixel blocks - a basic element for video compression) over the whole frame (e.g., in Fig. 2.1 on the right side we can see a block B which has been found to be similar to the block $\hat{B}$ in the same window in the next frame). The resulting video is then partitioned into picture groups containing intra-coded I-frames, inter-coded P-frames and optional bidirectionally inter-coded B-frames retaining their mutual relations (see Fig. 2.1 on the left).

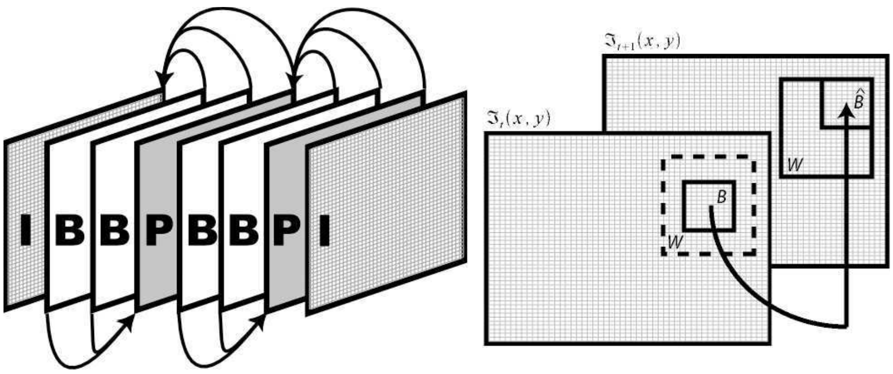

FIG. 2.1. *Typical compression scheme (left). Block matching principle (right).*

The principle of the abovementioned intra coding is that each frame is considered to be a standalone image and is processed accordingly. Thus each I-frame is encoded without using any information from the previous (or the following) frames in the video sequence. On the opposite, the inter-coded and bidirectionaly inter-coded P-frames and B-frames use their similarity to the surrounding frames (in one or both directions) for the encoding. In most cases, simply put, BMA searches for such movement vectors for each macroblock that, if applied to the macroblock in question, describes the following frame with minimal error.

The most common problems connected with classic video compresion schemes are the appearance of new block elements between two frames and a serious loss of detail at low bitrates. Also handling these video representations is a bit impractical, when there is a need to transform or interpolate a videosequence. The corresponding frames have to be decompressed and all their pixels processed. These drawbacks may be solved, in our case, by using an alternative triangulation-based video representation. Present solutions on this topic deal with the movement of 2D and 3D triangulations that represent synthetic objects in the video (see [3, 8]). Other approaches include the construction and movement of adaptive triangulations over the whole scene (see [14, 18]). The last group of approaches considers the videosequence to be a 3D object which is then tetrahedronised (see [15]).

2.2. Kinetic Delaunay Triangulation. The problem of maintaining the kinetic Delaunay triangulation has been addressed in various papers (see [1, 5, 6]) either as a theoretical standalone structure or as a part of applications such as collision detection system [5] or maintaining the spatial relationships between marine vessels [6]. Even though the mentioned papers describe the way of managing the kinetic data structures relatively very well, the required mathematical apparatus is often described only in general. So it is a known fact that in order to keep the kinetic data structure legal in two dimensions, one has to compute a certain number of polynomials of various degrees, but the way of doing so is often omitted. Because these polynomials are of the 4-th degree in general, the analytical solution is insufficient due to its low precision, so a suitable analytical method (or combination of methods) has to be used. In our previous work, we managed to design and implement one such method which uses Sturm sequences of polynomials [16, 17].

3. Video Processing by KDT. In our approach, we focus on the combination of the methods mentioned in [3, 8] and [14, 18] together with the abilities of KDT. That means that we perform both sampling of the pixels of the original image in coder and interpolation or warping of the full scene by using KDT. The idea of video representation by KDT is based on a creation and successive movement of a triangulation. In the first step, the relevant points and some random points have to be obtained from a frame which is considered to be intra-coded. These points then define the KDT until a new picture group is formed, starting with the next intra-coded frame. The second step means processing frames between the two following I-frames to form inter-coded frames by moving the points in the triangulation and reconstructing the frames from the current state of the triangulation. The movement of the points in KDT is defined by the vectors obtained during motion estimation. So the whole process may be described as follows (components of this process will be described in the following two sections):

Important point selection A set of important points is selected from the input image. KDT is created from these points.

Motion estimation Corresponding blocks centered around each vertex are found in consecutive frames. Their positions are then used to define motion vectors.

Movement of KDT The movement is initialized transforming the current frame into the following one.

- **Topologic events** Topologic events are computed and handled.
- **Stability improvements** Means for improving stability are introduced.

Video decoding A frame is reconstructed from the data stored in the KDT.

The use of DT is crucial for our method if we want to obtain usable compression ratio. It is vital to note that for every other type of triangulation we have to store not only the coordinates and color of each vertex but also an information on the surrounding topology. On the other hand in DT we may be sure that if no four points in the triangulation are cocircular, we will always reconstruct the same triangulation (independently on the algorithm used) without storing any additional topology data. As we have shown (see [12]) the compression may be expected to be meaningful (considering a method without any stored topology information) with up to 20% inserted points if we use a trivial compression method and up to 30% inserted points if a more sophisticated method which exploits delta coding is used.

3.1. Selecting the Points of Interest. To obtain valuable samples from the input frame, we mix edge points with random points at the ratio of 1:1 (the best ratio value was obtained from the performed ratio tests). Starting with an already given point count, we select the most suitable points from the frame by the methods of discrete convolution, selective thresholding and randomized selection. The resulting set of pixels then in the ideal case contains a predefined amount of points. These points include dense pixels on the edges, sparse pixels on the homogenous surfaces and uniformly distributed pixels everywhere else. For details on selecting the points see [12] as it is beyond the scope of this paper. Finally, KDT is created by incremental point insertion from the selected points.

3.2. Motion Estimation. Although the block matching algorithm often deals with macroblocks covering the whole frame, in our case we had to find which block in the next frame makes the best match for the block centered around each KDT point in the current frame. Each examined point then grows into a suitable search region W (see Fig. 2.1) which is intensively inspected. For an eveluation of the differences between the blocks B and $\hat{\mathrm{B}}$ we have used three different metrics: MSD (Mean Square Difference), MAD (Mean Absolute Difference) and PDC (Pel Difference Classification) - see [4]. These metrics allow us to recognize the corresponding blocks and thus obtain the needed motion vectors. Once all the vectors in a frame are known, velocities of their points in KDT are set (see Fig. 3.1).

3.3. Video Decoding. The previous steps provided us with the most important points in each frame and their inter-frame correspondency thus allowing us to compute the motion vectors for these points. By inserting the points obtained in each key frame into a KDT and moving them along the motion vectors we get a sequence of triangulation states that represent each frame between two keyframes. From these triangulations we must now decode the approximation of the original frame.

FIG. 3.1. *From left to right: the original KDT, motion vectors, compensation of KDT*

In the case of the intra-coded frames, we have to use an interpolation algorithm. At first we have to obtain a raster representation of the edges e_1, e_2, e_3 belonging to each triangle $p_1p_2p_3$ (see Fig. 3.2). For this purpose we use an implementation of Bresenham algorithm [2].

Successive computation consists of (among other equations) solving Eqn. (3.1) for all three triangle vertices. By solving this system of equations, we obtain the coefficients a, b, c which are then used to compute the intensities of general points inside the triangle.

$$
\begin{pmatrix} x_i + y_i + 1 \\ x_j + y_j + 1 \\ x_k + y_k + 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} z_i \\ z_j \\ z_k \end{pmatrix} \tag{3.1}
$$

where $[x_i, y_i], [x_j, y_j], [x_k, y_k]$ are the coordinates of the vertices of a triangle and z_i, z_j, z_k are the intensities of the corresponding pixels.

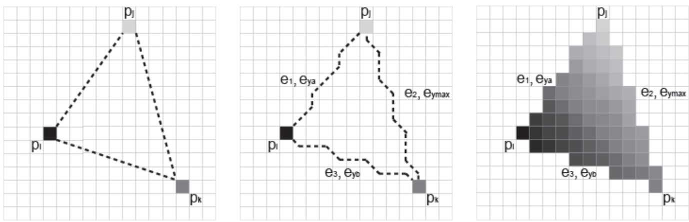

FIG. 3.2. *From left to right: vertices with associated intensities, edges by Bresenham algorithm, scanline fill and triangle interpolation*

In the case of inter-coded frames, we move KDT as described in Section 3. After then, either the aforementioned interpolation or feature based warping (see Fig. 3.3) may be applied.

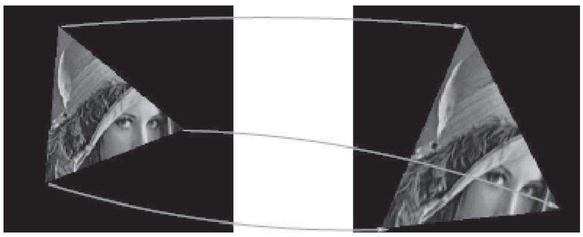

FIG. 3.3. *Triangle warping; taken from [10]*

In our case, the warping process takes the edges of a triangle in the current frame and corresponding transformed edges in the previous frame. Note that the triangle can change significantly from frame to frame. The task is to compute the intensities of the pixels within the transformed triangle. As we have a relatively accurate approximation of the intensities of all the pixels in the last intra coded frame, we perform the warping process for all the consecutive inter-coded frames after that frame. An incremental computation is also possible but it leads to notable loss of accuracy. We adopted the warping process for more line pairs as described in [19] gaining X' original pixel coordinates. The positions of the three points X'_i are then passed to weight function (3.2) with the weights w_i being proportional to the pixel-edge distance in order to produce more accurate pixel position X.

$$X = \frac{\sum_{i=1}^{n} w_i X'_i}{\sum_{i=1}^{n} w_i} \tag{3.2}$$

4. Managing the Kinetic Delaunay Triangulation. Delaunay triangulation $DT(S)$ over a set of points S is such a triangulation that fulfills the Delaunay criterion - no point in S lies inside a circumcircle of any triangle in $DT(S)$. However, due to the movement of the points in S, this criterion will sooner or later be broken. We have to compute the times when it is done (so called topological events) and during them perform such topological changes in the triangulation that will prevent the triangulation from breaking the Delaunay criterion (see further).

Overall functionality of the kinetic Delaunay Triangulation (described for instance in [16, 17]) may be divided into two steps - the preprocessing and the iteration. In the preprocessing step, the Delaunay triangulation of the points in their initial positions is created and the first topological event is computed for each pair of adjacent triangles by determining the nearest time their four points become cocircular. The following iteration step is then repeated as many times as required. During each of the iteration steps, the triangulation time is increased, some of the events are processed and new ones may be created (in our case this step is repeated for each frame of the video sequence).

4.1. Topologic Events and their Computation. As shown in [1, 5] and others, the topology of a triangulation with at least one point with nonzero velocity vector will have to change at a certain time instant. The reason for this topological changes is shown in Fig. 4.1. Let us suppose that only P_4 is moving, we may see that the empty circumcircle criterion of Delaunay triangulation is broken by this movement, thus we have to determine the time instant when it reaches the position marked as P_4' (the four points in the figure are then cocircular) and perform an edge swap (i.e., handle a topological event). We have to perform this edge swap because if the point moved further towards the location marked as P_4'', the Delaunay criterion would be violated.

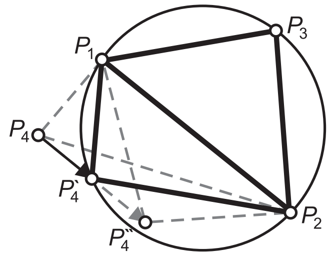

FIG. 4.1. *Triggering a topological event.*

To be able to determine the times of the topological events we have to compute the following equation:

$$\det \begin{pmatrix} x_1 & y_1 & x_1^2+y_1^2 & 1 \\ x_2 & y_2 & x_2^2+y_2^2 & 1 \\ x_3 & y_3 & x_3^2+y_3^2 & 1 \\ x_4 & y_4 & x_4^2+y_4^2 & 1 \end{pmatrix} = 0 \tag{4.1}$$

where $P_i = [x_i, y_i]$ are coordinates of the four inspected points.

The computation of the sign of this determinant is often referred to as the in-circle matrix test (see [7]) which is commonly used to determine the position of a point against a circumcircle of a triangle given by three points. If we state that the coordinates of the points $P_1, ..., P_4$ in Eq. (4.1) are functions of time, such as:

$$P_i(t) = [x_i(t), y_i(t)] \tag{4.2}$$

$$x_i(t) = x_i(0) + v_i^x \cdot t, y_i(t) = y_i(0) + v_i^y \cdot t \tag{4.3}$$

where $P_i(0) = [x_i(0), y_i(0)]$ are the initial coordinates of the point P_i and $v_i = (v_i^x, v_i^y)$ is the velocity vector of P_i, then we may expect that Eq. (4.1) is in the form of a polynomial of degree no greater than four. In order to solve this equation, we have designed and successfuly tested a hybrid numerical analytical method based on Sturm sequences of polynomials (see [13]), details are beyond the scope of this paper and may be found in [16, 17]. The main advantage of this method is that it allows us to discover the number and multiplicities of all roots on any subinterval of **R** thus preventing us from computing a polynomial with no usable roots.

4.2. Stability Improvements. Due to the fact that all the points inserted into the triangulation are representation of pixels in a grid and their velocities reflect their movement in this grid, we may easily encounter unwanted singular cases. These cases include for instance collisions of two moving points or number of points becoming cocircular at the same moment. Illustration of these singularities is given in Fig. 4.2.

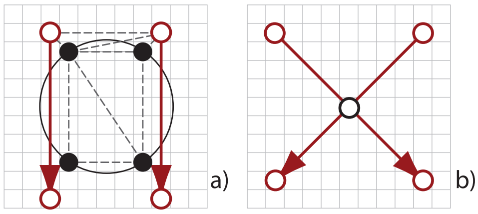

FIG. 4.2. *Singular cases for grid point movement. Left: six cocircular points, right: point collision*

Our experiments showed that there are two ways of removing these singular cases. We may either add a small random number to the velocities or to the initial coordinates of the points (or even to both of them). Only the moving points or all of the points in the triangulation may be altered in this fashion. These numbers must be of course small enough to be later eliminated by the rounding process without moving the points into wrong cells in the grid. We have tested both the mentioned types of randomization and various random values and finally we managed to determine the ideal combination - we used only velocity vector components randomization with uniform distribution of the random part and 10^{-3} maximum absolute value of the random addition.

5. Experiments and Results. The test application was implemented in C# and its purpose was to provide the results we needed for algorithm efficiency evaluation and for comparison with existing solutions. We wanted to test the main properties of our algorithm, such as the quality of the video which is indicated by PSNR, bitrate in bytes per frame and the overall performance of each step.

The tests were performed on three videosequences with the length of 100 frames, resolution of 176×144px and the length of the picture group set to 6. The sequence with the dynamic camera contains a lot of synchronized movement, the synthetic sequence contains a very large percentage of untextured and simplified surfaces and the talking head is mainly static. The quality criterion is controlled by the PSNR metric - see Eqn. (5.2).

$$MSE = \frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \|\Im(i,j) - \Im'(i,j)\|^2 \tag{5.1}$$

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \tag{5.2}$$

where $M \times N$ is the dimension of the image, $\Im$ is the original image and $\Im'$ is the decoded approximation of the original image.

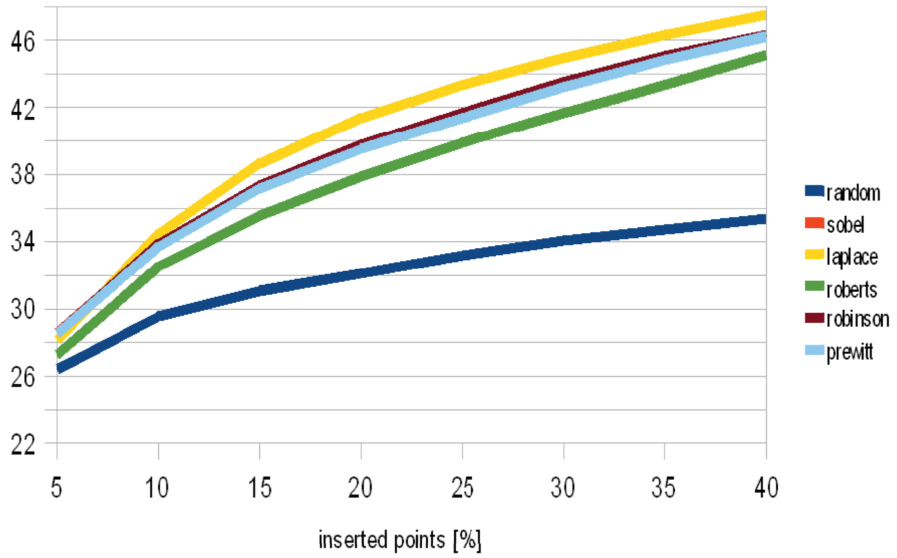

FIG. 5.1. *Dependency of PSNR on inserted points count*

In order to examine the quality of intra coding we studied an influence of mixed (50:50 edge:random) point selection against to solely random point selection. The set of chosen edge operators (random, Sobel, Laplace, Roberts, Robinson and Prewitt - for detailed information on these operators, see [9]) - see Fig. 5.1 - provides noticably better results than a random generator at the whole range of inserted points percentage. Note that acceptable PSNR value starts at 30dB.

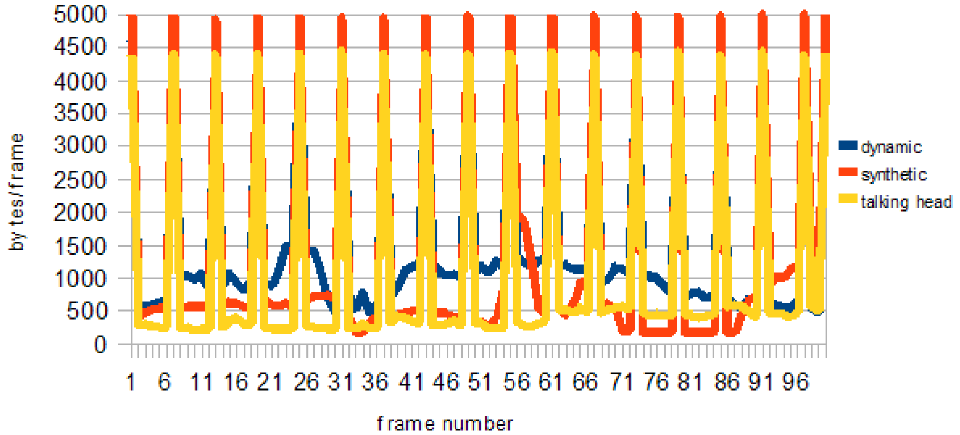

FIG. 5.2. *A Detailed Bitrate Behavior of Intra + Inter Coding for 5% Points*

The second comparison we made (see Fig. 5.2) shows the bitrate achieved in each frame of all the three test sequences. The size of an inter-coded frame is derived from the amount of the motion in the scene. At these bitrates we were able to provide compression ratio of around 20:1.

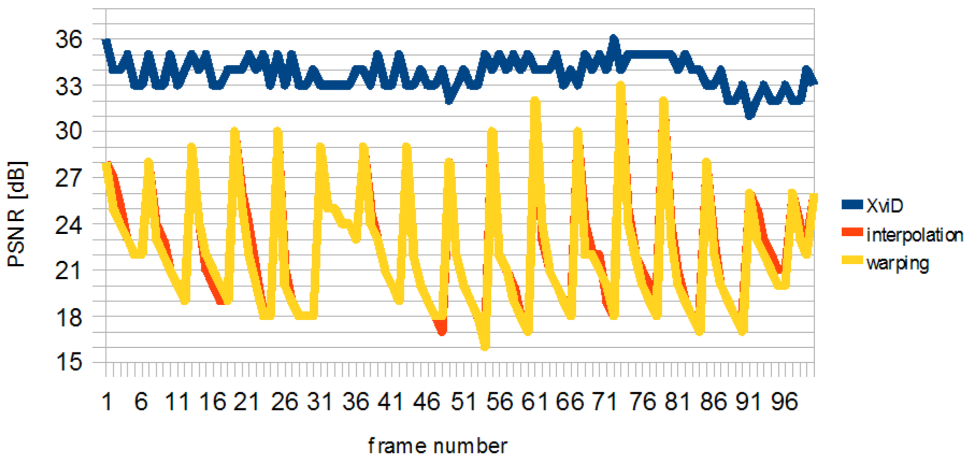

FIG. 5.3. *Intra + Inter Coding vs. XviD Quality Comparison*

In the third test (see Fig. 5.3), we compared the quality of our solution with XviD for the same coded output size. The initial amount of inserted points was 5% and the length of the pictures group was set to 6. The intra-coded frames provided quality nearly as good as XviD did, but inside the inter-frames quality dropped rapidly. Both techniques of decoding (interpolation and warping) were measured to be nearly equivalent, however, subjective comparison often prefers the warping prior to the interpolation because of triangular artifacts which may appear as a result of the interpolation.

The tests showed that if we use intra coding for the video frames only, all the used operators behave in a similar fashion and their quality response is in all the measured cases logarithmic (measured in PSNR). However, the best results were received for video sequences that contained large percentage of homogenous areas, such as rendered videos and talking head videos, because these types of video sequences are very suitable for coding and motion blur occurs very rarely, if at all.

6. Conclusion. Various requirements are often set for a new video representation method. These may include the ability to perform morphologic changes on the triangulation and receive the corresponding changes in the image matrix of the video, minimization of the bitrate and maximization of the quality. With these possible requirements in mind, we have designed and implemented a new method that allows us to encode the frames into a geometric form according to the primarily requested feature.

Inter-coding showed its major advantage in very low bitrates and good compression ratio. For 5 - 30% inserted points, we were able to achieve compression ratio from 20:1 to 4.5:1. Detailed tests and their results may be found in [12].

From the point of view of the kinetic Delaunay triangulation, the most important drawbacks (the lack of stability and the occurence of singular cases) of the application were succesfully removed by introducing the randomization of the velocity vectors of the moving points. However some performance issues are still left to be solved, especially in the area of computing the topological events, of which is a large percentage computed but not executed. Introducing some kind of nonlinear movement may also pose some advantage in this kind of application (for instance the movement along elliptic trajectories).

REFERENCES

[1] Albers, G., Guibas, L. J., Mitchell, J. S. B., Roos, T. Voronoi Diagrams of Moving Points. *International Journal of Computational Geometry and Applications*, 8(3):365-380, 1998.

[2] Bresenham, J. E. *Algorithm for Computer Control of a Digital Plotter*. IBM Systems Journal, Vol. 4, No. 1, str. 25-30, 1965.

[3] Chen, C. W., Zhang, YA. Q. *Visual Information Representation, Communication, and Image Processing*. CRC Press, 1999, ISBN 082471928X.

[4] Furht, B., Greenberg, J., Westwater, R. *Motion Estimation Algorithms For Video Compression*. Kluwer Academic Publishers, 1997, pp. 61-95.

[5] Gavrilova, M., Rokne, J., Gavrilov, D. Dynamic Collision Detection in Computational Geometry. *In 12th European Workshop on Computational Geometry*, pages 103-106, Munster, Germany, 1996.

[6] Goralski, I. R., Gold, C. M. Maintaining the Spatial Relationships of Marine Vessels using the Kinetic Voronoi Diagram. In *ISVD*, pages 84-90. IEEE Computer Society, 2007.

[7] Hjelle, Ø., Dæhlen, M. *Triangulations and Applications*. Berlin Heidelberg: Springer, 2006.

[8] ISO/IEC 14496-2:2001. *Coding of Audio-Visual Objects: Visual*. 2nd ed, 2001.

[9] Kohout, J. *Alternative Representation of Image Information, Tech report*. University of West Bohemia, Faculty of Applied Sciences, 2008.

[10] Marquant G. *Reprèsentation par Maillage Adaptatif Dèformable pour la Manipulation et la Communication d'Objets Vidèo*. Universit de Rennes, 2000.

[11] Parhi, K., Nishitani, T. *Digital Signal Processing for Multimedia Systems*. CRC Press, 1999, pp. 301–309.

[12] Puncman, P. *Použití triangulací pro reprezentaci videa*. Diploma thesis, University of West Bohemia, Pilsen, 2008.

[13] Ralston, A. *A First Course in Numerical Analysis*. McGraw-Hill, Inc.: New York, 1965.

[14] Srikanth, R., Ramakrishnan, A.G. Contextual Encoding in Uniform and Adaptive Mesh-Based Lossless Compression of MR Images. *IEEE Transactions on Medical Imaging*, Vol. 24, No. 9, pp. 1199–1206, 2005.

[15] Varga, M. *Využití tetraedrizace jako alternativy k objemovým datům*. Diploma thesis, Univerzita Karlova v Praze, 2007.

[16] Vomáčka, T. *Delaunay Triangulation of Moving Points in a Plane*. Diploma thesis, University of West Bohemia, Pilsen, 2008.

[17] Vomáčka, T. Delaunay Triangulation of Moving Points. In *Proceedings of the 12th Central European Seminar on Computer Graphics*, pages 67–74, 2008.

[18] Yaoping, Y., Chengke, W. A Novel Video Coding Scheme Using Delaunay Triangulation. *Journal of Visual Communication and Image Representation*, Vol. 9, No. 1, pp. 80–86, 1998.

[19] Žára, J., Beneš B., Felkel P. *Moderní počítačová grafika*. Computer Press, 1998, ISBN 80-7226-049-9.