Motion Estimation of Videos-Objects by Finite Elements Mesh-Size

M. Ettaouil and F. Belhabib

Abstract—In this paper, we present an approach for motion estimate adopted by standards actual videos. The proposed method consists modeling the considered domain in form of a structure of mesh-size using the finite elements method. Basing on the object of interest via the finite elements. Then, the obtained mesh is used in order follow-up of objects deformation. It is a question of determining the deformation of an object nets, from one image to another. For that we used the algorithm allowing following a deformable plane object. On the one hand, we improve its performance, and then we study the optimization of the error function by mesh simplification of our model, among the different triangular meshes associated with images references. At the end of this work, we present the simulation results.

Index Terms—Mesh-size, finite elements method, adaptive mesh-size, mesh deformable object, optimization of the error, object videos, interpolation.

I. INTRODUCTION

A video sequence is characterized by a large amount of data that should be conveyed through a transmission network or occupied on a storage medium. It is presented by a set of strongly correlated consecutive pictures. To date, the estimation and compensation of movement are techniques that are widely developed and used by video standards to reduce the temporal redundancies [1] [2].

Getting place on a domain that corresponds to the representation space of the physical problems, to simplify the presentation, we will constraint to the case where is a differential variety of 2D dimension. This case covers the tridimensional objects applications to the image and to the geometry. The first phase in the video traitement field, one construct au adequate meshes [3]. Generally, a mesh –size is defined as a partition in a finite number of elements. These latter can be chosen as triangles connect by nodes and edges. For this, we perform a discretization of the border in order to obtain of meshes surface [4].

A mesh size is therefore determined by geometry and topology (relations of connectivity between the nodes) [4]. It's said to be consistent if the intersection of two of these distinct triangles is either an edge or a node, or empty. In other words it prevents a summit to be specified only at the end of an edge. Property compliance is critical to ensure the continuity of function interpolating thereof. We distinguish different types of meshes. A priori triangles can be, but it

Manuscript received October 19, 2011; revised January 8, 2012.

often imposes constraints on the number of edges of these triangles. It often sets the number of edges by triangles (triangular meshes) or (quadrilateral meshes). Triangular meshes are however a number of advantages. On the one hand, triangles can more accurately model the border of any domain. Indeed, any area polygon border is triangularly. One the other hand triangular meshes interest to provide continuous representation modeled objects and the associated interpolation functions are often easier [5].

This paper is organized as follows: The first paragraph relates to the mesh-size finite elements method, that is to say, approximation by piecewise, the second consists introduced triangular mesh size by models of the surface; the third consists in modeling a model of the optimum adaptation of a mesh size by triangle surface of finite elements [6]. Simulation results are also available.

II. MESH-SIZE FINITE ELEMENTS METHOD

A. Finite elements Method

We present generally how can be approximated by piecewise that are to say mesh-size by finite elements [7][8]. The finite elements method uses a representation based nodes by estimating the values in the face by functions the nodal values weighting. Polynomial representation is also widely used in the field of image deformation or coefficients are obtained by identifying the nodal values to a polynomial approximation of piecewise. Mesh-size is used for modeling by discretization of partial differential equations. It is also possible, by a judicious choice of weighting functions of the finite elements method, these two performances are the same, and however, the nodes based representations have several advantages:

- Numerical analysis more efficient and more stable.
- The definition of positions and values of the nodes require less accuracy that polynomial coefficients estimation (this last point is especially important in applications of coding that all these parameters are subject to the quantification).
- It is easier to visualize a function or deformity interpolated from the positions of the nodes from polynomial coefficients.

The principle of the finite elements method is to show values calculated from approach variable taken in a number of end points properly chosen. We then rebuilt solution across the field by interpolation [9] [10]. Extends in the method discretization involves setting up a domain partition considers in simple elements. This partition is the mesh of the domain and must possess properties that depend on the application. The finite elements methods provide

Authors are with the Faculty of science and Technology of Fez, university Sidi Mohammed ben Abdelah; Box 2202, Fez, MORROCO (e-mail: mohamedettaouil@yahoo.fr; e-mail: F.BELHABIB@yahoo.fr).

approximations of exact solutions and we try to get the error associated with the smallest possible [11].

III. TRIANGULAR MESH SIZE BY MODELS OF SURFACE

Many problems are modeled by partial differential equations, whose solutions are approximated numerically by the finite elements method [12]. This method is to evaluate values approached the field looked only at certain points in the domain, and then deduct these values by interpolation, the solution at any point. A first step will be to achieve a mesh size of the field that will define, among other things, points (also called nodes). This pre-treatment is an essential phase of the method, the mesh nodes, will determine the convergence of the method to a good solution. Also, the domain can have a complex geometry, which implies that the mesh is not trivial. We focus only, in this section triangular mesh [13] [14].

A. Methodology

In this paper, the conception of the finite elements method of the mesh can be realized by three phases:

- The first phase is the analysis of the problems, which consist the studying of domain geometry that is very complicated problem will be broken down into simple forms problems.
- The second phase is the formal construction the mesh: Takes into account the analysis results, and defines simple objects allowing division the total work into phase.
- The third phase is the realization of the mesh, which includes the previous phases.

The solution of finite elements of type movements is interpolated on the structured cloud of points builds in the steps. To do it, we proceed as follows:

- The network of points calculated in the previous step is planned on the mesh of the face not deformable.
- These coordinate are used as functions to calculate the point surface of interpolation to deform; by using the functions of interpolation; calculate the solution corresponding to the type movements by the finite elements method and the learning algorithm [15].

B. Triangular Mesh by Finite Elements

The image I of the surface that we seek to approximate by a surface defined a polynomial in two variables: $\hat{I}(x, y) = P^n(x, y)$ we know that it is preferable to perform interpolation piecewise rather than global phenomena that lead to instability, but this involves studying between the different areas in order to ensure the consistency of the interpolate [16].

- We restrict ourselves by triangular elementary domains, also called elements. To triangulate and cover domain Ω image elementary domains support, triangle offers flexibility.
- This is particularly useful for approaching the boundaries whatsoever such as the edges of object. Image border Ω is a polygon and in this case Ω is fully and accurately covered by triangles Basic fields defining a partition of the domaine Ω [17]:

- An intersection is empty.
- Either an intersection reduced to a common vertex.
- Either an intersection reduced to one side common.
- Should therefore define a global numbering of elementary domains e_i, global numbering for all S_i summits and a relationship that each elementary domain associates list heights.

Given a triangulation $T_{\Omega} = \bigcup_{i=1}^{N} e_i$ formed by triangles of a closed polygon domain Ω . Knowing the value S_i to the surface to approximate in each vertex of a triangle, I_{S_i} , it seeks to build \hat{I} such as:

Restricting Court an e triangle is a polynomial of degree $P_T \leq 1$.

• Determining P_T on e triangle vertices S_1 , S_2 and S_3 knowing I (S_1), I (S_2), and I (S_3), conditions are imposed:

$$P_{\rm T}(S_{\rm i}) = I(S_{\rm i}) \ i\epsilon \{1, 2, 3\} \tag{1}$$

To do this, define 3 basic functions are commanded local ψ_i checking for an e triangle vertices S₁, S₂ and S₃:

$$\psi_i(S_j) = \delta_{ij} \ i, j = 1, 2, 3$$
 (2)

• Interpolated point value expression (x, y) owned by an element (Fig.1), therefore corresponds to:

$$\hat{I}(x, y) = \sum_{l=i,j,k} \psi_i^e(x, y) . I(S_l)$$
(3)

if $(x, y) \in e_{i,j,k}$



Fig. 1. Interpolation of value depending on the model of Lagrange, equivalent to the projection of the point (x, y) on the plan.



Fig. 2. Domains elementary trianglars

C. Algorithm Development

Our algorithm is based on the surface deformation before and after by finite elements method (movements of nodes)





Results: (Fig 4)



Fig. 4. Mesh deformable object (surface)

The steps of the algorithm are as follows:

- Construct a domain of the face undeformed.
- Projecting this point out the undeformed mesh support.
- Search for each point what triangle its projection belongs and calculate the local coordinates of these points in the triangle found.
- Determine the coordinates of the point in the media distorted based on initial coordinates before deformation, the local coordinates of the point within the triangle and movements of nodes (the finite elements solution).

Step 3:

The solution of finite elements type displacement is interpolated on the structured scatter built in step 2. To do this, is as follows:

- The network of points computed in the previous step is projected on to the surface mesh undeformed. Thus each point is propelled into a triangle.
- Local coordinates of the projected for each of the points in the triangle corresponding (Figure 3). These details are used as functions to calculate surface interpolation points distort using calculated interpolation functions solution finite type movements with algorithm of the learning. Given a point projected on a triangle that has three nodes N₁, N₂, N₃.

The algorithm (JD) [18] of follow-up of the learning decomposes into two successive stages:

- A first stage said about learning, which rests only on the reference picture I₀. She consists simply in replacing advantageously (stronger and more flexible):

$$\widehat{A} = [(I_0^i)(\beta)]^+$$

$$= [(\nabla I_0(m_j))^T J_{Tmj}(\beta)]_{j=1..n}^+$$
(4)

 (I_0^i) : The derivative of I_t evaluated by the β parameter $J_{Tmi}(\beta)$: Jacobi matrix

The second stage, which can be made on-line, thus real time, said about follow-up, consists in applying the relation of Matthews and Baker by calculating $(T^{\Delta\beta^i})^{-1}$ by means of result of the learning [19].

This algorithm is applicable for various models of movement, in particular for homographiques transformations.

On these bases, "Jurie and Dhome "[18] choose to consider A from the solution of the problem at the slightest following linear squares (Fig. 4):

$$\widehat{\mathbf{A}} = \begin{cases} \operatorname{argmin} \ \sum_{l=1}^{N_a} \left\| \mathbf{A} \Delta \mathbf{I}^l - \Delta \beta^l \right\|_2^2 \\ \mathbf{A} \in \mathbb{R}^{p \times n} \end{cases}$$
(5)

 N_a : The number of experiments (perturbations) brought into play for learning,

 $\Delta\beta^{l} \in \mathbb{R}^{p}$: Is the disturbance vector of parameters.

 ΔI^1 : Result of the disturbance

$$\Delta I^{l} = I_{0}^{i}(\beta_{0} + \Delta \beta^{l}) - I_{0}^{i}(\beta_{0}) \in \mathbb{R}^{n}$$
⁽⁶⁾

Noting:

It gets the flowing scheme (Fig 5):



Fig. 5. Nodes of the deformable meshes

Motion estimation (ME) based on a triangular mesh, partitions the image into a number of triangular patches where the vertices are denoted as grid points. The displaced grid points define a deformed mesh that describes the underlying motion. The deformed mesh of the reference frame is obtained from estimating the displaced position of the mesh vertices in the current frame.

The mesh structure that is used in this algorithm is based around an learning algorithm, adding each point, point by point, and structuring the mesh, until all points are added. Once the mesh is constructed using the triangulation algorithm, it is the initialized as a mesh. This was initially attempted using an edge following algorithm, but proved difficult due to problems with incomplete edges and image.

- In this implementation the mesh of the finite element method lines no image feature relationship, i.e. the mesh lines do not correspond to feature edges or lines within the image. However, in the mesh lines follow closely the edges of the object the image. In this mesh structure:
- The spatial relationship of multiple objects is apparent and can be described by the relationship and connections of the mesh lines.
- The mesh can provide reasonable region estimation since the points used for creating the mesh is one the boundaries of objects. If a more defined mesh is required then the number of feature points can be increased. Our objective is optimized the mesh by our approach.
- A simple motion vector interpolation method (Fig. 8 and Fig.9).

Even though such a technique could benefit significantly from the smooth motion field that is generated using zonal based algorithms, the process could also work with any other motion estimation algorithm including FS. This is very useful and practical, especially when considering that most current video sequences have been pre-encoded using different motion estimation algorithms. Zonal algorithms though have the benefit that they require very little computation and give better performance than almost all other motion estimation algorithms. It might also be desirable in a practical temporal interpolation system to perform additional motion estimation using smaller blocks, or even transcode the entire sequence, thus refining the motion field, and consequently the motion compensation technique discussed above.

Finally, as we have previously, interpolation could in some cases fail, causing additional artifacts instead of improving quality. To avoid such a problem we may classify a frame as predictable or non-predictable by examining and evaluating both motion vectors and the additional error necessary for generating a predicted frame. In particular, if a frame relies mainly on the error difference of the mesh (larger error) or even has too many intracoded, then we may claim that such a frame is non-predictable. A thresholding value could therefore be selected, according to which, if the total error difference in a frame is larger than this threshold then interpolation for this frame would be skipped. Motion vector correlation could also be used to further help in this process.

IV. MESH OPTIMIZATION FOR SURFACE APPOXIMATION

A. Mesh Representation

Intuitively, a mesh is a piecewise linear surface, consisting of triangular faces pasted together along their edges. For our purposes it is important to maintain the distinction between the connectivity of the vertices and their geometric positions. Formally, a mesh M is a pair (K,V), where: K is a simplified complex representing the connectivity of the vertices, edges, and faces, thus determining the topological type of the mesh; $V = \{v_1, _, v_m\}, v_i \in R^3$ is a set of vertex positions defining the shape of the mesh in R^3 (its geometric realization).

The mesh optimization problem considered in this paper can be roughly stated as follows: Given a collection of data points X in R^2 and an initial triangular mesh M0 near the data, find a mesh M of the same topological type as M0 that fits the data well and has a small number of vertices.

The problem of surface reconstruction from sampled data occurs in many scientific and engineering applications. In [20], we outlined a two phase procedure for reconstructing a surface from a set of unorganized data points. The goal of phase one is to determine the topological type of the unknown surface and to obtain a crude estimate of its geometry. An algorithm for phase one was described in [20]. The goal of phase two is to improve the fit and reduce the number of faces. Mesh optimization can be used for this purpose.

Although we were originally led to consider the mesh optimization problem by our research on surface reconstruction, the algorithm we have developed can also be applied to the problem of mesh simplification. Mesh simplification, as considered by Turk [21] and Schroeder et al. Refers to the problem of reducing the number of faces in a dense mesh while minimally perturbing the shape.

The principal contributions of this paper are:

• It presents an algorithm for fitting a mesh of arbitrary topological type to a set of data points (as opposed to volume data, etc.). During the fitting process, the number and connectivity of the vertices, as well as their positions, are allowed to vary.

• It casts mesh simplification as an optimization problem with distance between triangular meshes that directly measures deviation of the mesh from the original. As a consequence, the final mesh naturally adapts to curvature variations in the original mesh by full search.

The simplify meshes generated by "marching cubes" that may consist of more than a million triangles. In their iterative approach, the basic operation is removal of a vertex and re-triangulation of the hole thus created. The criterion for vertex removal in the simplest case (interior vertex not on edge or corner) is the distance from the vertex to the plane approximating its surrounding vertices. It is worthwhile noting that this criterion only considers deviation of the new mesh from the mesh created in the previous iteration; deviation from the original mesh does not figure in the strate. Turk's goal is to reduce the amount of detail in a mesh while remaining faithful to the original topology and geometry. His basic idea is to distribute points on the existing mesh that are to become the new vertices. He then creates a triangulation containing both old and new vertices, and finally removes the old vertices. The distance of the new vertices is chosen to be higher in areas of high curvature.

B. Distance between Triangular Meshes

For the sake of simplicity, we will only present the case of discrete 2D models represented by triangular meshes, since this is the most generic representation of such data. A triangular mesh M will be represented by a set P of points in R^2 (vertices), and by a set T of triangles describing how the vertices from P are linked together. We will denote by S and S0 two continuous surfaces.

Let us define the distance d (p, S') between a point p belonging to a surface S and a surface S0 as:

$$d(p, S') = \min_{p' \in S'} ||p - p'||_2$$
(9)

where $\|.\|_2$ denotes the usual Euclidean norm. From this definition, the Hausdorff distance between S and S', denoted by d(S, S') is given by:

$$d(S,S') = \max_{p \in S} d(p,S') \tag{10}$$

It is important to note that this distance is in general not symmetrical, i.e. $d(S, S') \neq d(S', S)$. We will refer to d(S,S') as forward distance, and to d(S', S) as backward distance. It is then convenient to introduce the symmetrical Hausdorff distance $d_x(S, S')$, defined as follows:

$$d_{x}(S,S') = \max[d(S,S'), d(S',S)]$$
(11)

This is a simple motion estimation that compares the source node of the mesh with the nodes at every position within the search area. As mentioned above, the calculation costs of this algorithm are very high, but it guarantees to find the optimal mesh position within the search range (Fig.10).



Fig. 6. Full search algorithm

The number of comparisons increases quadratic ally (n^2) with the search range :(Fig. 6, Fig. 7 and Table I).



V. EXPERIMENTS RESULTS

Experimental results are provided to compare the performance of the proposed motion estimate of the video objects by occlusion adaptive forward mesh tracking versus test sequence. The sequence starts with slow movements of the head of the Sequences Miss America, Foreman, Salesman, and Table Tennis were selected for the simulation and results can be seen in Table 2. However, the part between frames 56 and 87 is especially challenging and well suited to demonstrate the occlusion-adaptive mesh tracking concept. The processed video, the overall performance is improved in all test sequences at a fixed bitrate (0.2 bpp). We identified two results the motion estimate.

1) a new uniform mesh at each frame before interpolation, and 2) a new uniform mesh content-based optimization the surface at each frame. An algorithmic description of the motion-estimation by surface mesh approach is given in the Appendix. The case of redesigning a uniform mesh is expected to be a lower bound on the performance of motion estimation by the proposed forward tracking content-base mesh, since the structure of the mesh may not fit the motion boundaries well, leading to multiple motions within a single patch. However, it requires transmission of no overhead information about the mesh structure.

To this effect, the efficacy of the proposed method has been evaluated based on how it compares against these benchmarks in terms of motion-compensation PSNR and the number of node points whose coordinates need to be transmitted at each frame. The PSNR values refer to the prediction PSNR of each frame based on the original of the previous frame, using the affine motion field interpolated from the node-point motion vectors.

We have implemented the algorithms described in this paper in Java and employed them in a cloth simulation system. The tests were run on a standard Pentium 4 with 2.8GHz and 512 MB RAM, using the Java Runtime Environment 1.6.0.



a) Threshold of 0



c) Threshold after interpolation

Fig. 8. Points of intersed beford and after interplation by setting threshold $(a,b\ et\ c)$

Fig. 8: shows the results obtained by the new modeling. Our algorithm is based on the surface for deformation objects before and after interpolation by setting threshold.



Original image Foreman, frame 100



(a):Triangular mesh: Image Foreman, frame 100



(b): Image Foreman, frame 168 by traingular



(c): Image Foreman, Frame 168 : by traingular and deformation



Absolute Error: 117.4839 (d): Image Foreman, Frame 168 : reconstructed

Fig. 9. Triangular mesh before and after deformation (a, b, c and d)

Fig 9: shows the results obtained by the approach of the constructed mesh before and after deformation objects (a, b, c) and the reconstructed of the images by mesh (image d).

TABLE I: gives the Full Search (FS) for every sequences the dataset used in this study.

TABLE I: MOTION VECTOR ENTROPY VERSUS DIFFERENT PREDICTORS

	Full Search				
Sequences	(0,0)	Median	Set node		
Foreman	6.78	5.17	4.64		
MISS	6.86	5.31	4.37		



a) Motion estimation Miss: mesh before interpolation



b) Motion estimation Foreman: mesh optimization



(c):Image Foreman, Frame 168 by mesh optimization.



Absolute Error: 54.7963 (d):Image Foreman, frame 168 : error 54.7963



(e) :Motion estimation Miss: mesh optimization

Fig. 10. Mesh optimization (images: a,b,c,d) and reconstructed (image e)

A simple comparison between both the reconstructed images, via our approach, and the original, images shows that the proposed method the mesh optimization it is enable to construct images of good visual quality. Fig. 10.

We have tested the both proposed motion estimation approaches and the original motion estimation technique on sequences of the images. See the Figure 11.Performances of the above algorithms are evaluated in terms of bit rate (bits per pixel) and Peak Signal-to-Noise Ratio (PSNR) is given by:

$$PSNR = 10\log_{10}(\frac{255^2}{MSE})(endb)$$
(12)

where Mean Squared Error (MSE) is defined as follows:

$$MSE = \frac{1}{T} \sum_{i=1}^{T} (\hat{x}_i - x_i)^2$$
(13)

where \hat{x}_i and x_i denote, respectively, the original and the encoded pixel values and T is total number of pixel in triangular image.

To motion estimation of objects on sequences images, we dived it into triangular image. Numerical results obtained by applying the proposed modeling, presented in the Table II and Table III.

These tables list, respectively, the sequences of the test images, the number of iterations, Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) associated of different frames (50, 56 and 168).

	Nmax=300			
Sequence	N.It	Frame	N.nd	PSNR
Foreman	50	50	1100	28,01
MISS	100	56	1000	30,01
Foreman	150	168	2500	22,23
MISS	300	168	1050	28,01

Table III gives the error values for every dataset used in this study. It must be stressed that the algorithm was run with all its parameter values fixed to be the same for all datasets.

TABLE III: EXPERIMENTAL RESULTS: ABSOLUTE ERROR AND NODES

	Nmax=600			
Sequence	N.It	Frame	N.nd	MSE
Foreman	50	168	3000	117.4839
Foreman	100	168	2950	94.0877
Miss	150	168	2500	64.1786
MISS	300	168	1000	37.1628

If we assume the outputs with Error less than 94.0877 as the good motion estimation results, from this table we can see that datasets fall into this category. We can observe that all these good out puts of the Number of node (N. nd), indicating that these results overestimate the triangular image. An example of a good motion estimation result is shown in Fig. 11.



Absolute Error: 37,1628



Absolute Error: 64.1786

(b)

Fig. 11. Reconstructed images by the finite elements mesh-size: (a): Error =37.1628 and (b): Error=64.1786

VI. CONCLUSION

We introduced in this paper a method of the motion estimate of the video objects by finite elements the triangular mesh interpolated and optimization of the mesh. In this paper, we have presented a new modeling for the motion estimate of the video objects. The results obtained by the present method are promising. This process is based on the optimization of the error mesh between two surfaces. Our perspective is to use the refinement of mesh sizes with artificial neuronal networks for the compensation of videos-objects movement. The research issues such as efficiency of the compression rate the video will be studied in the future.

REFERENCES

- S. Baker and I. Matthews. "Equivalence and efficiency of image alignment algorithms." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1090–1097, 2001.
- [2] J. Garrigues, "La méthode des éléments finis," Ecole Supérieur de Mécanique de Marseille, Septembre 1999.
- [3] A. Björck. "Numerical Methods for Least Squares Problems." *SIAM, Philadephia (PA)*, 1996.
- [4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. "Active appearance models." *IEEE Transactions on Pattern Analysis Machine Intelligence*, 23(6).pp-681–685, 2001.
- [5] A. Elamin, Varun Jeoti and Samir Belhouari, "Region-Based Adaptive Distributed Video Coding," *International Journal of Computer Theory* and Engineering vol. 3, no. 2, pp.-304-310, 2011.
- [6] H. Folch, "Echantillonnage irrégulier et interpolation : application à la représentation d'images fixes et animées."- PhD. Thesis, Université de Rennes-1, Décembre 1997.
- [7] M. Ettaouil and Y. Ghanou, "Neural architectures optimization and Genetic algorithms," *Wseas Transactions On Computer*, Issue 3, Volume 8, pp- 526-537, 2009.
- [8] M. Ettaouil, Y. Ghanou, K. Elmoutaouakil, and M. Lazaar, "Medical image compression by a new architecture optimization model for the kohonen networks," *International Journal of Computer Theory and Engineering*, 2010.
- H. Le Floch. Echantillonnage irrégulier et interpolation : application à la représentation d'images fixes et animées. PhD thesis, SPM/IRISA, 1999
- [10] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework International Journal of Computer Vision,56(3) :pp-221–255, 2004.
- [11] P. George. Génération Automatique de maillage- Applications aux méthodes d'éléments Finis. Masson 1991.
- [12] K.HO-Le, "Finite element mesh generation methods:" Areview and classification I comput. Aidet Deseign 20(1): pp- 27-38, January 1988.
- [13] ISO/IEC 14496-10 and ITU-T Rec. H.264." Advanced Video Coding". 2003.
- [14] H. B. Jung and K. Kim, "A new parameterization method for nurbs surface interpolation," The International journal of Advanced Manufacturing Technology, Vol. 16, 2000, pp- 784-790.
- [15] S. Padmakala and Dr.G.S.AnandhaMala, "A novel video object segmentation approach for noisy video sequences towards effective video retrieval," *International Journal of Computer Theory and Engineering* vol. 2, no. 6, pp. 826-834, 2010.
- [16] J. A. Robinson and M. A. Ren. Data-dependant sampling of twodimentional signals. Multidimensional Systems and Signal Processing, 6: pp- 89–111, 1995.
- [17] I. E. G. Richardson. "H.264 and MPEG-4 video compression: video coding for next- generation multimedia. Ed. John Wiley & Sons," 2003.
- [18] F. Jurie and M. Dhome. "Hyperplane approximation for template matching." In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24(7), pp- 996- 1000, 2002.
- [19] A. Perera, C-L. Tsai, R. Flatland, and C. Stewart. Maintaining valid topology with active contours: theory and application. IEEE International Conference on Computer Vision.
- [20] H. C. Vijay Lakshmi and S. PatilKulakarni, "Segmentation algorithm for multiple face detection in color images with skin tone regions using color spaces and edge detection techniques," *International Journal of Computer Theory and Engineering* vol. 2, no. 4, pp. 552-558, 2010.
- [21] G. Turk., "Re-tiling polygonal surfaces," Computer, Graphics (SIGGRAPH '92 Proceedings), 26(2):pp-55–64, July 1992.