Image Coding based on Flexible Contour Model

Nanning Zheng*,Wenwei Song* and Wei Li[+],
*Xi'an Jiaotong University, 710049, Xi'an, P. R. China
nnzheng@xjtu.edu.cn, wwsong@hotmail.com
[+]Computer Science Department, State University of New York at Stony Brook,
Stony Brook, NY 11794-4400, USA
liwei@cs.sunysb.edu

Abstract: This paper presents a new scheme of model-based image coding method. First a new image model called Flexible Contour Model that can extract features of nonrigid objects in images is proposed, then we deduce the fast algorithms for calculating the parameters of the model and for matching the model to images. Furthermore the combination of the model with multiscale analysis and the triangulation of the model has been studied. As a result, reconstruction of original images with high compression rate and unnoticeable distortion was obtained.

1. Introduction

Recently model based image compression has attracted interests of many researchers, whose theoretical compression rate may be estimated as 10^4-10^5:[1]. The essence of the technique is to find an appropriate model that can represent main contents of images through its parameters that are actually the coding results, then by taking advantage of techniques of computer graphics, reconstruction from those parameters can be achieved. Hence the main distortions contained in the reconstruction are geometrical distortion to which the human vision system is much less sensitive than to the distortions such as staircase edge or block effect in reconstruction through traditional methods. Because face-to-face communication is one of the most important applications of image coding and various facial images generally posse quite similar features, model based facial image coding is

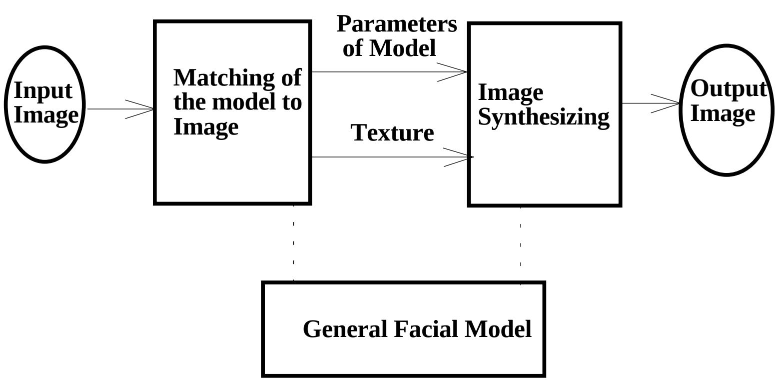


Fig. 1

Principe of Model based Facial Image Compression

currently the most popular trend.

Model based facial image coding is composed of the following procedures:[2] 1) Designing a general model of human faces; 2) Matching the general model to a specific image; 3) Storing or transmitting parameters of the model and some texture information if necessary; 4) Reconstructing the image according to model parameters and Texture. The general model can be accessed both in encoder and decoder. Consequently only model parameters and a small part of texture need to be transmitted or stored. The process is depicted by Fig. 1.

Section 2 will describe the structure of flexible contour model proposed by the authors and the algorithms of parameter calculation and optimization of the model. Furthermore the combination of multiscale analysis with the model was studied. In section 3, Delaunay triangulation applied to flexible contour model is briefly discussed. Section 4 is about the reconstruction from model parameters. Section 5 concludes the whole paper.

2. Flexible Contour Model

2.1 The Structure of Flexible Contour Model

First we would like to remind readers of the scene of little kids drawing human faces. They first draw a big circle represent the contour of a face, and 2 smaller circles as 2 eyes, finally other 2 circles as a mouth and a nose. Getting a glimpse of such a drawing, we seldom fail to relate it to human faces, for those simple scratches covering the basic features of faces. This is just the stimulation of the authors of the idea of flexible contour model.

Without losing generality, we suppose that natural images can be segmented into several multi-connected regions. Pixels belong to the same region have same or similar properties. The property referred here may be simply intensity or it may be some more complicated features such as local texture and local fractal dimension. For the simplicity of the algorithm, in this paper we only use pixel intensity as region feature.

A flexible contour model[3] is composed of several of closed contour lines whose shapes are retained by sets of control points. So we may regard the contours in the model as polygon, that is why we also refer control points as vertex. One contour can be completely enclosed in another, but it is prohibited for any two contours to intersect not at vertex. In other words, if two contour lines have any common point, then the point should be at least one of the contours' control point. By shifting control points in 2-D domain to adjust the shape and the position of each contour, it is possible to match the model to an image in which shapes of objects are outlined by contours.

For each image, we define a border contour containing 4 vertices that are fixed at the 4 corners of the image to prevent other contours to extent out of the image range.

In this paper, we use C to denote a contour, specifically C_B is the border contour. If contour *C'* is completely enclosed in *C* and there is no other contour in *C* and containing *C', we say C'* is a child contour of *C*. We use C_c^i to represent a child contour of C, where subscript c means child and superscript i=1.N, N is the total number of child contour of C; p indicates a pixel, $\overline{C}$ is a set of pixels and $\overline{C}$={p|p is in C}; we say $p\Theta C$, if and only if $p \in \overline{C}$ but p$\notin \bigcup_i^N \overline{C_c^i}$, which means that p is in the multi-connected region bounded by C and C_c^i, i=1 to N.

Besides the coordinates of control points, a contour *C* is associated with two important parameters: ***Area:*** is equal to the number of pixels *p*, where $p\Theta C$, and ***Mean:*** the mean values of pixels *p*.

We take MSE (mean square error) between original image and reconstruction through contour model as the reconstruction distortion measure:

$$E^2 = \sum_{\forall C} \sum_{p \in \overline{C}} (I_p - \hat{I}_p)^2 \qquad \textbf{(2. 1)}$$

It is reasonable to consider that a contour model is optimized if an image E^2 is minimized. Therefore the optimization of flexible contour model is converted the minimization of MSE.

2.2 Calculation of Model Parameters

Apparently, it is difficult to find the minimum of E^2 directly, so we can only exploit an iteration method to let E^2 gradually converge to one of its local minimal. Then in each step of the iteration, the calculation of model parameters is necessary. In order to reduce computation, we take advantage of the unanimity of pixel values within one continuous region and derived a method for fast calculating ΔE^2 . As shown in Fig. 2, let *V* denote a vertex of contour *C*, *Vl*、*Vr* are the neighbor vertices of *V* on the same contour. We use *C'* to represent the contour by deleting vertex *V* and connect *Vl,Vr*. Clearly,

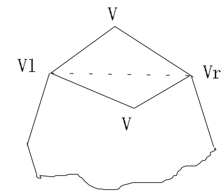


Fig. 2

there are two cases according to position: inside and outside *C'*.

For the convenience of explanation, we first define the following symbols: *Ap*、*Mp*、*Ap'*、*Mp'*、*At*、*Mt*、*Ac*、*Mc*、 *Ac'*、 *Mc'* are the *Area* and *Mean* value of the parent contour of *C*、the parent contour of *C'*、the triangular contour *VVlVr*、contour *C* and *C'* respectively. If **V** is outside of *C'*，then the change of E^2 by substituting *C'* for *C* is:

$$\Delta E^2 = \sum_{p\Theta Cp'} (Mp'-Ip)^2 + \sum_{p\Theta C'} (Mc'-Ip)^2 - \sum_{p\Theta Cp} (Mp-Ip)^2 - \sum_{p\Theta C} (Mc-Ip)^2 \quad \textbf{(2. 2)}$$

By simplifying formula (2. 2), we get:

$$\Delta E^2 = Ap(Mb'-Mb)^2 + Ac(Mc'-Mc)^2 + At(Mp'^2 - Mc'^2 - 2Mp'Mt + 2Mc'Mt) \quad \textbf{(2. 3)}$$

In formula (2. 3), *Ac*、*Mc*、*Ap*、*Mp* can be deducted from *Ac'*、*Mc'*、*Ap'*、*Mp'*, and vice versa. If *Mc*、*Ap*、*Mp* are known, we have:

$$\begin{cases} Ac' = Ac - At \\ Ap' = Ap + At \\ Mc' = \dfrac{AcMc - AtMt}{Ac - At} \\ Mp' = \dfrac{ApMp - AcMc + AtMt}{Ab + At} \end{cases} \quad \textbf{(2. 4)}$$

If *Ac'*、*Mc'*、*Ap'*、*Mp'* are known, then:

$$\begin{cases} Ac = Ac' + At \\ Ap = Ap' - At \\ Mc = \dfrac{Ac'\,Mc + AtMt}{Ac' + At} \\ Mp = \dfrac{Ap'\,Mp' + Ac'\,Mc' + AtMt}{Ap' - At} \end{cases} \quad \textbf{(2. 5)}$$

According to formula (2. 3), for getting the difference of E^2 we only need to count the number of pixels and calculate their mean value within triangle *VVlVr*, instead of computing the sum of square pixelwise for the whole image, therefore much reduction of computation can be acquired.

For the case of *V* inside *C'*, so long as we put a minus before the area value of triangle *VVlVr*, the above formulas are still applicable. We define symbol $\hat{At}$ and replace *At* with $\hat{At}$ in formula (2. 3)5):

$$\hat{At} = \begin{cases} At, \text{if NOT V}\Theta\text{C'} \\ -At, if V\Theta C' \end{cases} \quad \textbf{(2. 6)}$$

There exists a special situation, that is, contour *C* is a triangle, then *Ac'*=0，*At*=*Ac*，and:

$$\Delta E^2 = Ap(Mb' - Mb)^2 + Ac(Mp' - Mc)^2 \quad \textbf{(2. 7)}$$

With the above results, it is very easy to calculate the change of E^2 if a control point is moved while all the others kept stable. If vertex *V* is shifted from position *V1* to *V2*，then:

$$\Lambda E^2 = \Lambda E_{V2}^2 - \Lambda E_{V1}^2 \quad \textbf{(2. 8)}$$

Where ΛE_{V1}^2 and ΛE_{V2}^2 can be computed by formula (2. 3).

2.3 Optimization of the Model

Optimization of contour model is performed by shifting control points to decrease E^2. If after each step of shifting, the change of E^2, ΔE^2 is constantly minus or zero, then the model will gradually converge to a local optimum.

For integrating the optimization of the model and various constraints during the process, we introduce the concept of "**Force**". Each control point is affected by several forces and it is just according to the direction and strength of the sum of those force vectors that the position of points will be altered. If vertex *V* is affected by force $\vec{F}$ at step K of the iteration, then at step K+1, there is a trend for the vertex to shift $\|\vec{F}\|$ pixels in the direction of $\vec{F}/\|\vec{F}\|$, where $\|\vec{F}\|$ and $\vec{F}/\|\vec{F}\|$ are the strength and direction of the force separately. The forces include:

Gradient force: $\vec{F}g$. This is the most important force in our model, which attracts control points to the places where E^2 is decreased. There are several ways defining gradient force, among which the simplest method is the searching method. That is, for every control point, to search for a new position where if the point shift, a local minimum of E^2 will be gotten. Then the vector between the original location of the point and its searched destination may be exploited to indicate the gradient force for the point. In stead of using this time consuming method, we have developed an estimation algorithm for finding the local optimal position of vertices of contours.

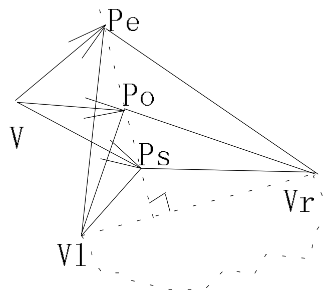


Fig. 3

As shown in Fig. 3, let *V* denote a vertex of contour *C*, *Vl* and *Vr* are it's two neighbors on the contour. The current position of *V* is *Ps*. We restrict *V*'s shifting along the line *l* that is perpendicular to segment *VlVr* and across *Ps*. We use *Co*、*Ce*、*Ce-o* to denote the contours confined by *VlPoVrPs*、*VlPeVrPs* and *VlPeVrPo* respectively.

Suppose *Po* is the optimal position of *V* on *l*. Now we can describe the evaluation algorithm. First we move *V* along l to a new position *Pe*. Naturally, there are two choices of *V*'s moving direction, let us consider the case of $Pe \notin C$ first. We suppose $\|\overline{PePs}\| > \|\overline{PoPs}\|$, *Pe* and *Po* are on the same side of *Ps*. For the purpose of simplicity, we hypothesize：

$$Ip = \begin{cases} Mc, p\Theta Co \\ Mp, p\Theta Ce-o \end{cases} \tag{2. 9}$$

Then the change of E^2 caused by moving *V* from *Ps* to *Pe* can be computed by:

$$\Delta E^2 \approx \sum_{p\Theta Ce}((Ip-Mc)^2-(Ip-Mp)^2 = -AreaCe(Mp^2-Mc^2)+2(Mp-Mc)\sum_{p\Theta Ce} Ip \tag{2. 10}$$

Let $a=\|\overline{VlVr}\|$, $b=\|\overline{PsPe}\|$, $\|\overline{PsPo}\| \approx t\|\overline{PsPe}\|$, then:

$$\Delta E^2 = ab(\frac{1}{2}-t)Mp^2+(\frac{1}{2}-t)abMc^2+abMpMc(2t-1) \tag{2. 11}$$

From (2. 11), the solution of *t* can be found:

$$t=\frac{1}{2}(1-\frac{\Delta E^2}{1/2ab(Mb-Mc)^2}) \tag{2. 12}$$

If *Po* and *Pe* do not lie on the same side of *Ps* or $\|\overline{PePs}\| \le \|\overline{PoPs}\|$, then the solution of t will be greater than 1 or less than 0. For such cases, we assign t the values 1 or 0. That is:

$$t=\begin{cases} \frac{1}{2}(1-\frac{\Delta E^2}{1/2ab(Mb-Mc)^2}),\ 0 \le \mathrm{t} \le 1 \\ 1,\ \ \mathrm{t}>1 \\ 0,\ \ \mathrm{t}<0 \end{cases} \tag{2. 13}$$

It is not difficult to derive that for the situation of $Pe \in C$ the result is just the same. Because *Pe* may situate on either side of *Ps*, two *Po* can be located, we can simply choose the one that make E^2 decrease more. If the two ΔE^2 corresponding to both *Po* are positive, just let t=0. Through experiment, we find that it is appropriate to take *b*=0.5*a*.

In order to make the gradient force affecting on a vertex *V* be continuous in strength and direction in the neighborhood of *V*， smoothing of images before matching the model is necessary. In our experiment, we applied 5×5 median filtering.

Rigidity Force: $\vec{F}r$. Its introduction is for keeping contour shape to be smooth and preventing points, which is appreciated to equally spaced along contour curve, from too much condensed to some parts of contour.

Push Force: $\vec{F}p$. When a contour shape is deformed, it may coincide with other contours. We must prevent such violation of the model's property and further constrain the deformation of contours. Let $\vec{F}$ denote the resultant of forces affecting on vertex *V* on contour *C*, the current position of *V* is $\vec{P}$. Suppose $\vec{P}i$ is a point in the direction of $\vec{F}$ and satisfies the following condition: if *V* is shifted to $\vec{P}i$, any two contours don't intersect except at vertices; but for any ε>0, if *V* is moved by the distance of $\varepsilon + \|\vec{P} - \vec{P}i\|$ in the same direction, then invalid intersection will occur.

We stipulate the distance for vertex *V* should be $\min(\|\vec{P} - \vec{P}i\|, \|\vec{F}\|)$. If $\|\vec{P} - \vec{P}i\| < \|\vec{F}\|$, then the distance for V to move is smaller than the resultant of forces affecting on it. This reminded us of the situation of a moving objects blocked by obstacles, so we design that the contours hindering contour *C* to move will be affected by a push force whose direction is the same as $\vec{F}$ and strength is equal to $\vec{F}p = \gamma(\|\vec{F}\| - \|\vec{P} - \vec{P}i\|)$, where 0<γ<=1 is tunable coefficient. Consequently, at the next step of iteration, it is possible for those hindering contours to yield places for hindered contours.

By the strategy descried above, we not only preserve the unoverlapping property of flexible contours but also keep information of movements of vertices.

Knowledge Force: $\vec{F}k$. It is based on prior knowledge that this force directs and constrains the shifting of contour vertices.

2.4 The Combination of Flexible Contour Model with Multi-Scale Analysis

From the study of psychophysics, it is approved that the observation of human vision system is a coarse-to-fine process. We would like to endue flexible contour model the similar property while matching to images. At the beginning, the model will try to match itself to an image with low resolution that contains relatively less detail thus is easier for the model to be optimized. Then by increasing resolution, more details will be added to the image. If at the lowest resolution, the model is optimized, then it is possible for the model to be further tuned by searching in a quite small region to represent the details at higher resolutions. The advantage of this idea is that control points can converge to their optimum faster and false deform of contours may be avoided.

Let the image used for analysis is the discrete approximation of the original natural image at resolution 2^0, $\{\vec{V}_k^i\}$ is the set of coordinates of contour control points at resolution 2^i, the subscript *k* is the index of vertices, then the shapes of contours are entirely determined by $\{\vec{V}_k^i\}$ and the corresponding connections. Suppose the coordinate vector of vertex *V* at resolution 2^i is $\{x, y\}$, then at resolution 2^{i+1} it will be $\{x/2, y/2\}$. We write:

$$\vec{V}_k^i = floor(2^{-1}\vec{V}_k^{i+1}), \forall n \tag{2. 14}$$

Where *floor(x)* returns the maximal integer not greater than *x*. When computing the gradient force for vertex *V* with the searching method, 8 pixels need searching if the searching radius is 1, if the radius is 2, 15 pixels, mathematically, if the searching radius is n, we need calculate ΔE^2 for $(2n+1)^2 - 1$ pixels.

Assume at resolution 2^{-n} a model matched to an image optimally, that is, errors of coordinates of vertices are less than 0.5 pixel for all the vertices. Generally, at resolution 2^{-n+1}, the errors will be less than 1 pixel if the model is not deformed. Hence we only need to use 1 pixel as the searching radius for adjusting the model at the higher resolution. When the resolution reaches 2^0 at last, the total number of pixels searched is $n \times (3^2 - 1) = 8n$. Whereas if the searching only applied at resolution 2^0,

$(2n+1)^2-1$ pixels have to be tried. The ratio of the two is: $\frac{(2n+1)^2-1}{8n}=\frac{n+1}{2}$ . . .

Obviously, if n is big enough, the difference of computation of the two methods is great, so the introduction of the theory of multi-resolution analysis to the optimization of the model can definitely save time. Certainly, the main aim is to ensure the correctness of matching.

3. Reconstruction

3.1 Delaunay Triangulation of Flexible Contour Model

For image synthesizing from flexible contour model, texture mapping will be applied. Both theory and experiment have showed triangular region is the most convenient for the purpose. Thus it is necessary to do the triangulation of the model, or in other words to convert flexible contour model to wire-frame model with triangle faces. Simply, the triangulation of a set of points is just to connect point pairs with segments as more as possible so long as any two segments will not intersect except at endpoints, thus a tessellation of triangular faces will be generated[5]. Evidently the triangulation of a certain set of points is not unique. Due to its interesting characteristics, Delaunay triangulation is taken in the paper.

It is worth noticing that the triangulation of flexible contours discussed in this paper is different to that in general purpose, in that the contour segments connecting points are forced to be part of the triangulation, or in another point of view, those point pairs adjacent in contour lines are obliged to be Voronoi neighbors deliberately, and no isolated point exists. Therefore we call this type of triangulation as Delaunay triangulation with internal constraints. For details of the characteristics of Delaunay triangulation with internal constraints and an applicable algorithm, please see [4].

3.2 Texture Mapping based on Triangulation

We need to store the parameters of flexible contour model matched to the first frame of image and the texture information of the same image to obtain the reconstruction. Next parameters of the image to be coded are extracted by matching the model the image. Then the set of vertices corresponding to one of the two sets of parameters will be triangulated by the method mentioned above, and the other will be triangulated according to the triangulation accomplished. At the end, texture mapping will be applied in every triangular region.

4. Experiment Results

First, we use a simple example to demonstrate the capability of flexible contour model to outline objects in image. Fig. 4 shows the process of a single contour matching to a circular object in image. We find only after a few steps of iteration, that the model can represent the shape of the objects quite precisely.

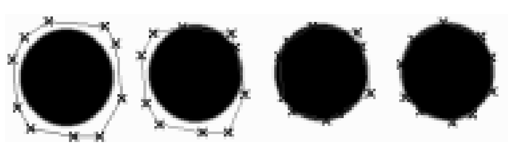


Fig. 4

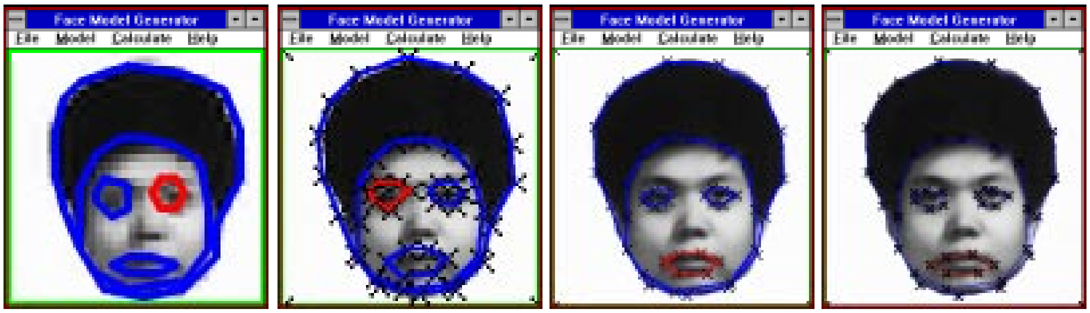
The Matching Process of a Single Contour to an
Image Containing a Circular Object.

From Table 1, we can make a comparison of the two methods for computing gradient force. In the form, $\|\vec{F}g'\|$ refers to the results obtained by the evaluation method while $\|\vec{F}g\|$ by the searching method that may be regarded as the accurate values. It not difficult to notice that the two are very resembling, the maximal error is two pixels.

$\|\vec{Fg}\|$(pixels)	-21	-16	-13	-6	-3	2	3	4	8	12	15	18	25
$\|\vec{Fg}'\|$(pixels)	-21	-16	-12	-6	-3	0	5	6	9	13	16	18	26

Table 1

The comparison of gradient force computed by Searching and Evaluation Method.



(a) (b) (c) (d)

Fig. 5

The Matching Results of Flexible Contour Model to a Facial Image at Resolution from 1/8 to 1.

Fig. 5 (a)-(d) display the matching result of the model to facial image at resolution from 1/8 to 1. For the convenience of observation, the images have been zoomed, so the images are seemingly of the same size but the difference of width of contour lines reveals the difference of resolution.

Fig. 6 is the result of Delaunay triangulation of a facial model and is a reconstruction of facial image.

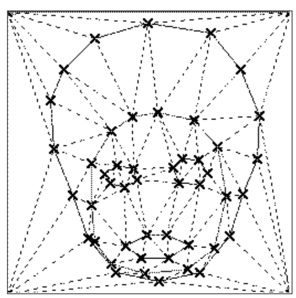
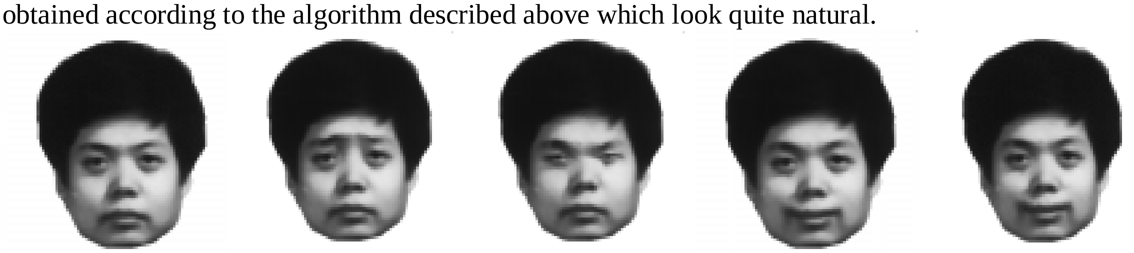


Fig. 6

Delaunay Triangulation of Facial Model

Fig. 7 (a) is an original facial image, while Fig. 7 (b)-(e) are several reconstructed images obtained according to the algorithm described above which look quite natural.



(a) (b) (c) (d) (e)

Fig. 7 (a) Original Image; (b)-(e) Reconstructed Facial Images

5. Conclusion

In the paper, a new image model —flexible contour model was proposed that is different from any 2D and 3D model already reported. The model may be considered as a complex flexible pattern possessing the ability of extracting outlines of objects with irregular shape or tracing the motion of nonrigid object.

The results of the paper indicate that flexible contour model is competent for the task of image analysis and image compression. Although what has been achieved in the paper is far from practicable face-to-face communication system, the authors are confident in the future of flexible contour model.

6. Acknowledgement

This research is partly supported by the National Nature Science Foundation of China, and Trans-Century Training Program Foundation for the Talents by the State Education Commission of China. Dr. Lixing Yuan provided both great help in experimentation and constructive sugestions for the work of this paper.

7. References

[1] M. Kunt, A. Ikonmopolos, and M. Kocher, “Second-Generation Image Coding Technique,” *Proc. IEEE*, vol. 73, pp. 549～574, Apr. 1985.

[2] C. S. Choi, K. Aizawa, H. Harashima and T. Takebe, “Analysis and Synthesis of Facial Image Sequences in Model-Based Image Coding,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 4, no. 3, pp.257～275, June, 1994.

[3] Wei Li, Nanning Zheng and Lixing Yuan, Modelling Facial Image with Flexible Contour Method, In R. Mohr，C. Wu eds. *Proceedings of Europe-China Workshop on Geometrical Modeling and Invariants for Computer Vision，GMICV'95*, Xi’an, Apr. 1995, Xidian University Press, pp.352～355.

[4] Wei Li, Nanning Zheng and Lixing Yuan, Delaunay Triangulation of Flexible Contour Model, In R. Mohr，C. Wu eds. *Proceedings of Europe-China Workshop on Geometrical Modeling and Invariants for Computer Vision，GMICV'95*, Xi’an, Apr. 1995, Xidian University Press, pp.41～46.

[5] L. De Floriani,B. Falcidieno, and C. Pienovi, " Delaunay-based Representation of Surfaces Defined over Arbitrarily Shaped Domains", *Computer Vision, Graphics, And Image Processing*, 32 (1985), 127～140.

Nanning Zheng is a professor of Xi'an Jiaotong University, P. R. China. His research interests include image processing, robot vision, neural networks, sensor-based virtual reality, objects-oriented computer graphics and parallel computation. He received his BE and MS degree from Xi'an Jiaotong University in 1975 and 1981, respectively, and his Ph.D degree from Keio University in Japan in 1985. He is the director of the Institute of Artificial Intelligence and Robotics in Xi'an Jiaotong University.

Wenwei Song is an assistant professor of Xi'an Jiaotong University, P. R. China. Her research interests include signal processing, real-time She got her BE and MS from Xi'an Jiaotong University in 1992 and 1995.

Wei Li is a Ph.D. student in the Computer Science Department of State University of New York at Stony Brook, USA. His research interests include Computer Graphics, Visualization, Internet Application, Multimedia and Computer Vision. He got his M.S and B.S degree in 1995 and 1992 respectively, both from Xi'an Jiaotong University, P. R. China.