A Robust Model Generation Technique for Model-Based Video Coding

Manson Siu, Yuk-Hee Chan, and Wan-Chi Siu

***Abstract*—In conventional model-based coding schemes, predefined static models are generally used. These models cannot adapt to new situations, and hence, they have to be very specific and cannot be generated from a single generic model even though they are very similar. In this letter, we present a model-generation technique that can gradually build a model and dynamically modify it according to new video frames scanned. The proposed technique is robust to the object's orientation in the view and can be efficiently implemented with a parallel processing technique. As a result, the proposed technique is more attractive to the practical use of model-based coding techniques in real applications.**

***Index Terms*—Model-based coding, model synthesis.**

I. Introduction

THE MODEL-BASED coding system is the hottest member in the object-based coding system. A vast amount of research has been carried out on relevant topics such as feature extraction [1], motion tracking [2], and model-synthesizing techniques [3], [4], but model generation is still a time-consuming procedure that is not attractive for real-time applications.

In conventional model-based coding schemes, pre-defined models are generally used. These models are static and can neither be adaptive to fit the real features of the object of interest nor be updated dynamically in the course of application. Accordingly, a very specific model is necessary for a particular object, and one cannot use a generic model to generate models for similar objects whenever it is necessary so as to save memory space. Therefore, for applications involving real-time video coding, a simple model-generation method that can easily construct a model or modify it dynamically according to the scenes processed is necessary.

There are methods for generating an object model with stereo graphics [5] and laser scanning. Laser scanning can provide a very accurate model for an object, but the size and the cost of the equipment involved makes it infeasible for the aforementioned applications. As for methods based on stereo graphics, they generally work only on specific views. Because of these reasons, practical model-synthesis technique becomes an important investigation in our model-based coding research.

In this letter, we propose a new technique to generate a model for a particular object with a generic model. The proposed approach has several advantages over the conventional methods using stereo graphics. First, no specific view with known orientation of the object is required. Second, updating the generic model with information extracted from multiple views is supported. Third, the feature-extraction processes for different views can be done independently, and hence parallel processing is allowed. Accordingly, our method is comparatively more efficient and practical for real applications.

II. Robust Model Generation

In the proposed approach, a generic model for objects of similar shapes is used in the model-based encoding and decoding processes. The usage of the generic model can achieve very low bit-rate coding, which benefits communication applications. The idea of using the generic model is to use a standard collection of graphics models to represent standard objects. However, a single object may have various shapes that a standard model may not represent well. Our proposed algorithm can solve this problem easily as it can dynamically adapt a generic model to the shape of the object of interest whenever new information is available.

Note that the algorithm presented in this letter can actually operate in two modes for different applications. In its first operation mode, a target model can be generated as a pre-defined model for later applications based on different views of the object of interest. In its second operation mode, the target model can be gradually built or updated with new information extracted from new video frames being processed. Parallel processing is allowed in both circumstances.

Fig. 1 shows the process flow of the proposed model-generation scheme. In the following part of this section, we shall discuss the processes involved in details.

A. Orientation Extraction

In order to extract matching information for generating the object's model with the generic model on hand, the orientation of the object appeared in the captured image or video frames must be known. This piece of information can be obtained via an analysis of the image context. Various research for this purpose have been done. Typical methods include those use face features to estimate a human head's orientation such as *feature-location tracking* [1] and *point-matching* [2]. In our approach, we adopt the *point-matching* technique [2] to determine the object's orientation in a captured image.

B. Visible Surface Determination

After we have identified the orientation of the object, we rotate the generic model accordingly to match their orientations. The visible surface of the model is then determined with a modified Z-buffering algorithm. In order to increase the efficiency and reduce the computation effort, we project the vertices onto

Manuscript received July 21, 1999; revised September 30, 2000. This work was supported by The Hong Kong Polytechnic University under Grant A/C V592 and Center of Multimedia Signal Processing. This paper was recommended by Associate Editor H. J. Zhang.

The authors are with the Center of Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: mansonsiu@hotmail.com; enyhchan@hotmail.com; enwcsiu@polyu.edu.hk).

Publisher Item Identifier S 1051-8215(01)10157-6.

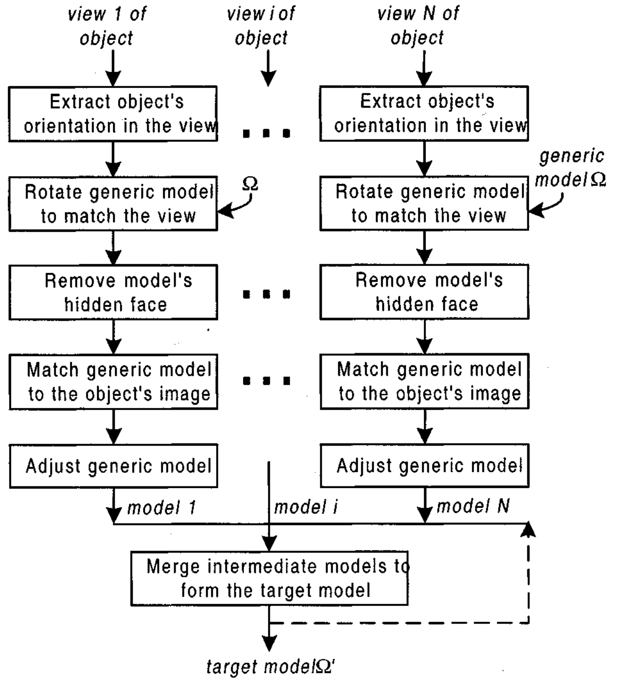
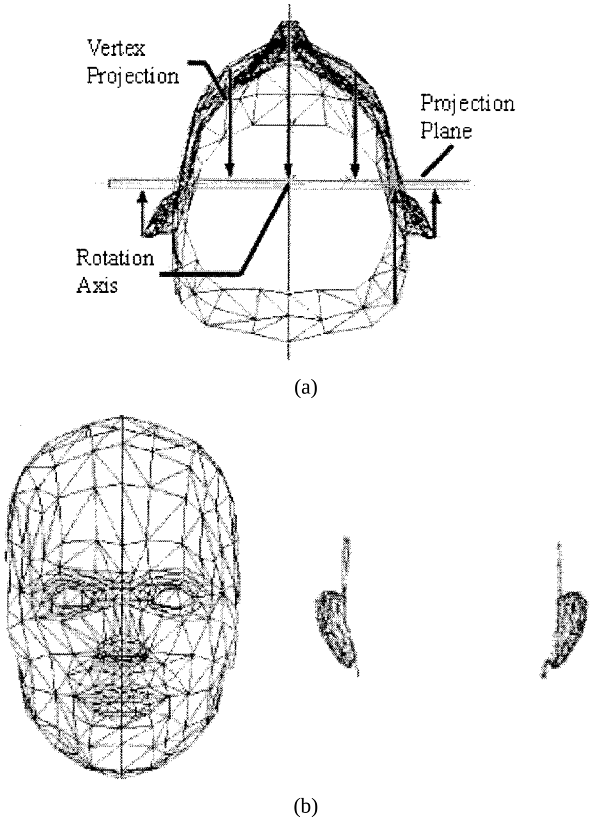

view 1 of object
view i of object
view N of object
Extract object's orientation in the view
Extract object's orientation in the view
generic model Ω
Ω
Rotate generic model to match the view
Rotate generic model to match the view
Remove model's hidden face
Remove model's hidden face
Match generic model to the object's image
Match generic model to the object's image
Adjust generic model
Adjust generic model
model 1
model i
model N
Merge intermediate models to form the target model
target model Ω'

Fig. 1. Flow of the model synthesis process.



(a)

(b)

Fig. 2. Visible surface determination. (a) Vertex projection. (b) Visible surfaces.

a projection plane that passes the center of the model and is parallel to the view plane. Fig. 2 shows the visible face of a head model and how it is obtained. The left portion of Fig. 2(b) shows the surface closer while the right one shows the surface farther than the projection plane to the viewpoint.

C. Matching a Generic Model to an Object's Image

The processed model will then undergo a model-matching procedure. First, a projected view of the processed model is obtained by projecting the model onto the view plane with orthographic projection. Then, the boundary information of this projected view and the object's image is extracted. After translating the generic model to make the center of its projection coincident with the center of the object's image, the model is adjusted such that the boundary of the projected view of the model matches that of the object's image.

Besides the boundary vertices, remarkable feature points like eye and mouth corners are also modified in this process. The corresponding feature points on the generic model will be matched to that of the object's image by making use of feature matching techniques. The feature-matching technique proposed by Tang and Thomas in [3] is adopted in our system to match face features. Fig. 3 shows the intermediate result of the process.

D. Modifying Generic Model

Modifying the position of a vertex in an object model will result in a change of the positions of its connected vertices. Their change is not uniform. The closer a vertex from the adjusted vertex, the more significant the change should be. The modification algorithm we adopted here is an iterative algorithm.

Fig. 4 shows a particular vertex V_m of the model and its connected neighbors. In general, the resultant displacement V_m due to its neighboring vertices' displacements, say $\vec{R}_m$, is given as

$$\vec{R}_m = \sum_{V_n \in \Lambda_m} \vec{r}_n \times \frac{w_{m,n}}{w_m} \tag{1}$$

where $\vec{r}_n$ is the radial displacement of vertex V_n, and Λ_m is the set of V_m's connected neighboring vertices. The weighting parameters $w_{m,n}$ and w_m are, respectively, defined as

$$w_{m,n} = \frac{\sum\limits_{V_k \in \Lambda_m} d_{m,k}}{N \cdot d_{m,n}} \tag{2}$$

and

$$w_m = \sum_{V_n \in \Lambda_m} w_{m,n} \tag{3}$$

where N is the total number of elements in Λ_m and $d_{m,n}$ is the distance between vertices V_m and V_n.

The vertices are updated one by one until all of them are processed. These procedures are then repeated again and again until there is no more change of the resultant displacement for every vertex in the model.

E. Updating Resultant Model

For a particular view of the object, a new model can be obtained by adjusting the generic model with the aforementioned procedures. For the sake of reference, we refer to the model associated with view orientation i as Ω'_i. This intermediate model will then be used to update the target model Ω' in order to produce a natural model.

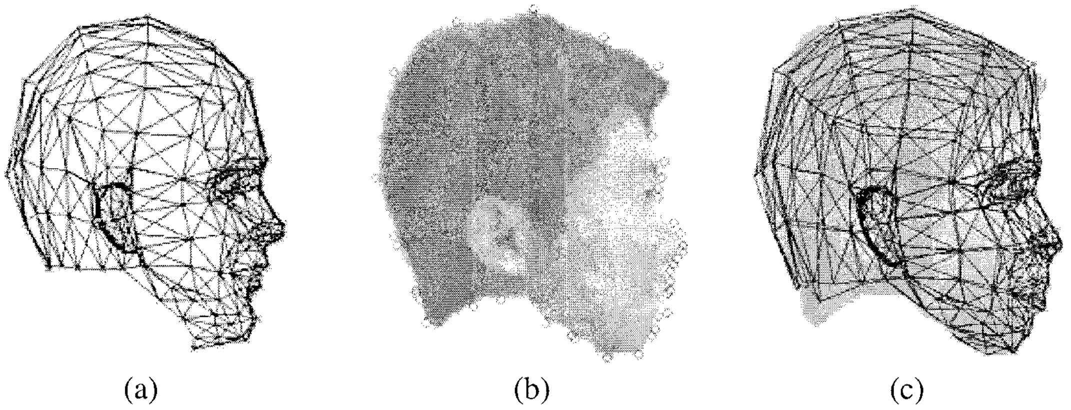


Fig. 3. Model matching: (a) projected view of a model; (b) corresponding positions of the vertices appeared at the boundary of the model's projected view; and (c) adjusted model.

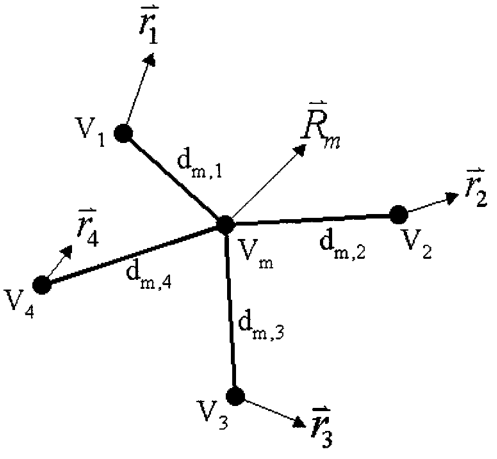


Fig. 4. Corresponding resultant displacement and its connected neighbors.

Let the vertices of model Ω_i' be $\{V_{i,j}: j = 1, 2 \cdots K\}$, where K is the total number of vertices of the model. The line connecting the origin to a particular vertex $V_{i,j}$ defines a vector $(\alpha_{i,j,x}, \alpha_{i,j,y}, \alpha_{i,j,z})$, as shown in Fig. 5. The component $\alpha_{i,j,z}$ is used as an input parameter to calculate $S_{i,j}$, the confidence level of $V_{i,j}$ with respect to its counterparts $V_{k,j}$'s, where $k \neq i$. Specifically, $S_{i,j}$ is given by

$$S_{i,j} = \begin{cases} 1 - |\cos \alpha_{i,j,z}|, & \text{if } V_{i,j} \text{ is visible in view } i \\ 0, & \text{else.} \end{cases} \tag{4}$$

Hereafter, this equation is referred to as *significance equation*, as it tells the significance of a particular vertex of Ω_i' in updating Ω'.

The coordinates of the corresponding vertex in Ω', V_j', are then updated with the following rule accordingly:

$$V_j' = \begin{cases} V_{i,j}, & \text{if } S_j < S_{i,j} \\ V_j', & \text{else.} \end{cases} \tag{5}$$

Threshold S_j is the maximum value of $S_{k,j}$'s for all processed views k's. It is initialized to be zero at the very beginning. Since the computation of $S_{i,j}$'s is independent of each other, the model can be updated in parallel. This increases its efficiency.

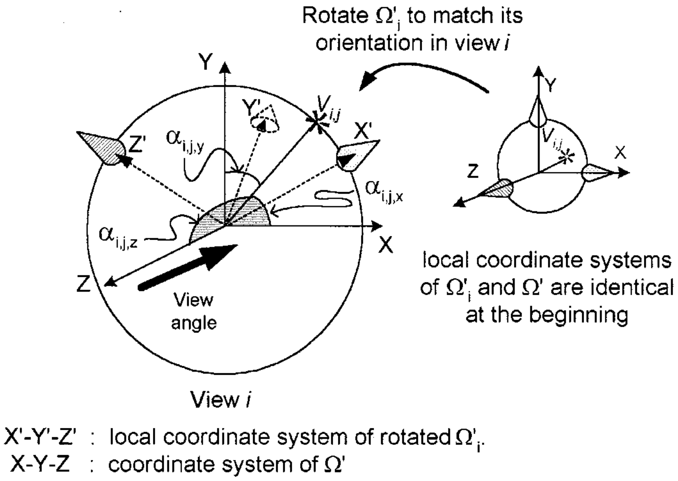

Rotate Ω'_i to match its orientation in view i
Y
Y'
$V_{i,j}$
X'
Z'
$\alpha_{i,j,y}$
$\alpha_{i,j,x}$
$\alpha_{i,j,z}$
X
Z
View angle
View i
Y
$V_{i,j}$
X
Z
local coordinate systems of Ω'_i and Ω' are identical at the beginning
X'-Y'-Z' : local coordinate system of rotated Ω'_i.
X-Y-Z : coordinate system of Ω'

Fig. 5. Vertex j of the adjusted model associated with view orientation i.

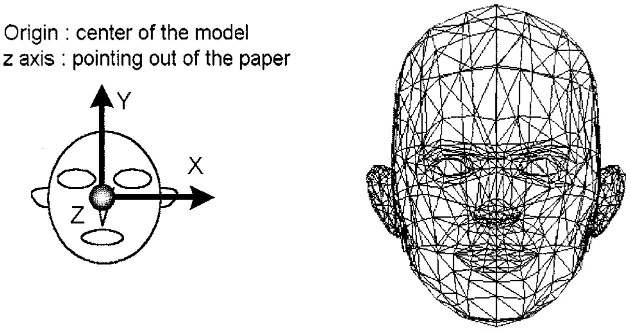

Origin : center of the model
z axis : pointing out of the paper
Y
X
Z

Fig. 6. Coordinate system and generic model used in the simulation.

In our approach, the target model is initialized to be the generic model. The updating rule (5) guarantees that the target model converges and eventually converges into an ultimate state as long as sufficient views are provided if the object is rigid. Since V_j' is always updated to be $V_{i,j}$, the confidence level of which is maximum, it is obvious from (4) that, in the ultimate target model, all vertices are of confidence level 1.

The number of iterations required to obtain the ultimate target model is input dependent. If a well-organized input sequence of

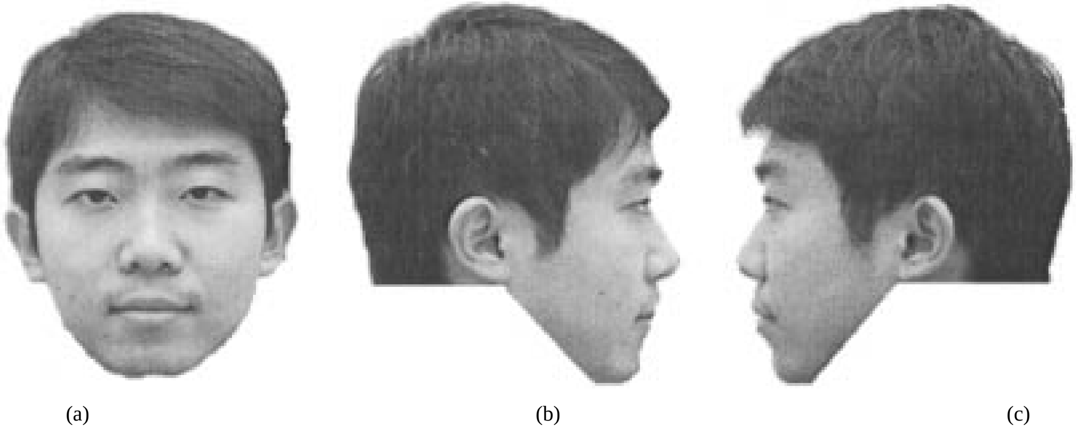


Fig. 7. Three test input images in different orientations.

views are provided, the ultimate target model can be obtained with a few iterations. However, if a particular view of the object never appears, it is possible that the confidence levels of some vertices of Ω' are less than 1 and the ultimate target model cannot be obtained forever.

In practical applications, such as model-based coding, it is generally not necessary to obtain the ultimate target model. Once a particular view of the concerned object appears, it can be incorporated into the current target model. The coding process can be based on the most updated target model instead of the ultimate target model, as it already contains the best knowledge we have had. Another argument is that, if some views of the object of interest never appear, we need not bother with them and it is sufficient to work with a target model which is not ultimate in a way that the coordinates of the hidden vertices are not properly defined.

When the object of interest is not rigid, it is impossible to get an ultimate target model, as some vertices of its model may not be stable. Using a head model to encode head-shoulder sequences is one of the typical examples. In such a case, the coding process should be based on the most updated target model.

III. Simulation Result

Simulation has been carried out to evaluate the performance of the proposed algorithm. The coordinate system and the generic model used in the simulation are shown in Fig. 6. Based on the generic model on hand, a single spherical texture map is first generated with three different views shown in Fig. 7 for later application [6].

The generic model is fitted to the object's image according to the object's orientation in the image. Our proposed method is then used to adjust the generic model to form three intermediate models. In Figs. 8–10, the intermediate models generated from the three different views shown in Fig. 7 are shown.

The intermediate model generated in each view is then used to update the target model. The extraction procedures in obtaining the update model can be optimized with parallel computing. The confidence levels of the vertices of the intermediate models are calculated and the target model is modified with a reference to these parameters. Fig. 11 shows five views of the target model.

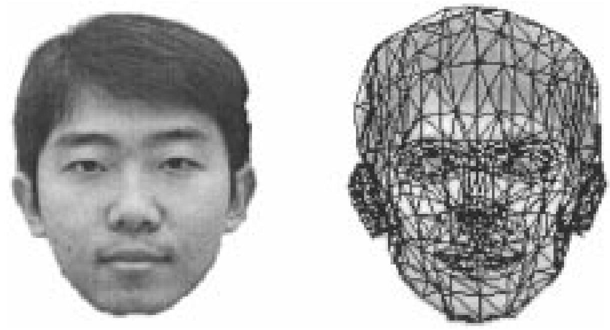


Fig. 8. The intermediate model obtained with view 7(a).

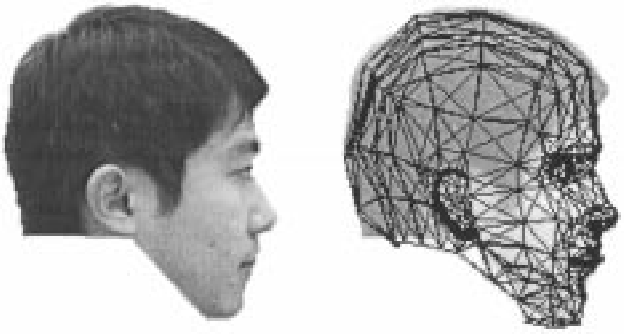


Fig. 9. The intermediate model obtained with view 7(b).

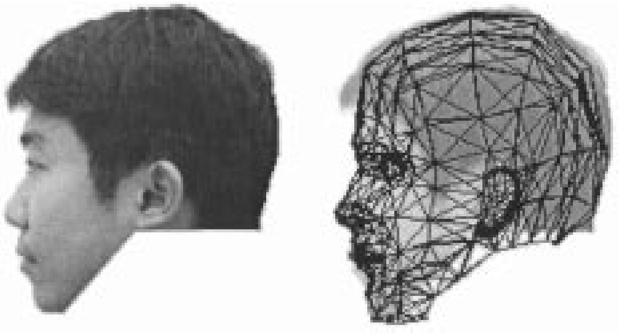


Fig. 10. The intermediate model obtained with view 7(c).

One can see that three appropriate views are enough to adapt a generic model to an object.

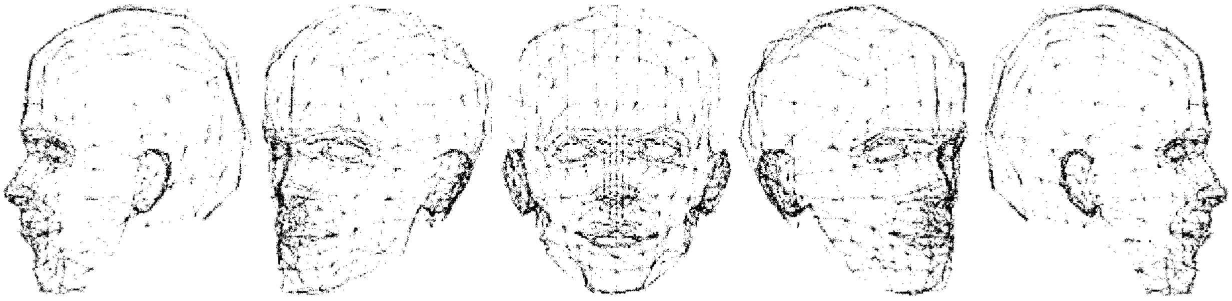


Fig. 11. Resultant model in the simulation.

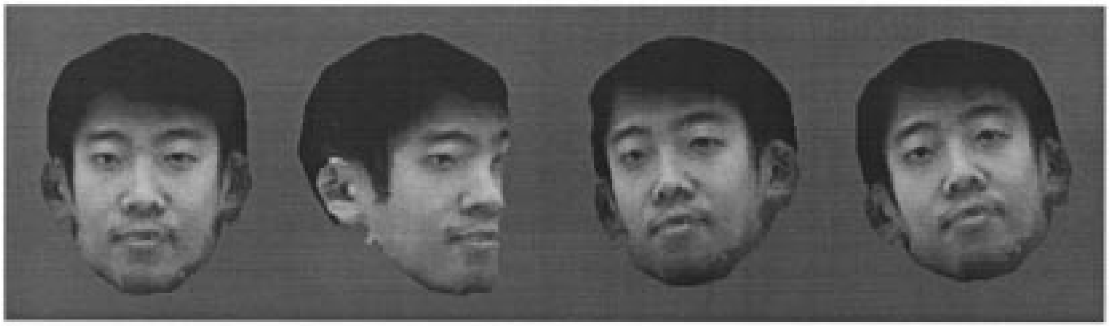


Fig. 12. Rendered graphics models with a spherical texture map.

Once the target model is obtained, one can map the spherical texture to the target model to construct different views of the object. Fig. 12 shows some texture-mapped results obtained with the target model.

IV. Conclusion

In our work on model synthesis, we concern the practicality of model-based coding techniques. With the help of our model-generation technique, several advantages can be achieved.

1) It is robust to the object's orientation and size in the view.
2) The resultant model can be gradually built or updated with model update parameters extracted from different views under the guidance of a set of significance functions.
3) The extraction of update parameters from different views can be carried out in parallel without interference to each other.

Synthesizing a graphical model is always a time-tolerating procedure in model-based coding systems. With the help of our approach, a specific human head model can be obtained more efficiently as compared with other conventional approaches, which is therefore more attractive to the practical use of model-based coding techniques in real applications.

References

[1] M. Kokuer and A. F. Clark, "Feature and model tracking for model-based coding," in *Proc. Int. Conf. Image Processing and Its Applications*, 1992, pp. 135–138.

[2] K. M. Lam and H. Yan, "An analytical-to-holistic approach for face recognition based on a single frontal view," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 673–686, July 1998.

[3] L. A. Tang and T. S. Huang, "Automatic construction of 3D human face models based on 2D images," in *Proc. IEEE ICIP'96*, vol. 3, Lausanne, Switzerland, 1996, pp. 467–470.

[4] L. Yin and A. Basu, "MPEG4 face modeling using fiducial points," in *Proc. IEEE ICIP'97*, vol. 1, Santa Barbara, CA, USA, 1997, pp. 109–112.

[5] L. H. Chen and W. C. Lin, "Visual surface segmentation from stereo," *Image and Vis. Comput.*, vol. 15, pp. 95–106, 1997.

[6] M. Siu and Y. H. Chan, "A robust universal texture extraction technique for model-based coding," in *Proc. SPIE, Vision Geometry VIII*, vol. 3811, Denver, CO, 1999, pp. 329–336.