

# A ROBUST MODEL GENERATION TECHNIQUE FOR MODEL-BASED CODING

Manson SIU and Yuk-Hee CHAN

*Center of Digital Signal Processing for Multimedia Applications  
Department of Electronic and Information Engineering  
The Hong Kong Polytechnic University, Hong Kong*

## 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 paper, 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 parallel processing technique. As a result, the proposed technique is more attractive to the practical use of model-based coding techniques in real applications.

**Keywords:** Model synthesis, Model-based Coding

## 1. INTRODUCTION

The model-based coding system is the hottest member in the object-based coding systems. 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 paper, 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 of all, 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, 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.

## 2. 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 generic model can achieve very low bit rate coding which benefits communication applications. The idea of using 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 well represent them. Our proposed algorithm can solve this problem easily as it can make a generic model dynamically adapt to the shape of the object of interest whenever new information is available.

Note the algorithm presented in this paper 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.

Figure 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.

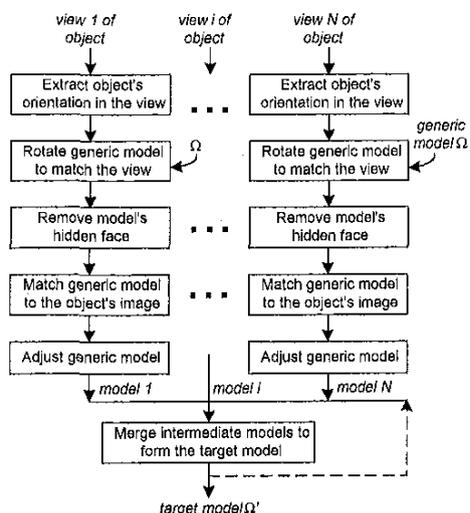


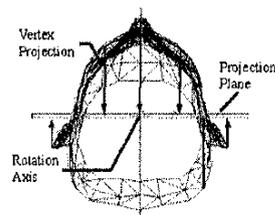
Figure 1. Model Synthesis Process Flow chart

### 2.1 Orientation extraction

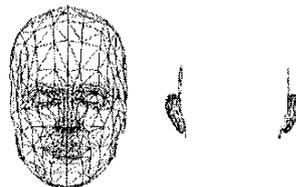
In order to extract matching information for generating the object's model with the generic model on hand, the orientation of the object appeared in the captured image or video frames must be known. This piece of information can be obtained via an analysis of the image context. Various researches for 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.

### 2.2 Visible surface determination

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



(a) Vertex Projection



(b) Visible surfaces

Figure 2. Visible surface determination

### 2.3 Matching a generic model to an object's image

The processed model will then undergo a model matching procedure. First of all, 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.

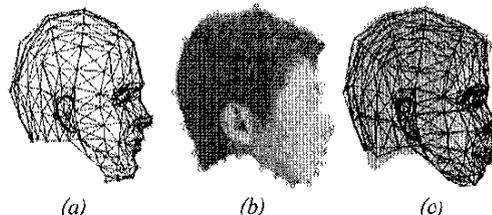


Figure 3. Model Matching: (a) Projected view of a model (b) Corresponding positions of the vertices appeared at the boundary of the model's projected view (c) Adjusted model

Besides the boundary vertices, remarkable feature points like eye corners 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. Figure 3 shows the intermediate result of the process.

## 2.4 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.

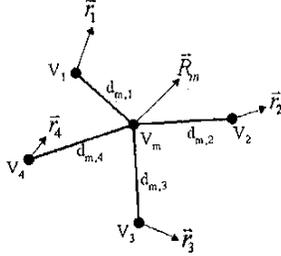


Figure 4. Corresponding resultant displacement and its Connected neighbors

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

$$\bar{R}_m = \sum_{V_n \in \Lambda_m} \bar{r}_n \times \frac{w_{m,n}}{w_m} \quad (1)$$

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

$$w_{m,n} = \frac{\sum_{V_k \in \Lambda_m} d_{m,k}}{N \cdot d_{m,n}} \quad (2)$$

$$\text{and } w_m = \sum_{V_n \in \Lambda_m} w_{m,n} \quad (3)$$

where  $N$  is the total number of elements in  $\Lambda_m$  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.

## 2.5 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 the model associated with view orientation  $i$  to  $\Omega'_i$ .

This intermediate model will then be used to update the target model  $\Omega'$  in order to produce a natural model.

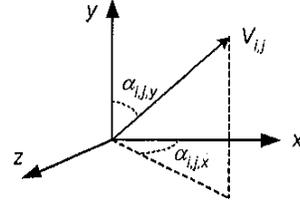


Figure 5. Representation of a corresponding vertex direction  $i$

Let the vertices of model  $\Omega'_i$  be  $\{V_{i,j}; j=1,2,\dots,K\}$ , 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  $(\alpha_{i,j,x}, \alpha_{i,j,y})$  as shown in figure 5. This vector is used as an input parameter to calculate  $S_{i,j,x}$ ,  $S_{i,j,y}$  and  $S_{i,j,z}$ , the confidence levels of the three components of  $V_{i,j}$ 's coordinate with respect to its counterparts  $V_{k,i}$ 's, where  $k \neq j$ . Specifically,  $S_{i,j,x}$ ,  $S_{i,j,y}$  and  $S_{i,j,z}$  are given by

$$\begin{aligned} S_{i,j,x} &= |\cos \alpha_{i,j,x}| \times (1 - |\cos \alpha_{i,j,y}|) \\ S_{i,j,y} &= 1 - |\cos \alpha_{i,j,y}| \\ S_{i,j,z} &= (1 - |\cos \alpha_{i,j,x}|) \times |\cos \alpha_{i,j,y}| \end{aligned} \quad (4)$$

Hereafter, this set of equations is referred to as *significance equations* as it tells the significance of a particular vertex component of  $\Omega'_i$  in updating  $\Omega'$ .

The coordinate of the corresponding vertex in  $\Omega'$ , say  $(V'_{j,x}, V'_{j,y}, V'_{j,z})$ , is then updated with the following set of rules accordingly.

$$\begin{aligned} V'_{j,x} &= \begin{cases} V_{i,j,x} & \text{if } S_{j,x} < S_{i,j,x} \\ V'_{j,x} & \text{else} \end{cases} \\ V'_{j,y} &= \begin{cases} V_{i,j,y} & \text{if } S_{j,y} < S_{i,j,y} \\ V'_{j,y} & \text{else} \end{cases} \\ V'_{j,z} &= \begin{cases} V_{i,j,z} & \text{if } S_{j,z} < S_{i,j,z} \\ V'_{j,z} & \text{else} \end{cases} \end{aligned} \quad (5)$$

Thresholds  $S_{j,x}$ ,  $S_{j,y}$  and  $S_{j,z}$  are, respectively, the maximum values of  $S_{k,j,x}$ 's,  $S_{k,j,y}$ 's and  $S_{k,j,z}$ 's for all processed views  $k$ 's. They are all initialized to be zero at the very beginning. Since the computation of  $S_{i,j,x}$ ,  $S_{i,j,y}$  and  $S_{i,j,z}$  is independent of each other, the model can be

updated with respect to the 3 coordinate components in parallel. This increases its efficiency.

In our approach, the target model is initialized to be the generic model. The updating rule (5) guarantees that the target model can eventually converge into a stable state after a certain number of iterations on different views.

### 3. 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 figure 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].

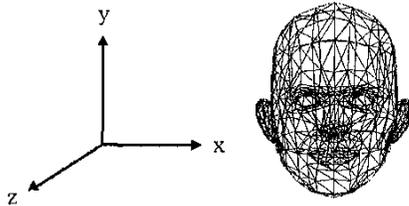


Figure 6. Coordinate system and generic model used in the simulation

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 figures 8, 9 and 10, the intermediate models generated from the three different views shown in figure 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. Figure 11 shows five views of the target model.

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

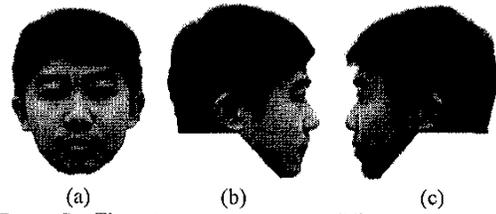


Figure 7. Three test input images in different orientations

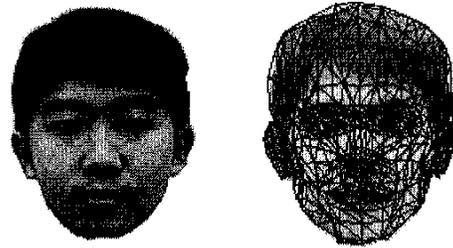


Figure 8. The intermediate model obtained with view 7a

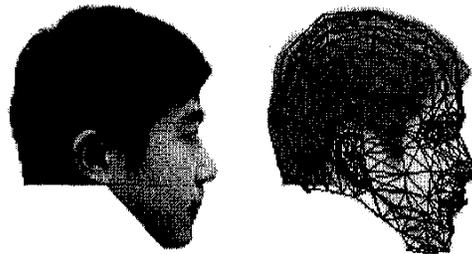


Figure 9. The intermediate model obtained with view 7b

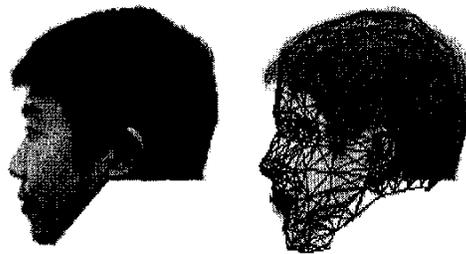


Figure 10. The intermediate model obtained with view 7c

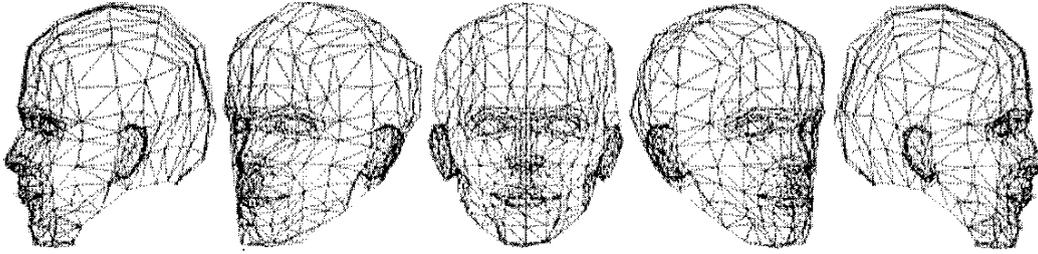


Figure 11. Resultant model in the simulation



Figure 12. Rendered graphics models with a spherical texture map

#### 4. CONCLUSIONS

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.

#### ACKNOWLEDGMENTS

This work was supported by The Hong Kong Polytechnic University under Grant A/C V592.

#### REFERENCES

- [1] M. Kokuer and A. F. Clark, "Feature and model tracking for model-based coding", *Image processing and its applications*, pp. 135-138, 1992.
- [2] Kin-Man Lam and Hong Yan, "Face recognition using point-matching based on a single front view", *RWC'97*, pp.418-423, Japan, 1997.
- [3] Li-an and Thomas S. Huang, "Automatic construction of 3D human face models based on 2D images", *IEEE ICIP*, vol. 3, pp. 467-470, 1996.
- [4] Lijun Yin and Anup Basu, "MPEG4 face modeling using fiducial points", *IEEE ICIP*, vol. 1, pp.109-112, 1997.
- [5] Liang-Hua Chen and Wei-Chung Lin, "Visual surface segmentation from stereo", *Image and Vision Computing*, vol. 15, 1997.
- [6] Manson Siu and Yuk-Hee Chan, "A Robust Universal Texture Extraction Technique for Model-based Coding", *Proceedings of SPIE Vol. 3811, Conference 3811, Vision Geometry VIII*, 19-20 July 1999.