Novel view image synthesis based on photo-consistent 3D model deformation

Meng-Hsuan Chia, Chien-Hsing He and Huei-Yung Lin*

Department of Electrical Engineering,
National Chung Cheng University,
Chia-yi 621, Taiwan
Fax: 886-5-272-0862
E-mail: silverwolf8291@hotmail.com
E-mail: sonily7811@hotmail.com
E-mail: hylin@ccu.edu.tw
*Corresponding author

Abstract: In this paper, a system is designed for improving the quality of novel view synthesis. To make the virtual view synthesis look better, the basic idea is to perform model refinement with important camera information of the novel view. In this system, we first use the reconstructed visual hull from shape from silhouette and then refine this 3D model based on the view dependency. The 3D points of the model are classified into the outline points and the non-outline points according to the virtual viewpoint. To refine the model, both of the outline points and the non-outline points are used to move the 3D points iteratively by minimising energy function until convergence. The key energy is the photo-consistency energy with additional smoothness energy and contour/visual hull energy. The latter two energy terms can avoid the local minimum when calculating the photo-consistency energy. Finally, we render the novel view image using view-dependent image synthesis by blending the pixel values from reference cameras near the virtual camera.

Keywords: 3D model reconstruction; view synthesis; visual hull.


Biographical notes: Meng-Hsuan Chia received his BS and MS in Electrical Engineering from National Chung Cheng University, Taiwan. He has been engaging in the research areas of computer vision and image processing. He is currently with Institute for Information Industry, Taiwan.

Chien-Hsing He received his BS in Electrical Engineering from National University of Kaohsiung, Taiwan. He is currently an MS student at the Department of Electrical Engineering, National Chung Cheng University, Taiwan. His research interests include computer vision and image processing.

Huei-Yung Lin received his PhD in Electrical and Computer Engineering from the State University of New York at Stony Brook. In 2002, he joined the Department of Electrical Engineering, National Chung Cheng University, Taiwan, as an Assistant Professor and currently an Associate Professor. His current research interests include computer vision, robotics, pattern recognition, and image processing.

This paper is a revised and expanded version of a paper entitled ‘Novel view image synthesis based on photo-consistent 3D model deformation’ presented at the 2011 National Computer Symposium, Taiwan, 2–3 December 2011.

1 Introduction

In the past few decades, three-dimensional (3D) reconstruction has been widely investigated, and the results are used to many 3D applications like 3D video game, 3D movie, virtual reality, medical imaging, or digital art collection. Many researchers (e.g., Furukawa and Ponce, 2010; Delaunoy et al., 2008; Goesele et al., 2006) produce more and more precise 3D models, but they always spend over one hour for calculation.

Since it takes a long time to reconstruct a 3D model, we deal with this problem with a different approach. It is the common cases that two-dimensional (2D) monitors are still accepted by most consumers. That is, no matter how precise the 3D model is, it needs to be converted to 2D images for display. Thus, in our approach, we reconstruct the objects mainly specifically for novel view image synthesis.

Although many previous works have discussed how to reconstruct more precise 3D models, most of them are not
specifically designed for novel view synthesis. (Matsuyama et al., 2004) proposed a parallel pipeline processing method for reconstructing a dynamic 3D object shape from multi-view video images. Different from shape from silhouettes algorithms, they introduced a plane-based volume intersection algorithm. The voxel-based representation of the 3D model is then transformed into meshes. Finally, the 3D model is rendered by view-independent and view-dependent: the former determines which cameras are used to blend by point normal, and the latter performs the novel view synthesis according to the viewing direction.

Goesele et al. (2006) proposed a simple concept which reconstructed 3D models only using photo-consistency. They projected every pixel at every viewpoint into the 3D space, and find the most matching voxel which has similar colour with the other viewpoint. In this case, it might produce highly accurate 3D points. However, some regions cannot be reconstructed due to the occlusion problem caused by low similarities of the colour information.

Ukita et al. (2008) proposed a free-viewpoint imaging method with sparsely located cameras. It aims to be used in a complicated scene. Different from stereo vision, their cameras are separated by about 1.5 metres. They found the corresponding points for two cameras, and used them to create meshes for these images. Finally, these warped corresponding meshes were blended in the novel view images. Since it made some neighbouring meshes become discontinuous along their edges, they split the incorrect mesh according to photo-consistency for reducing effect of discontinuous regions.

In many researches of multi-view 3D model reconstruction, shape from silhouette algorithm is the most popular one (Laurentini, 1994). It projects the silhouettes of the captured images into the 3D space, and creates the visual hull based on their intersection regions. This method reconstructs fast but only a rough result can be obtained because no detailed texture information is used. More importantly, the concave region of the 3D model cannot be reconstructed with the visual hulls. Thus, we will improve this drawback using the photo-consistency property among the images.

Photo-consistency for 3D model reconstruction was presented at first by Seitz and Dyer (2002), and improved by Kutulakos and Seitz (2000). They used voxel-based representations to partition the 3D space into $W \times H \times L$ voxels, and then project each voxel to the visible images and calculate their colour differences. A voxel is a true point if the difference is smaller than a threshold. Using this approach, more precise 3D points can be obtained. But it usually spends too much time for the 3D model refinement.

Hence, our method aims to combine the advantages of the above techniques and apply it to image synthesis. In this paper, we first reconstruct the visual hull and remove the meshes invisible from the novel viewpoints, and then refine the model by photo-consistency check. Similar to the active contours, the 3D model refinement is defined via the minimum energy function. According to the position and the viewing direction of the novel view, 3D points are classified as outline points and non-outline points. Each type has its own energy function term. Finally, we use view-dependent image synthesis to render a high quality image for the given viewpoint.

2 Model refinement and novel view synthesis

Our system consists of three stages:
1 3D reconstruction
2 model refinement
3 view-dependent image synthesis.

In the model refinement stage, 3D points are classified as the outline points and the non-outline points according to the virtual viewpoint, and processed separately. The overall system flowchart is shown in Figure 1.

Figure 1 System flowchart of the proposed model refinement and novel view synthesis technique

2.1 3D reconstruction

To generate the initial 3D model, we first create the visual hull using shape from silhouettes algorithm. It projects every viewable region of the images to the 3D space, which is named visual cone. The visual hull is the intersection volume of all visual cones. Figure 2 shows a cross-section view when processing the visual hull. This method can only reconstruct a rough 3D model, but it requires less computation time. See Lin and Wu (2008) for more details.

We then adopt the powercrust algorithm (Amenta et al., 2001; Aurenhammer, 1991; Lee and Schachter, 1980) to generate the meshes connecting all 3D points with their neighbours. It contains the following steps:
1 Calculate the Voronoi diagram.
2 Determine if the vertices of the Voronoi diagram are poles.
3 Calculate the power diagram of the poles.
4 The poles are classified into intrinsic points and extrinsic points.
Create a surface of the power diagram so that the intrinsic and extrinsic points can be separated.

Connect all intrinsic points using Delaunay triangulation (Lee and Schachter, 1980) and form the mesh model.

The result 3D model is fully coated with meshes generated from triangulation. In this case, a surface normal vector can be represented by the cross product of any two vectors in the mesh. The 3D mesh model can detect the self-occlusion areas when synthesising the novel view images. If we need to find the neighbours of a point, the mesh model is also efficient since the connection information of points has been created. There is no need to search from all of the points.

Since the invisible meshes are not required for novel view synthesis, they are removed according to the given virtual viewpoint. Our invisible mesh removal algorithm is divided into two stages:

1. For each mesh, a triangular viewing cone is generated by the centre of projection $P_c$ and the triangle $P_1P_2P_3$ as shown in Figure 3. It is used to determine whether a 3D point, say $P$, is invisible or not. If the point is inside this infinite triangular space, then it is invisible because of the occlusion introduced by the mesh $P_1P_2P_3$.

2. After the invisibility test has been carried out for all points, whether a mesh is visible can then be determined accordingly. A mesh is invisible only if all its three vertices are not visible from the centre of projection.

### 2.2 Refinement of outline points

When the 3D model is projected to a novel view image plane, it is relatively easy to identify the model’s contour. Thus, we can refine the ‘outline points’ first. The outline points are defined as the image points on the contour of the 3D model’s projection silhouettes in the novel view image plane. Our refinement will move the outline points progressively for better image quality. The flowchart of our outline point refinement is shown in Figure 4.

At the second step, we should define the moving direction before every iterative movement. We project the outline points to the 3D model silhouettes and capture a $5 \times 5$ window, an example is shown in Figure 5. The curve indicates the contour of the 3D model silhouettes; $P_x$ is the projection points of the outline points. In the binary $5 \times 5$ window, the white region is the visible 3D model silhouettes and the black region is the background. We then find the centre of gravity of the background, $P_g$. The 2D vector $V_p$ is the normalised $P_xP_g$ at image plane, but $V_p$ is not enough because the outline point is in 3D space. Thus, we change $V_p$ from the image coordinate to the world coordinate using intrinsic and extrinsic camera parameters of the novel view. The final moving direction is given by the 3D vector.
We construct an energy function and the final position of the outline point is found when it has the minimum energy. The energy function is defined as

$$E = E_s + \alpha E_c + \beta E_r$$  

where $E_s$ is the photo-consistency energy in that the colour difference between the projections of outline points to different reference images. $E_c$ is the distance between the model contour in the neighbouring image and the projection point of the outline point, which is called the contour energy. $E_r$ is a smoothness energy which acts as an attraction of neighbouring 3D points to the outline point. Each energy will be discussed in more details at a later stage. The weights $\alpha$ and $\beta$ are user-defined parameters.

### 2.2.1 Photo-consistency energy

Photo-consistency plays an important role in stereo vision, which is used to find the coordinates in the 3D space using the corresponding points in a stereo image pair. In our system, it is used to check the consistency of the colour information in multi-view images. Thus, we choose a key image which is closest to the virtual viewpoint and some secondary images which are the neighbours of the key image.

The outline points are then projected onto the key image and secondary images. Taking those projection points as centres, a key window and secondary windows are selected alternately. We calculate the sum of squared differences (SSD) values of the key window with each secondary window separately. The photo-consistency energy is computed as the average of the sum of the SSD values

$$E_s = \frac{\sum_{k=0}^{n_{\text{sub}}-1} w_k}{n_{\text{sub}}}$$  

where $n_{\text{sub}}$ is number of secondary images and $w_k$ is the SSD value.

Since the projections of the outline points should always be around the contour and the key window or secondary windows may grab the background pixel that we do not need, we create a binary foreground mask to handle this situation. For each pixel of the mask, it is set as 0 if the corresponding pixel is the background in the key image or any secondary images, otherwise it is set as 1. On the other hand, we do not calculate the colour difference when any pixel is on the background. So it is written as:

$$\sum_{x,y} \left[ I_{\text{key}}(x, y) - I_{\text{sub}}(x, y) \right]^2$$

$$\sum_{x,y} I_{\text{key}}(x, y)^2 \cdot \sum_{x,y} I_{\text{sub}}(x, y)^2 \cdot M_f(x,y)$$  

### 2.2.2 Contour energy

The outline points should locate around the contour of the neighbouring view images unless the angle between the novel view and its neighbours. Hence, we set the image contour as an energy of attraction. The outline points are projected to each image of the neighbouring viewpoint $I_k$ ($k = 0, 1, \ldots, n_{\text{near}}$). Find the closest contour pixel and calculate the distance $d_k$. The contour energy is the average of the sum of all $d_k$:

$$E_c = \frac{\sum_{k=0}^{n_{\text{near}}-1} d_k}{\sigma \cdot n_{\text{near}}}$$  

where $n_{\text{near}}$ is number of neighbouring cameras, and $\sigma$ is used to normalise the range of all $d_k$ from 0 to $\sigma$.

Each contour distance image can be made in advance for reducing the heavy calculation time. Figure 6 illustrates the process of distance transforms (Borgefors, 1986):

- a the input contour image
- b setting the distance measurement mask which is in the Euclidean space
- c the result image.

In the image, brighter pixels indicate closer points, and vice versa.

![Figure 6 Distance transform](image-url)
2.2.3 Smoothness energy

So far, we have used the contour and colour information of the 3D model for estimation. However, the 3D model can be seen as a deformable model because the shape may be changed when the 3D points are moving. The surfaces of a deformable model are like membranes that pull each other, so we can reduce the probability of local minimum of photo-consistency and smooth surface, which is termed ‘smoothness’. Because the local minimum also happens under occlusion, reflection or other unexpected situations, preventing it to avoid abrupt colouring area for image synthesis is required.

For a given image containing 3D model projection, it has the property of smooth outline points. Like active contours, the outline points pull each other without non-outline points according to the tension and curvature. Because the smooth energy should be normalised, the angle proportion will be used instead of the length. Figure 7 illustrates how an outline point is calculated for the smoothness energy. \( V_i \) (for \( i = 0, 1, \cdots, n_s, n_s \) is the number of neighbouring novel view outline points) is a vector from the outline point to its neighbour, but not includes the non-outline points. \( V_m \) is the normalised moving direction vector we’ve mentioned previously. The energy is then calculated as the average of the sum of \( V_m \) and all \( V_i \) as

\[
E_s = \frac{1}{n_s} \sum_{i=0}^{n_s} \| V_i \cdot V_m \| \tag{5}
\]

where \( V_m \) and \( V_i \) are the normalised \( V_m \) and \( V_i \), respectively. We take the absolute value of the result because the smoothness energy should be large whenever protrusion or concavity occur and the smallest when it is vertical.

Figure 7 Smoothness of outline point (see online version for colours)

2.3 Refine non-outline points

Our system uses visual hull for the initial model, but the concavity of the 3D model cannot be reconstructed based on the visual hull. To solve this problem, Starck et al. (2005) use photo-consistency energy \( E_s \) to find the correct position of the 3D point, and add the visual hull energy \( E_{vh} \) and the smoothness energy \( E_s \) to avoid local minimum. The energy function for moving the non-outline point is given by:

\[
E = \beta E_s + (1 - \beta) E_{vh} + \alpha E_s \tag{6}
\]

where the weight \( \beta \) is according to reliability of photo-consistency, and the weight \( \alpha \) is assigned by the user. The procedure is similar to Section 2.2 but the details are different because of the non-outline points.

For some 3D model refinement algorithms like model deformations, they move the 3D points along the point normal or mesh normal. But some problems might happen if a point passes through another mesh. It will cause the meshes to cross each other and result in self-occlusion. This generally introduces more regions of discontinuous colour when performing the image synthesis. Since we have the viewing direction of the novel view for image synthesis, it is possible to let the non-outline points move along the line passing through itself and the centre of the novel view. This will prevent mesh cross problem and improve the quality of image synthesis at the same time. Besides, we restrict the moving direction to be the direction vector from the centre of the novel view to the non-outline point so that the concavity can be reconstructed.

2.3.1 Photo-consistency energy

First, we choose the key image according to the normal vector of the non-outline point, and its neighbours are assigned as secondary images. To reduce the effect of calibration error, the searching range is set in the secondary images. Each pixel will be the centre of an estimated window, and then we calculate the sum of the square difference of each estimated window and the key estimated window in the key image. The smallest one are denoted as \( W_k (k = 0, 1, \cdots, n_{sub}, n_{sub} \) is the secondary image number) in each secondary image and their centres are \( P_k (k = 0, 1, \cdots, n_{sub}) \). Find the 3D points \( P_{sub,k} \) which cross the ray of \( P_{key} \) and each \( P_k \) using the triangulation algorithm. The photo-consistency energy is the sum of difference of the non-outline point \( P \) and \( P_{sub,k} \):

\[
\beta E_s = \frac{\sum_{k=0}^{n_{sub}-1} w_k \| P_{sub,k} - P \|}{n_{sub}} \tag{7}
\]

and \( \beta \) means the reliability of this energy:

\[
\beta = \frac{\sum_{k=0}^{n_{sub}-1} w_k}{n_{sub}} \tag{8}
\]

2.3.2 Visual hull energy

The visual hull is reconstructed by the contours of each view image, so it can be an energy when moving the non-outline points. Specially, for some situation like self-occlusion, \( \beta \) is small and the weight of visual hull energy becomes higher instead. The energy is the distance between the non-outline point \( P \) and its closest point \( P_{vh} \) in the visual hull:

\[
E_{vh} = \| P_{vh} - P \| \tag{9}
\]
2.3.3 Smoothness energy

As mentioned in Section 2.2.3, we also adopt the smoothness energy to avoid the local minimum. Generally speaking, the smoothness is the sum of vectors \( \vec{r}_i \) from \( P \) to its neighbouring point. However, in this work, the moving direction is restricted by the line \( L \) which passes through \( P \) and \( O \) (Figure 8). The pulling force between \( \vec{P} \) and \( \vec{P}_{i} \) should also have the same property. The vector \( \vec{P} - \vec{P}_{i} \) is projected to \( L \) with the direction. Thus, the smoothness energy is given by:

\[
E_s = \sum_{i=0}^{n-1} \left( \vec{P} - \vec{P}_{i} \right) \cdot \vec{V}
\]

(10)

where \( \vec{V} \) is a unit vector.

**Figure 8** Pulling force illustration (see online version for colours)

3 View-dependent image synthesis

Assuming the novel view camera parameters are known, then every pixel in this camera is fixed in the world coordinate system. As showing in Figure 9, similar to the ray tracing, the ray which passes through the camera centre and the pixel \( P_{ij} \) must intersect the 3D model. There might be one or more intersection points but the closest one is the point \( V \) seen by the pixel \( P_{ij} \). Project \( V \) to the reference cameras and capture the colour information, and the colour of \( P_{ij} \) is the sum of these colour values multiplied by \( W_c \) respectively, where \( W_c \) is the value for which the novel viewing direction dot the referential viewing direction.

The above process is generally very slow because every ray must check with every mesh to see whether an intersection is found. In this work, we create a mesh lookup table to meet the closest intersection point. Because the 3D model is available, the depth map of the novel view can be easily derived. Each mesh is projected to the novel view and the mesh index is written in the table if the depth of the mesh and the map is equal. In the final result (Figure 10), the intersection points are found easily using the mesh lookup table.

**Figure 9** View-dependent image synthesis (see online version for colours)

**Figure 10** Mesh lookup table (see online version for colours)

As mentioned in Verlani et al. (2006), only two to three neighbouring cameras are needed when blending the images because the possibility of self-occlusion rises if the camera is far away, and the synthetic image becomes more blurred. Hence, we set \( W_c \) to the power of \( m \).

4 Experiment result

We use the dinosaur dataset for our experiment. This dinosaur model is made by plaster with no texture or colour, but there is a little light on the model when performing the image acquisition. A CCD camera is used to capture the images from different viewpoint and 12 of them are chosen as our system input. The image resolution is 640 × 480. A calibration board was set and captured by these 12 viewpoints, so we can calibrate these input cameras to get their intrinsic and extrinsic parameters using Jean-Yves Bouguet’s MATLAB toolbox. The novel view camera is set between camera 1 and camera 2 as shown in Figure 11. Note that the novel view image in this figure is the image from the dataset for verification, not the input or the result of image synthesis. In this case, we set \( \alpha \) and \( \beta \) as 0.2 for the outline points refinement, and set \( \alpha \) is 0.1 for the non-outline points refinement.
We first show the image synthesis results in each stage. As shown in Figure 12, (d) is the ground truth. The 3D model in (a) is the visual hull, and it can be seen clearly that a piece of dorsal fin is cut by the shape of silhouettes. The lost parts are fixed after moving the outline points iteratively in (b). On the other hand, the leg becomes thinner because of low sensitivity of the concavity in (a) visual hull. Thus, we move the non-outline points to solve this problem in (c), which is the final result of our system.

![Figure 12](image-url) Image synthesis in each stage, (a) visual hull (b) after refine outline points (c) after refine non-outline points (d) ground truth (final result) (see online version for colours)

We then evaluate the image colour error for every outline point refinement or the non-outline parts. We calculate the root mean square (RMS) of the difference between the result and the ground truth in the foreground. The evaluation is shown in Table 1 and Table 2. We can see that the RMS errors are decreasing after every refinement in both stages. Note that the RMS error slightly increases after the 7th refinement as shown in Table 1. This is because the smoothness energy is large enough to pull other points away from the exact position.

In this experiment, the total execution time is 298 seconds running on a notebook computer equipped with AMD Turion 1.8 GHz processor and 2 GB RAM. The 3D model reconstructed from the same dataset using Goesele’s approach (Goesele et al., 2006) takes about 12 hours and 24 minutes. The major difference between our technique and the algorithm for accurate 3D model reconstruction is that only high quality novel view images are produced, and not necessary the precise 3D model.

<table>
<thead>
<tr>
<th>Times</th>
<th>RMS error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33.57</td>
</tr>
<tr>
<td>2</td>
<td>31.98</td>
</tr>
<tr>
<td>3</td>
<td>31.27</td>
</tr>
<tr>
<td>4</td>
<td>30.92</td>
</tr>
<tr>
<td>5</td>
<td>30.57</td>
</tr>
<tr>
<td>6</td>
<td>30.24</td>
</tr>
<tr>
<td>7</td>
<td>30.26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Times</th>
<th>RMS error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29.12</td>
</tr>
<tr>
<td>2</td>
<td>28.97</td>
</tr>
<tr>
<td>3</td>
<td>28.90</td>
</tr>
<tr>
<td>4</td>
<td>28.90</td>
</tr>
<tr>
<td>5</td>
<td>28.82</td>
</tr>
</tbody>
</table>

We also perform the experiment using the 3D model ‘bunny’. This model was created without texture information, so we map woody texture on it using VTK (visualisation toolkit) library. The virtual cameras are used to capture images as the input of our algorithm. In this...
experiment, we set 12 cameras around the 3D model with 30 degrees separation between two consecutive views. The parameters $\alpha$ and $\beta$ are set as 0.3 for the outline points refinement, and $\alpha$ is set as 0.3 for the non-outline points refinement.

The novel view camera is set at a location between camera 0 and camera 1, aiming at the origin of the world coordinate system as shown in Figure 13. Note that this novel view image is only used to compare with the final result, not the input. Figures 14(a) to 14(c) show the results in each stage, and Figure 14(d) is the ground-truth image. We can see that the ear of the bunny is recovered after the refinement of outline points as shown in Figure 14(b). The concavity between the front and rear foot becomes more clear-cut as shown in Figure 14(c).

The RMS errors calculated for the Bunny object are shown in Tables 3 and 4. We can see that the RMS errors decrease after the refinement of outline points. Note that the RMS error is large at the beginning of the non-outline points refinement mainly due to the textureless woody skin. The total execution for this object is about 271 seconds.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>RMS error for every outline points refinement of the bunny object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times</td>
<td>1    2    3    4    5</td>
</tr>
<tr>
<td>RMS error</td>
<td>14.33 12.79 12.86 12.81 12.81</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>RMS error for every non-outline points refinement of the bunny object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times</td>
<td>1    2    3    4    5</td>
</tr>
<tr>
<td>RMS error</td>
<td>13.51 13.23 13.13 13.06 13.04</td>
</tr>
</tbody>
</table>

5 Conclusions

This paper introduces a novel view image synthesis system based on moving two different types of 3D points. With some information of the novel view direction, photo-consistency, image contour and surface smoothness, we can refine the 3D model and improve image quality. The contributions of this work include the followings:

- We introduce a concept that the 3D model refinement is used to improve quality of image synthesis.
- The 3D points are classified as the outline points or the non-outline points by novel view. The former corrects the shape of model and the latter improves the colour accuracy in the synthesised image.
- Design suitable energy functions according to two different types of 3D points, which consist of photo-consistency, contour/visual hull and surface smoothness.

**References**


