Depth-Map Merging for Multi-View Stereo with High Resolution Images

Shuhan Shen
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
shshen@nlpr.ia.ac.cn

Abstract

By taking advantage of recent progress in patch match based stereo [1], we propose a depth-map merging based Multi-View Stereo (MVS) reconstruction method for large-scale scenes. Combining the strength of the accuracy of patch match stereo and the redundancy information provided by multiple views, our method can produce quite satisfactory MVS reconstruction results. Besides, the proposed method could be easily parallelized at image level, i.e., each depth-map is computed individually, which makes it suitable for large-scale scene reconstruction with high resolution images.

1. Introduction

Multi-View Stereo (MVS) reconstruction of objects and scenes receives more and more interests nowadays. Although great efforts have been made in this area and some efficient algorithms have been proposed, it is still valuable to investigate accurate and efficient methods to deal with large-scale scenes using high resolution images (6 Megapixel and above) that can be readily acquired by modern digital cameras.

According to [9], MVS algorithms can be divided into four classes, called voxel based methods [13][10], surface evolution based methods [4][3], depth-map merging based methods [6][2][8][12], and feature point growing based methods [7][5]. Voxel based methods require a bounding box of the object, and its accuracy is limited by the resolution of the voxel grid, which makes it only suited for compact objects within a tight enclosing box. Although some adaptive volume subdivision methods, such as [10], are proposed to reduce computational and memory costs, they are difficult to be used for large-scale scenes. Surface evolution based methods iteratively evolve an initial guess to improve the photo consistency measurement. The key of these methods is a close and reliable initial guess which is difficult to obtain for large-scale scenes. Depth-map merging based methods are more flexible and suited for large-scale reconstruction. Such methods first compute individual depth maps and then merge them together into a single model by taking visibility into account. Feature point growing based methods first reconstruct points in textured regions, and then expand these points to untextured ones. A typical method of this kind is [5], which is now considered as the state-of-the-art MVS method.

In this paper we extend the work of patch match based stereo [1] to multi-views and propose a depth-map merging based method for large-scale scenes with high resolution images. The key of our method is an efficient patch based stereo matching process plus a depth-map merging step that enforces consistency over multiple views.

2. Algorithm Description

The proposed method consists of three steps: stereo pair selection, depth-map computation, and depth-map merging. We describe the details of each step in this section.

2.1. Stereo Pair Selection

For each image in the image set, we need to select a reference image for stereo computation. The selection of stereo pair is important for the accuracy of the stereo matching and also the final MVS results.

We use a method that is similar to [8] to select eligible stereo pairs. Suppose we have $n$ images, and for the $i$-th one, we compute $\theta_{ij}, j = 1, ..., n$ which is the angle between principal view directions of image $i$ and $j$, and compute $d_{ij}, j = 1, ..., n$ which is the distance between optical centers of image $i$ and $j$. Then for images satisfy $5^\circ < \theta_{ij} < 60^\circ$, we compute the median
$d$ of their $d_{ij}$, and remove images whose $d_{ij} > 2d$ or $d_{ij} < 0.05d$. After these computations, the remaining images are considered as the neighboring images of image $i$, denoted as $N(i)$, and the one with minimal $\theta_{ij}d_{ij}$ among $N(i)$ is selected as image $i$’s reference image to form a stereo pair.

2.2. Depth-Map Computation

For each eligible stereo pair, we follow the idea in [1] to compute the depth-map. The core idea is to find a good support plane at each pixel in the target image that has minimal aggregated matching cost, as shown in Fig.1.

![Figure 1. For each pixel $p$ in the target image, we estimate its corresponding 3D plane. The blue curve is the object surface, $C_i$ and $C_j$ are the camera centers of the target and reference images respectively, $f_1$, $f_2$ and $f_3$ are three 3D planes in $p$’s viewing ray. Obviously $f_2$ has the minimal aggregated matching cost.](image)

In our work each plane in 3D space is represented by a 3D point $X_i$ and its normal $n_i$ in camera $C_i$’s coordinate, where $C_i$ is the camera center of the target image, and $C_i - xyz$ is the camera’s coordinate.

![Figure 2. The 3D plane is represented by a 3D point $X_i$ and its normal $n_i$ in camera $C_i$’s coordinate, where $C_i$ is the camera center of the target image, and $C_i - xyz$ is the camera’s coordinate.](image)
core between \( q \) and \( H_{ij}(q) \), as:

\[
m(p, f_p) = 1 - \frac{\sum_{q \in W} (q - \bar{q})(H_{ij}(q) - \bar{H}_{ij}(q))}{\sqrt{\sum_{q \in W} (q - \bar{q})^2 \sum_{q \in W} (H_{ij}(q) - \bar{H}_{ij}(q))^2}}
\]

(5)

After the initialization, each pixel in the target image \( I_t \) corresponds to a random 3D plane. Then we process pixels in \( I_t \) one by one to refine the planes in three iterations. At the first iteration, we start from the top-left pixel and traverse in row wise order until we reach the bottom-right pixel. At the second iteration, we reverse the order to visit the pixel from the bottom-right to the top-left pixel, also in row wise order. The third iteration is exactly the same as the first one.

At each iteration, each pixel has two operations, called spatial propagation and plane refinement. Spatial propagation is used to compare and propagate the planes of neighboring pixels to that of the current pixel. In the first and third iterations, the neighboring pixels are the left and upper neighbors, and in the second iteration are the right and lower neighbors. Let \( p_{PN} \) denotes the neighbor of current pixel \( p \), and \( f_{PN} \) denotes \( p_{PN} \)'s plane, we check the condition \( m(p, f_{PN}) < m(p, f_p) \). If this condition is satisfied, we propagate \( f_{PN} \) to the current plane, i.e. set \( f_p = f_{PN} \). This spatial propagation process relies on the fact that neighboring planes are very likely to have similar 3D planes especially for high resolution images.

For each pixel \( p \), after spatial propagation, we further refine the plane \( f_p \) using random assignment. Given a range \( \{ \Delta \lambda, \Delta \theta, \Delta \phi \} \), we 1) select a random plane parameter \( \{ \lambda', \theta', \phi' \} \) in the range \( \lambda' \in [\lambda - \Delta \lambda, \lambda + \Delta \lambda], \theta' \in [\theta - \Delta \theta, \theta + \Delta \theta], \phi' \in [\phi - \Delta \phi, \phi + \Delta \phi] \). 2) Compute the new plane \( f_p' = \{ X', n' \} \) using Eq.2 and Eq.3. 3) If \( m(p, f_{p'}) < m(p, f_p) \), we accept \( f_p = f_p' \). 4) We halve the range \( \{ \Delta \lambda, \Delta \theta, \Delta \phi \} \). 5) Go back to step one. The above refinement process is repeated for 10 times. In this paper, we set \( \Delta \lambda = \frac{\lambda_{max} - \lambda_{min}}{4}, \Delta \theta = 15^\circ, \Delta \phi = 90^\circ \).

We would note that this depth-map computation process has four main differences compared with [1]. Firstly, the plane is defined in the image coordinate in [1] but in the camera’s coordinate in our work because the stereo pair is not rectified in this paper. Secondly, the aggregated matching cost \( m(p, f_p) \) in our work is a simple normalized cross correlation, not the more complex adaptive support weight version in [1], because the high resolution images can provide more reliable matches than lower resolution ones and a simple normalized cross correlation is reliable enough to measure the photometric consistency. Thirdly, we only use the spatial propagation compared to spatial and view propagation in [1] because we only compute the depth-map on \( I_i \), not on both \( I_i \) and \( I_j \) as in [1]. Finally, the method in [1] contains another post-processing step that applies occlusion treatment via left/right consistency checking and fills invalidated pixels as well, our method does not have this process since the following depth-map merging step could reach the similar effects.

### 2.3. Depth-Map Merging

Once the depth-maps have been computed, they are merged together to get a complete 3D model, and our merging process is similar with [12]. For each point \( p \) in image \( I_i \), if its matching cost \( m(p, f_p) \) is smaller than 0.2, we back project it to 3D using its depth \( \lambda \) and the camera parameters, as:

\[
X = \lambda R_i^T K_i^{-1} p + C_i
\]

where \( p \) is the homogeneous coordinate denoted in Eq.1, \( X \) is the 3D point in world coordinate. Then we project \( X \) to \( I_i \)'s neighboring images \( N(i) \). Suppose \( I_j \) is the \( j \)-th neighboring image in \( N(i) \), we define \( d(X, j) \) by the depth of \( X \) with respect to camera \( j \) and define \( \lambda(X, j) \) by the depth value computed at the projection of \( X \) in \( I_j \) using \( I_j \)'s depth-map. If the matching cost of the projection of \( X \) in \( I_j \) is smaller than 0.2 and \( \frac{d(X, j) - \lambda(X, j)}{\lambda(X, j)} < 0.01 \), we say \( X \) is consistent in \( I_i \) and \( I_j \). \( X \) is retained if it is consistent for at least 2 neighboring images in \( N(i) \).

Finally, all the retained 3D points form the final point cloud to represent the scene.

### 3. Experimental Results

We test our method on the large-scale data set provided in [11] with \( 3072 \times 2048 \) resolution (6 Megapixel) images. Two image sequences, Fountain-P11 and Herz-Jesu-P8 which has 11 and 8 images respectively, are used here. The square window for aggregated matching is set to \( 35 \times 35 \) pixels. Fig.3 shows some reconstruction results using our method. To measure the quality of the results, we project the reconstructed 3D points and the ground-truth onto images, and compute the depth errors on each image. A reconstructed 3D point is considered to be correct if the smallest ratio of its depth error to the ground-truth depth across all images is below a certain threshold. Fig.4 shows the number of correct reconstructed points as a function of the allowed error threshold, using the methods in this paper, in [12], and in [5] respectively. In Fig.4 we can see that the proposed method can generate more dense point clouds than others.
4. Conclusions

In this paper we propose a depth-map merging based MVS reconstruction method for large-scale scenes by taking advantage of recent progress in patch match based stereo [1]. Combining the strength of the accuracy of patch match stereo and the redundancy information provided by multiple views, our method can produce quite satisfactory MVS reconstruction results. Besides, the proposed method could be easily parallelized at image level, i.e., each depth-map is computed individually, which makes it suitable for large-scale scene reconstruction with high resolution images.

References