GENERATING DENSE DEPTH MAPS USING A PATCH CLOUD AND LOCAL PLANAR SURFACE MODELS

Daniel Herrera C., Juho Kannala, Janne Heikkilä

Machine Vision Group, University of Oulu, Finland

ABSTRACT

Patch cloud based multi-view stereo methods have proven to be an accurate and scalable approach for scene reconstruction. Their applicability, however, is limited due to the semi-dense nature of their reconstruction. We propose a method to generate a dense depth map from a patch cloud by assuming a planar surface model for non-reconstructed areas. We use local evidence to estimate the best fitting plane around missing areas. We then apply a graph cut optimization to select the best plane for each pixel. We demonstrate our approach with a challenging scene containing planar and non-planar surfaces.

Index Terms— Computer vision, Stereo image processing

1. INTRODUCTION

Video-plus-depth has become a leading standard for the representation and transmission of scenes in 3DTV [1]. As a transmission format it lends itself to high compression rates due to the correlation between the color and depth images. Moreover, it is the format used for depth image-based rendering, which has promising applications for the 3DTV industry, such as free viewpoint rendering.

However, not all 3D reconstruction methods produce fully dense depth maps. Textureless regions are often not reconstructed due to the lack of photometric constraints. Furthermore, using individual depth maps to reconstruct the scene can lead to inconsistencies across the different depth maps. This is a concern for multi-view stereo methods which use several cameras simultaneously to estimate the scene structure.

One common approach of state-of-the-art methods is to reconstruct the scene using primitives in a global reference frame (e.g. patches in [2]). This ensures scene consistency across all the views. However, when the scene is reprojected onto the source cameras, not all pixels are assigned a depth, mainly due to missing surfaces, resulting in semi-dense depth maps. Even more recent works that improve Furkawa and Ponce's algorithm (e.g. [3]) are not able to produce dense reconstructions. We propose an algorithm to estimate the depth of the remaining pixels after the initial 3D reconstruction assuming a planar surface model for the textureless surfaces.

978-1-61284-162-5/11/$26.00 © 2011 IEEE

2. PREVIOUS WORK

Recently, several multi-view stereo methods have emerged which assume that the scene has piecewise planar structure [4, 5, 6]. The strongest assumption is that the world has a *Manhattan* structure [4]. This means that only three perpendicular plane directions are allowed. Gallup et al. [7] use the Manhattan world assumption to extend the stereo plane-sweeping approach. They estimate the world reference frame rotation and then make a plane sweep along the three different directions. The results show an improvement over the traditional plane-sweep algorithm but is limited by the assumption of orthogonal plane directions. Moreover, the computation of individual depth maps does not guarantee scene consistency.

Furukawa et al. propose an another approach based on the Manhattan world assumption [4]. They begin with a multi-view consistent reconstruction of textured surfaces using a state-of-the-art method [2]. They then extract the dominant plane directions from the patch cloud and finally construct a depth map by labeling the pixels with a set of axis-aligned planes. They show very good results for architectural scenes but their approach is not designed to handle non-planar surfaces.

Sinha et al. enforce planar constraints on the scene surface but allow for arbitrary plane directions [5]. They first recover reliable points and 3D line segments using multi-view methods, and use them to create a set of plane hypothesis. Evidence is collected for the planes and a final depth map is generated by labeling the pixels with the plane labels. Although their approach is flexible in the plane direction, they do not take non-planar surfaces into consideration.

Finally, Gallup et al. recently presented a method that uses planar models but considers non-planar surfaces on the scene [6]. They use traditional multi-view stereo methods to generate a dense depth map for the scene. Plane hypothesis are generated using RANSAC and the 3D points. A labeling stage assigns pixels to planes and includes a *non-planar* label. To improve efficiency, a classifier is applied to detect non-planar surfaces incorrectly assigned as planar. An important difference between this method and our own is that we only com-

Algorithm 1 Plane hole filling algorithm

```
Project patches to image
plane_list ← ∅
for all hole ∈ depth_map do
    local_plane_list ← ∅
    for all patch ∈ boundary(hole) do
        if patch matches plane in local_plane_list then
            Update matching plane with patch
        else
            Add patch to plane_list
        end if
    end for
    for all plane ∈ local_plane_list do
        Refine parameters
        Calculate global inliers
    end for
    Merge local_plane_list with plane_list
end for
Select best planes
Compute cost function
Graph cut labeling
Fill depth map
```

pute depth values for missing pixels in the depth map and existing depth values are not replaced. This eliminates the possibility of reducing the quality of the depth map by incorrectly assigning a region as planar. Moreover, we use a local approach to detect the plane parameters around each missing region which ensures that the estimated plane parameters are coherent with the local surface.

3. DENSE DEPTH MAPS FROM PATCH CLOUDS

The algorithm takes as input a set of images, their corresponding projection matrices, and a semi-dense 3D patch cloud. The projection matrices can be obtained through any of the existing structure-from-motion algorithms. The patch cloud is produced by a semi-dense multi-view stereo method like Furukawa and Ponce's [2]. It is worth noting that Furukawa and Ponce's multi-view stereo method is publicly available and has demonstrated its applicability to large scale scenes as well as for synchronized multi-view video. Each patch produced by this method consists of a 3D position, a normal direction, and the indices of images that observe the patch. The algorithm produces a dense depth maps for each image suitable for transmission and processing (e.g. free viewpoint video). Our method is summarized in Algorithm 1.

3.1. Initial depth map

An initial depth map is generated for an image by projecting all visible patches onto the camera's image plane and taking the depth of the nearest patch. That is, the center of each patch is projected and a 3x3 pixel neighborhood is set to the patch's depth if no other patch is closer. This generates a semi-dense depth map with holes where no patches could be reconstructed. These are usually regions of very low texture that cannot be reliably matched. An example of a depth map generated in this way is shown in figure 1b.

A list is constructed that indicates which patch generated the depth for each pixel so that we refer to a patch by its 2D pixel location in the image.

3.2. Plane hypotheses

The algorithm assumes that textureless surfaces can be approximated by a plane. It is expected that some patches will be reconstructed on the plane from areas of moderate texture or the intersection of the plane with other surfaces. Since each reconstructed patch defines a plane we have as many plane hypotheses as there are patches. However, testing all planes would be prohibitevely expensive, thus we seek to refine the list of hypotheses to the most likely ones. For this we first use a local approach to find the plane parameters and then a global measure to rank the estimated planes.

Pixels with no depth information are grouped into connected regions called holes. Each hole is processed individually to produce a list of plane hypotheses. It is expected that at least some part of the hole boundary will contain reconstructed patches lying on the plane. Thus, the patches on the boundary of the hole are used to estimate the plane parameters. A patch is considered on the boundary of the hole if it projects to a pixel on the boundary.

Because many patches describe very similar planes, the plane hypotheses are clustered to eliminate redundant planes. Planes are merged in the same cluster if the angle between their normals and the difference between their distances from the origin are smaller than a given threshold. The members' parameters are averaged to obtain the cluster's normal and distance from the origin.

3.3. Local plane refinement

The parameters of the patches produced by the multi-view stereo stage are inherently noisy and can be refined by considering neighboring patches. We wish to leverage this spatial information in the local neighborhood of the hole because those patches are more likely to belong to the plane we are seeking. The refinement stage first finds inlier patches for the initial plane through region growing around the hole boundary and then performs least-squares plane fitting in order to extract the refined plane parameters.

We use only the patch position, because the normal is considerably noisier. A patch with position p is an inlier for a plane if the distance to the plane is below a threshold δ_p:

$$|n^T p - d| \leq \delta_p \tag{1}$$

where n is the unit normal of the plane, d is the distance from the origin, and δ_p accounts for noise.

To leverage local information a region growing approach to inlier selection is used. The hole boundary patches that satisfy equation (1) are used as seeds. Pathces on the boundary of the region are iteratively tested and added to the region if they satisfy equation (1). In this manner the region contains only pixels that are connected to the hole. Once locally connected inliers are extracted, a plane is fitted to their 3D locations using least-squares.

3.4. Global plane ranking

Once the plane parameters have been refined, all patches visible in the image are tested against equation (1). A smaller δ_p is used because we now have more accurate plane parameters. The number of patches that satisfy equation (1) indicates how much evidence there is for that plane. Testing all patches allows us to take into account evidence for the plane that may not be spatially connected to the hole (e.g. due to occlusions or obstacles).

The local plane lists for the holes are merged into a single list. Planes estimated from different holes that have similar parameters are removed and only the one with a higher global inlier count is kept. Depending on the complexity of the scene and the available processing power, the final list of planes can be too big for timely processing. The planes can be sorted by their global inlier counts and only the N first are kept, where N is the number of planes that we can process in later stages. This ensures that processing power is spent only on the most relevant plane hypotheses.

3.5. Photometric consistency

A per-pixel cost function is defined for each plane that models the photometric consistency of the images. A pixel in the reference image p has a cost defined for each plane hypothesis h and each image j in the dataset. A homography H_j^h is used to find the corresponding coordinates in image j. Once corresponding coordinates are computed the colors are compared.

To compute the cost function over a set of images the median over the absolute color differences is used.

$$C^h(p) = Med_j(||I_j(H_j^h p) - I_{ref}(p)||) \tag{2}$$

Per-pixel cost functions are considerably affected by occlusions. Gallup et al. [7] make the observation that if a point is occluded when the camera moves in a given direction, the point is usually not occluded when the camera moves in the opposite direction. We incorporate this knowledge by performing principal component analysis on the camera centers to estimate the dominant translation direction. The image set is then divided into two groups: before and after the reference image along the dominant direction (equivalent to the left and right groups in [7]). The final cost is the minimum of the two groups for each pixel.

3.6. Pixel labeling

We assign a plane label to each pixel using a graph cut minimization [8]. The energy function contains a per pixel data term $E_d(h_p)$ and a pairwise smoothness term $E_s(h_p, h_q)$:

$$E = \sum_p E_d(h_p) + \lambda \sum_{p,q \in N(p)} E_s(h_p, h_q) \tag{3}$$

where h_p is the plane hypothesis assigned to pixel p and $N(p)$ denotes the 4 connected neighborhood of pixel p.

The data term is the cost function from equation (2). Because the labels correspond to planes and have no specific ordering or numeric value, the Potts model is used for the smoothness term. A uniform cost is assigned if the labels are different:

$$E_d(h_p) = C_h(p) \tag{4}$$

$$E_s(h_p, h_q) = \begin{cases} 0 & h_p = h_q \\ 1 & \text{else} \end{cases} \tag{5}$$

Once each pixel has been labeled with a plane hypothesis, those pixels that were missing depth information can now be assigned a depth by finding the intersection of the optical ray with the plane.

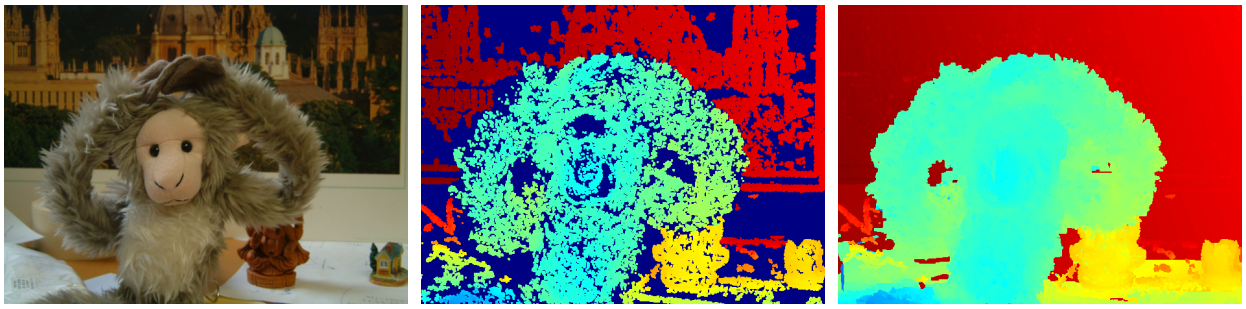
4. EXPERIMENTAL RESULTS

The proposed algorithm was tested using the publicly available monkey sequence.[1] This is a set of 89 images taken with a handheld camera of a teddy monkey in front of a planar background. Figure 1a shows a source frame from the sequence.

The publicly available PMVS2 software [2] was applied to the dataset to extract the patch cloud. The initial depth map obtained from the patch cloud is shown on figure 1b. After applying the hole filling algorithm the two planar surfaces present in the image were succesfully detected and assigned perpendicular normals. The final dense depth map is presented in figure 1c. The holes were assigned depth values that are coherent with the scene structure and the initial depth map. Note how the white marquee of the background on the lower right was assigned the proper depth even though it has no visible texture. Even regions that are not planar but have a continuous surface, like the monkey were completed with acceptable depth values.

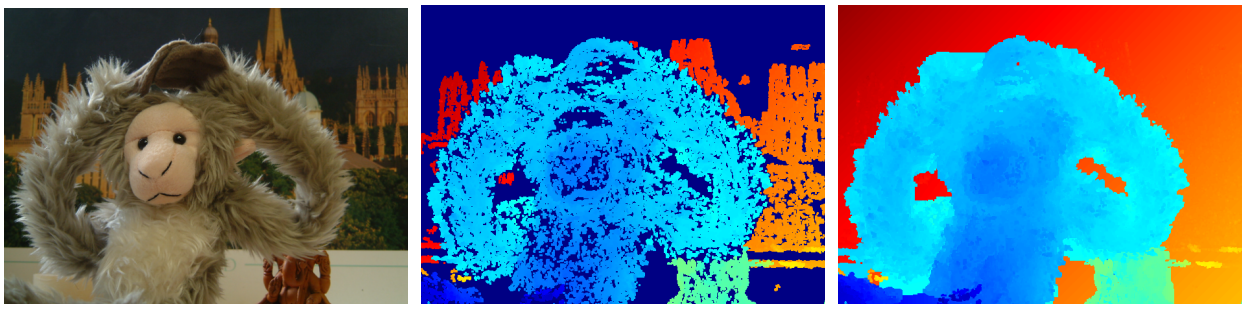
Figure 2 shows the results for another frame. In this case only the background plane is visible and the textureless area covers a much bigger region of the image. The background plane was correctly recovered. Almost all patches behind the monkey were filtered out by the visibility analysis of the PMVS software. Only one spurious background patch was marked as visible through the missing regions in the top of the monkey.

[1] http://www.robots.ox.ac.uk/~awf/ibr/



(a) Source image (b) Depth map after patch projection (c) Depth map after hole filling

Fig. 1: Results for an image with two planes.



(a) Source image (b) Depth map after patch projection (c) Depth map after hole filling

Fig. 2: Results for an image with one plane.

5. CONCLUSIONS

The proposed algorithm is capable of using the recovered 3D information from textured regions to estimate the depth of non-textured surfaces. It uses a piecewise planar sufrace model to complete the semi-dense depth map produced by state-of-the-art multi-view stereo methods. It can handle multiple planes in the scene and uses a local region growing approach to find the best fitting plane around depth map holes. It was shown to perform adequately in the presence of planar and non-planar surfaces simultaneously.

6. REFERENCES

[1] A. Smolic, K. Müller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. A. Triantafyllidis, and A. Koz, "Coding algorithms for 3dtv - a survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1606–1621, 2007.

[2] Yasutaka Furukawa and Jean Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, 2010.

[3] Tai-Pang Wu, Sai-Kit Yeung, Jiaya Jia, and Chi-Keung Tang, "Quasi-dense 3d reconstruction using tensor-based multiview stereo," in *Computer Vision and Pattern Recognition*, 2010, pp. 1482–1489.

[4] Y. Furukawa, B. Curless, S.M. Seitz, and R. Szeliski, "Manhattan-world stereo," *Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 1422–1429, 2009.

[5] S.N. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *International Conference on Computer Vision*, 2009.

[6] D. Gallup, J.M. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1418–1425.

[7] David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *CVPR*, 2007.

[8] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2002.