Robust and Accurate Multi-View Reconstruction by Prioritized Matching

Markus Ylimäki, Juho Kannala, Jukka Holappa, Janne Heikkilä
University of Oulu

Sami S. Brandt
University of Copenhagen

Abstract

This paper proposes a prioritized matching approach for finding corresponding points in multiple calibrated images for multi-view stereo reconstruction. The approach takes a sparse set of seed matches between pairs of views as input and then propagates the seeds to neighboring regions by using a prioritized matching method which expands the most promising seeds first. The output of the method is a three-dimensional point cloud. Unlike previous correspondence growing approaches our method allows to use the best-first matching principle in the generic multi-view stereo setting with arbitrary number of input images. Our experiments show that matching the most promising seeds first provides very robust point cloud reconstructions efficiently with just a single expansion step. A comparison to the current state-of-the-art shows that our method produces reconstructions of similar quality but significantly faster.

1 Introduction

Multi-view stereo reconstruction is a classical research area in computer vision which has rapidly developed during the recent years [12, 15]. A wide variety of different approaches have been proposed. They can be roughly categorized into three main groups based on the properties of the reconstruction algorithm and the underlying scene representation method: (a) global approaches which globally optimize a cost function for shapes defined on a dense volumetric grid [13, 1, 6], (b) depth map based approaches which first generate multiple depth maps from subsets of input views and then fuse the depth maps into a single surface model [10, 16, 15], and (c) surface expansion approaches which iteratively expand a sparse set of matched interest points into a quasi-dense point cloud [8, 3, 7].

The global approaches are robust since they are not sensitive to bad initialization. However, most such global optimization approaches are not suitable for large-scale scenes because they use a volumetric voxel representation whose computational and memory costs are high [1, 6]. In contrast, surface expansion methods and depth map based approaches usually have a wider applicability and are suitable also for large scenes. In fact, impressive results have been recently obtained using both of these approaches [2, 15].

In this paper, we concentrate on surface expansion techniques for multi-view reconstruction. That is, we propose a multi-view matching method which iteratively expands a sparse set of tentative matches into a quasi-dense point cloud that represents the surfaces of the scene. Our approach is inspired by the method of [3] and it builds upon the techniques introduced in [5, 7]. However, instead of the heuristic match expansion strategy of [3], we use a justified prioritized matching approach which expands the most promising seeds first. Further, we generalize the match expansion algorithm of [7] for more than three views. In fact, to the best of our knowledge, the proposed algorithm is the first one which successfully applies prioritized matching in the generic multi-view stereo setting.

The main motivation for the prioritized matching algorithm is to improve the efficiency of match expansion methods while maintaining their robustness. Indeed, our results show that the proposed algorithm allows us to obtain accurate and outlier-free point clouds substantially faster than [3]. Further, the resulting point clouds have typically less noise and outliers than the point clouds obtained by the fast and parallel plane-sweep stereo method used in [15]. Thus, if a surface mesh model is the required output instead of a point cloud, it could be particularly useful to combine the high-quality point clouds produced by our algorithm with the robust surface mesh generation approach of [15]. This might help to avoid erroneous surface meshes caused by outliers of the point cloud [4].

2. Algorithm

Our algorithm builds upon the two-view and three-view matching methods [5, 7]. In this paper we extend the approach to the generic multi-view stereo problem.

Overview The outline of our approach is as follows. First, we obtain a set of initial seed points by detecting corresponding affine covariant regions in several pairs of input views and then reconstructing (i.e. triangulating) the regions in 3D space [11]. The pair of views from which each seed point ($\mathbf{s}$) is triangulated defines the reference views for the seed ($\mathbf{s}.a$ and $\mathbf{s}.b$). The seed points are sorted according to their matching scores ($\mathbf{s}.s$) which are obtained by combining pairwise similarity scores of local image patches computed between the first reference view ($\mathbf{s}.a$) and other views. Next, the seed points are iteratively expanded in the score order. That is, at each iteration the seed point with the highest score value is chosen as the current seed and new candidate matches are searched nearby the current seed in the reference views. Promising candidate matches with a high pairwise similarity score are triangulated and projected to other views. If the number of views for which the pairwise score exceeds a certain threshold is sufficient, the new point is accepted (i.e. added to the point cloud $\mathcal{M}$) and the pixels in corresponding views are marked reserved. Also, the pairwise scores of the new point are combined to its total score and the point is added to the priority queue of seeds. In this way, the accepted surface points are seeds too and the surfaces may expand until the queue of seeds becomes empty.

Details The details are shown in Alg. 1 and the data structure for the seed points is given in Def. 1.

In order to get the input seed points $\mathcal{S}$, one may use any method for finding corresponding affine covariant regions from view pairs [11]. Given a pair of regions in views a and b, their centroids $\mathbf{x}_a$ and $\mathbf{x}_b$ can be triangulated to get a 3D point $\mathbf{X}$. Local affine frames defined at $\mathbf{x}_a$ and $\mathbf{x}_b$ allow to estimate the surface normal $\mathbf{n}$ at $\mathbf{X}$. In addition to a, b, $\mathbf{x}_a$, $\mathbf{x}_b$, $\mathbf{X}$, and $\mathbf{n}$, each seed point contains variables s, s_{ab}, v_{ab}, and V. Here s is the total matching score (defined below), s_{ab} is the pairwise similarity score of local image patches in views a and b, v_{ab} is the minimum intensity variance of the two image patches, and V is a table of binary variables indicating the views in which the seed is visible. However, the latter four variables (s, s_{ab}, v_{ab}, V) are not needed for the initial seeds as they are computed in Alg. 1.

The total matching score $\mathbf{s}.s$ for a given seed $\mathbf{s}$ is obtained by combining several pairwise similarity scores. One first computes the pairwise score s_k between the reference view $\mathbf{s}.a$ and each view k where $\mathbf{s}$ may be visible. That is, we define similarity measure $\mathtt{sim}$,

$$[s_k, v_k] = \mathtt{sim}(\mathbf{s}, \mathcal{I}_{\mathbf{s}.a}, \mathcal{I}_k, \mathbf{P}_a, \mathbf{P}_k), \qquad (1)$$

which computes the normalized cross-correlation s_k of local patches in images $\mathcal{I}_{\mathbf{s}.a}$ and $\mathcal{I}_k$ as well as the minimum value v_k of intensity variances of the two patches. A square patch P, centered to the projection of $\mathbf{s}$, is used in $\mathcal{I}_{\mathbf{s}.a}$, and the patch in $\mathcal{I}_k$ is an affine transformed version of P, where the affine transformation is determined

Definition 1: Data structure for multi-view seed points

```
struct seedpoint{ int a, b;  int[] V;
                  double x_a, x_b, X, n, s, s_ab, v_ab; };
```

Algorithm 1: Multi-view match propagation

Input: images $\mathcal{I}_j$, camera matrices $\mathbf{P}_j$, seed points $\mathcal{S}$, thresholds ϵ_d, ϵ_e, t, z, K

Output: list of points $\mathcal{M}$, matching tables $\mathcal{J}_j$

Initialize $n=0$, $\mathcal{M}=\emptyset$, $\mathcal{J}_j(\mathbf{p})=0$ for all j, $\mathbf{p}$
for each seed point $\mathbf{s}$
  for each view k where $\mathbf{s}$ is in the field of view
    Compute pairwise similarity score s_k between view k and the reference view $\mathbf{s}.a$, i.e. $s_k = \mathtt{sim}(\mathbf{s}, \mathcal{I}_{\mathbf{s}.a}, \mathcal{I}_k, \mathbf{P}_a, \mathbf{P}_k)$
  end for
  Combine all pairwise scores s_k to get the total score $\mathbf{s}.s$
end for
Sort the seeds according to the scores $\mathbf{s}.s$
Initialize priority queue $\mathcal{Q}$ with sorted seeds
while $\mathcal{Q}$ not empty
  Draw the seed $\hat{\mathbf{q}} \in \mathcal{Q}$ with the best score $\hat{\mathbf{q}}.s$
  Set $a=\hat{\mathbf{q}}.a$ and $b=\hat{\mathbf{q}}.b$
  for each new match $\mathbf{q}^i$ nearby $\hat{\mathbf{q}}$ which satisfies the disparity gradient limit ϵ_d and the epipolar constraint ϵ_e
    Set $\mathbf{q}^i.s_{ab}=-\infty$ and $\mathbf{q}^i.V_j=0$ for all j
    if $\mathcal{J}_a(\mathtt{round}(\mathbf{q}^i.\mathbf{x}_a))=0$ **&** $\mathcal{J}_b(\mathtt{round}(\mathbf{q}^i.\mathbf{x}_b))=0$
      $[\mathbf{q}^i.s_{ab}, \mathbf{q}^i.v_{ab}] = \mathtt{sim}(\mathbf{q}^i, \mathcal{I}_a, \mathcal{I}_b, \mathbf{P}_a, \mathbf{P}_b)$
  end for
  Sort matches $\mathbf{q}^i$ according to the scores $\mathbf{q}^i.s_{ab}$
  for each $\mathbf{q}^i$ satisfying $\mathbf{q}^i.s_{ab} \geq z$ and $\mathbf{q}^i.v_{ab} \geq t$
    Set $n=n+1$ and $\mathbf{q}^i.\mathbf{n}=\hat{\mathbf{q}}.\mathbf{n}$
    Set $\mathbf{q}^i.V_j=1$ for $j=\{a,b\}$
    Triangulate, $\mathbf{q}^i.\mathbf{X}=\mathtt{triang}(\mathbf{q}^i.\mathbf{x}_a, \mathbf{q}^i.\mathbf{x}_b, \mathbf{P}_a, \mathbf{P}_b)$
    for each view k where $\mathbf{q}^i$ is in the field of view
      Project $\mathbf{x}_k = \mathbf{P}_k(\mathbf{q}^i.\mathbf{X})$, set $s_k=-\infty$
      if $\mathcal{J}_k(\mathtt{round}(\mathbf{x}_k))=0$
        $s_k = \mathtt{sim}(\mathbf{q}^i, \mathcal{I}_a, \mathcal{I}_k, \mathbf{P}_a, \mathbf{P}_k)$
      if $s_k \geq z$
        Set $\mathbf{q}^i.V_k=1$
    end for
    if $\mathtt{sum}_j(\mathbf{q}^i.V_j) \geq K$
      Combine pairwise scores s_k to get $\mathbf{q}^i.s$
      Set $\mathcal{Q}=\mathcal{Q}\cup\{\mathbf{q}^i\}$ and $\mathcal{M}=\mathcal{M}\cup\{\mathbf{q}^i\}$
      for views k such that $\mathbf{q}^i.V_k=1$
        Set $\mathcal{J}_k(\mathtt{round}(\mathbf{q}^i.\mathbf{x}_k))=n$
      end for
  end for
end while

using camera matrices $\mathbf{P}_a, \mathbf{P}_b$ and the surface position and orientation at $\mathbf{s}$. Then the pairwise scores s_k are combined to get the total score $\mathbf{s}.s$ defined by

$$\mathbf{s}.s = \sum_k \max\left(0, 1 - \frac{(s_k - 1)^2}{(z-1)^2}\right), \qquad (2)$$

where parameter $z \in [0, 1]$ acts as a threshold [7].

The prioritized match expansion algorithm first sorts the initial seeds to a priority queue $\mathcal{Q}$ according to their scores (lines 1-9 in Alg. 1). Then it starts processing seed points in the score order, adding obtained new surface points as seeds into the priority queue (lines 10-37). At each iteration, the seed with the highest score is chosen and new candidate matches are searched in its surroundings in the reference views. The search of candidate matches is similar to [5, 7]. Good candidate matches, whose similarity scores between the reference views exceed the threshold z, are triangulated and projected to other views. The new point is considered to be visible in such views where the local similarity to the primary reference view a exceeds the threshold z. If the number of views where the point is visible is greater than equal to K, the point is added to the point cloud $\mathcal{M}$ and to the priority queue $\mathcal{Q}$. The expansion process is tracked by matching tables $\mathcal{T}_j$ which have the same size as input images and in which the pixels corresponding to already reconstructed surface points contain a pointer to the respective point in the point cloud $\mathcal{M}$. The expansion continues as long as there are seeds in $\mathcal{Q}$.

3 Experiments

Evaluation of accuracy. We did experiments with five datasets: Fountain-P11 and Herz-Jesu-P8 datasets of [14], Dino and Temple sparse ring datasets of [17], and our own dataset of 17 images of a calibration object. We had ground truth triangle meshes for three datasets: Fountain-P11, Herz-Jesu-P8 and our own dataset. The meshes were projected to images to get ground truth depth maps for each camera. Then, by using the ground truth depth maps, we compared the completeness and accuracy of point cloud reconstructions obtained by our method and the method of Furukawa and Ponce [3].

In order to perform the comparison with [3], we used the PMVS program provided by Furukawa. The output of this program is a point cloud. As both [3] and our method use zero-mean normalized cross-correlation (ZNCC) as a similarity measure for image patches, we used the same values of ZNCC thresholds and patch sizes for both methods. The Middlebury College evaluation [17] was not used for Temple and Dino datasets because it requires triangle mesh models but the outputs of the compared methods are point clouds.

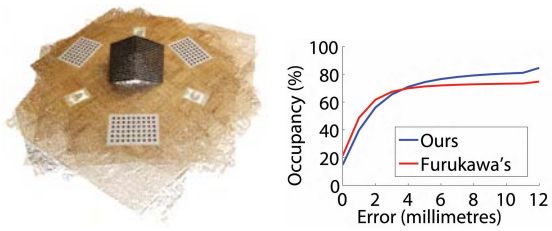

Figure 1. Our point cloud (left) and its comparison to Furukawa's (right).

Furukawa's program contains a built-in seed generation (i.e. initial feature matching) whereas for our method we extracted the seed matches by matching Hessian-Affine regions [11] using SIFT descriptors [9]. In both cases, seed matches were detected from every image pair of each dataset.

The point cloud obtained from our dataset by the proposed method is shown in Fig. 1. Furukawa's point cloud was visually similar but is not shown here due to lack of space. The comparison of point clouds is shown on the right in Fig. 1 (as in [14]), where the y-axis (i.e. the height of the curves) shows the proportion of pixels which are reconstructed and whose reconstruction error with respect to the ground truth depth map is less or equal than the corresponding value on the x-axis. The reconstructions and performance curves for Fountain-P11 and Herz-Jesu-P8 datasets are shown in Figs. 2 and 3, respectively. The bumps in the right end of the curves are due to the fact that the last point on each curve contains all pixels whose errors are greater than the limit (i.e. the highest value on the x-axis).

Overall, it can be seen that Furukawa's point clouds appear to be slightly more accurate whereas our point clouds are more dense. However, the minor differences in accuracy are not significant in practice because, in most multi-view stereo systems, point clouds are finally transformed to triangle meshes which are iteratively refined in any case [3, 15]. Hence, it can be concluded that the quality of results is approximately similar with both methods. Figures 2 and 4 allow to verify this visually.

Computational efficiency. Table 1 shows the execution times and the number of points in the point clouds. For both methods the reported values are the total exe-

Table 1. Comparison of efficiency.

Dataset	Number of points		Execution time (s)	
	[3]	Our	[3]	Our
Temple	228 053	394 090	3892	489
Dino	301 486	463 443	4548	328
Calibration Cube	398 546	681 490	3535	392
Fountain-P11	426 587	803 912	5176	529
Herz-Jesu-P8	368 323	590 637	3687	611

Figure 2. Fountain and Herz-Jesu reconstructions produced by Furukawa's method [3] (left) and our method (right).

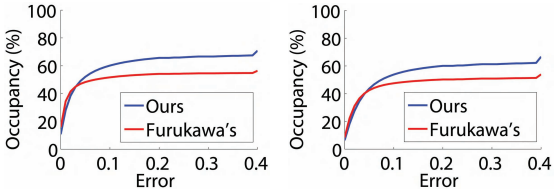

Figure 3. Comparison of Fountain (left) and Herz-Jesu (right) reconstructions.

cution times including seed extraction and expansion. However, the seed extraction stage is not optimized in our program and, hence, further speed-up could be achieved by improving its efficiency. Nevertheless, as one can see from Table 1 and Figs. 2 and 4, already the current implementation of our method produces denser point clouds than [3] and substantially faster.

4. Conclusion

In this paper, we have proposed a prioritized matching approach for multi-view stereo reconstruction. The proposed approach takes a sparse set of seed matches as input and propagates the seeds to neighboring regions. The approach allows using the best-first matching principle, where the most promising seed is propagated first, in the generic multi-view stereo setting with an arbitrary number of input images and with a single expansion step. The comparison to the current state-of-the-art showed that our method produces denser reconstructions with similar accuracy but significantly faster.

References

[1] N. D. F. Campbell et al. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV*, 2008.

[2] Y. Furukawa et al. Towards Internet-scale multi-view stereo. In *CVPR*, 2010.

[3] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *TPAMI*, 2009.

[4] M. Jancosek and T. Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *CVPR*, 2011.

[5] J. Kannala and S. S. Brandt. Quasi-dense wide baseline matching using match propagation. In *CVPR*, 2007.

[6] K. Kolev et al. Continuous global optimization in multiview 3D reconstruction. *IJCV*, 84(1):80–96, 2009.

[7] P. Koskenkorva et al. Quasi-dense wide baseline matching for three views. In *ICPR*, 2010.

[8] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *TPAMI*, 27(3):418–433, 2005.

[9] D. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 60:91–110, 2004.

[10] P. Merrell et al. Real-time visibility-based fusion of depth maps. In *ICCV*, 2007.

[11] K. Mikolajczyk et al. A comparison of affine region detectors. *IJCV*, 65:43–72, 2005.

[12] S. M. Seitz et al. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006.

[13] S. N. Sinha et al. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *ICCV*, 2007.

[14] C. Strecha et al. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008.

[15] H.-H. Vu et al. High accuracy and visibility-consistent dense multi-view stereo. *TPAMI*, 2011.

[16] C. Zach et al. A globally optimal algorithm for robust TV-L^1 range image integration. In *ICCV*, 2007.

[17] http://vision.middlebury.edu/mview/.

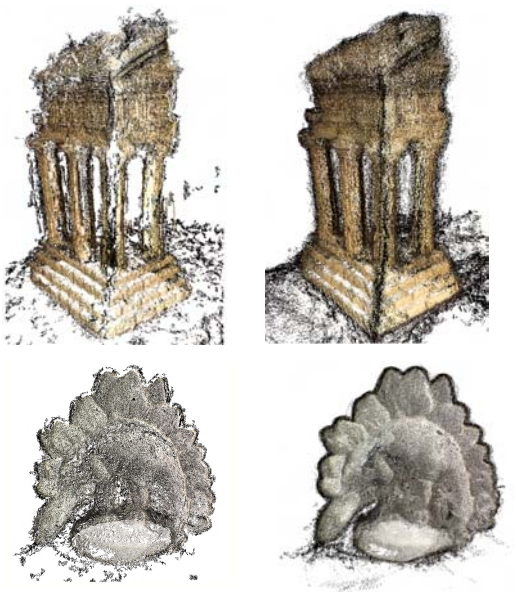

Figure 4. Temple and Dino reconstructions produced by Furukawa's method [3] (left) and our method (right).