Quasi-Dense Wide Baseline Matching for Three Views

Pekka Koskenkorva, Juho Kannala, and Sami S. Brandt
Machine Vision Group, University of Oulu, Finland
{pkoskenk,jkannala,sbrandt}@ee.oulu.fi

Abstract

This paper proposes a method for computing a quasi-dense set of matching points between three views of a scene. The method takes a sparse set of seed matches between pairs of views as input and then propagates the seeds to neighboring regions. The proposed method is based on the best-first match propagation strategy, which is here extended from two-view matching to the case of three views. The results show that utilizing the three-view constraint during the correspondence growing improves the accuracy of matching and reduces the occurrence of outliers. In particular, compared with two-view stereo, our method is more robust for repeating texture. Since the proposed approach is able to produce high quality depth maps from only three images, it could be used in multi-view stereo systems that fuse depth maps from multiple views.

1. Introduction

Multi-view reconstruction [6] is a classical, but still topical problem in computer vision [15, 4]. Much of the early research on stereo was concentrated on the small baseline case [13], whereas the more recent works have focused on matching widely separated views [8, 15]. Further, it is known that increasing the number of views can notably improve the quality of reconstruction [1], though many recent multi-view reconstruction systems are based on two-view matching [10, 14].

Many of the most successful multi-view stereo methods have been based on depth map fusion [9, 2, 14, 12]. That is, they divide the reconstruction process into two stages, where the first stage estimates depth maps from several subsets of neighboring input views and the second stage combines the different depth maps into a global surface estimate. Various methods have been proposed for both stages. For example, the individual depth maps could be computed by matching pairs of neighboring views with two-view stereo methods [14] or by using plane sweep matching for sets of several views [5, 12]. On the other hand, there are also several methods for combining multiple depth maps into a single surface estimate [14, 9, 2, 12].

In this paper, we deal with the problem of depth map estimation. We propose a three-view matching algorithm for wide baseline images which can readily replace the two-view match propagation methods used for example in [8, 15] if the trifocal tensor is known. The output of the algorithm is a set of sub-pixel matches which can be transformed to a depth map or to a quasi-dense 3D point cloud by triangulation [6].

Our approach for three-view matching is based on the best-first match propagation principle [10]. In [8], this principle was used for matching two wide baseline images and, in this paper, it is extended to the case of three views. Our work is motivated by the recent success of correspondence growing methods in two-view [8, 3] and multi-view matching [4]. The proposed implementation is build on [8] but the three-view constraint is used for scoring and sorting of seed matches during growing. Hence, our approach can be directly applied in wide baseline conditions and no rectification is needed [16]. The results show that the third view usually removes the ambiguities caused by periodic structures and, hence, greedy best-first matching, as proposed, is often sufficient. However, if robustness is preferred over efficiency, it would be straightforward to allow multiple match hypotheses per pixel as in [3]. Thereafter, the multiple hypotheses could be resolved by using some global approach as in [3] or [2].

The main difference between our method and the multi-view approach of [4], one of the top performers in [17], is the fact that [4] does not use the best-first growing strategy as we do. This strategy allows us to acquire the result with a single growth stage and to avoid the time-consuming, repeated expansion and filtering stages of [4].

2. Algorithm

Background. Given two views $\mathcal{I}_1$, $\mathcal{I}_2$ and the associated fundamental matrix, the method in [8] produces a quasi-dense set of point correspondences by growing a sparse set of seed matches, which are determined by matching affine covariant regions [8, 11].

Definition 1: Data structures for seed matches

```
struct twoviewseed{  int $a, b$;
        double $\mathbf{x}_a, \mathbf{x}_b, \mathbf{A}_{ab}, s_{ab}, v$; };
struct threeviewseed{  int $a, b, c$;
        double $\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{A}_{ab}, \mathbf{A}_{ac}, s_{ab}, s_{ac}, s, v$; };
```

Hence, the method in [8] contains two stages: the initial matching stage and the growth stage. The output of the initial matching stage is a set of seed matches $\{\mathbf{s}^i\}_i$, where each seed contains image coordinates $\mathbf{x}_a$ and $\mathbf{x}_b$, which denote the centroids of the matched regions [11], and an affine transformation matrix $\mathbf{A}_{ab}$, which approximates the local geometric transformation between the views. For each seed, the index $a \in \{1, 2\}$ indicates the reference view, and b is the other view. The reference view is determined so that the affine transformation $\mathbf{A}_{ab}$ from a to b is magnifying, i.e. $|\det \mathbf{A}_{ab}| \geq 1$ [8]. Further, each seed is associated with a texture similarity score s_{ab} and intensity variance score v. The zero-mean normalized cross-correlation (ZNCC) of geometrically normalized image patches is used as the similarity measure, and the score v is set equal to the minimum intensity variance of the two patches. The data structure for the two-view seeds is summarized in Def. 1. The same structure is used also for the grown matches.

In the growth stage, the seeds are sorted into a priority queue $\mathcal{Q}$ according to their similarity scores and then propagated by iterating the following steps:

(i) The seed $\mathbf{s}$ with the best score is removed from $\mathcal{Q}$.

(ii) New candidate matches are searched nearby $\mathbf{s}$ by using $\mathbf{s}.\mathbf{A}_{ab}$ for the normalization of local patches.

(iii) The candidates, which satisfy the disparity gradient limit and the epipolar constraint and have a sufficiently high similarity, are added to $\mathcal{Q}$ and to the list of matches after updating their affine transformation estimates. The corresponding pixels in the matching tables are marked as reserved.

The geometric normalization of patches and the update of the affine transformations are detailed in [8]. Further, for a grown match, the indices a and b are determined from the updated transformation matrix.

The result of two-view match propagation is a list of grown matches $\mathcal{M}$ and two matching tables $\mathcal{J}_1$ and $\mathcal{J}_2$, which have the same size as images $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. The nonzero values in $\mathcal{J}_1$ and $\mathcal{J}_2$ indicate the pixels which are nearby to the sub-pixel matches of $\mathcal{M}$. That is, the closest pixel to a given coordinate vector $\mathbf{x}$ is $\mathbf{p} = \texttt{round}(\mathbf{x})$, and a nonzero value $\mathcal{J}_k(\mathbf{p})$ is an index to the corresponding item in $\mathcal{M}$. The value $\mathcal{J}_k(\mathbf{p}) = 0$ indicates that pixel $\mathbf{p}$ is not matched.

Algorithm 1: Three-view match propagation

```
Input:  images $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3$, seeds $\mathcal{S}_{12}, \mathcal{S}_{13}, \mathcal{S}_{23}$,
          thresholds $z$, $t$, $\epsilon$, trifocal tensor $\mathcal{T}_1^{23}$
Output:  list of matches $\mathcal{M}$, matching tables $\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3$
1  Initialize $n = 0$, $\mathcal{M} = \emptyset$, $\mathcal{J}_k(\mathbf{p}) = 0$ for all $k$, $\mathbf{p}$
2  Compute pairwise similarity scores $\mathbf{s}.s_{ab}$ and variance
   scores $\mathbf{s}.v$ for all seeds $\mathbf{s}$, $[\mathbf{s}.s_{ab}, \mathbf{s}.v] = \texttt{sim}(\mathbf{s}, \mathcal{I}_{\mathbf{s}.a}, \mathcal{I}_{\mathbf{s}.b})$
3  Extend the two-view seeds to three views by trifocal
   transfer: $[\mathbf{s}.\mathbf{x}_c, \mathbf{s}.\mathbf{A}_{ac}] = \texttt{transfer}(\mathbf{s}, \mathcal{T}_1^{23})$
4  Compute pairwise similarity scores $\mathbf{s}.s_{ac}$ for all seeds $\mathbf{s}$
5  Combine similarity scores, $\mathbf{s}.s = \texttt{score}(\mathbf{s}.s_{ab}, \mathbf{s}.s_{ac}, z)$
6  Sort the seeds according to their total scores $\mathbf{s}.s$
7  Initialize priority queue $\mathcal{Q}$ with sorted seeds
8  while $\mathcal{Q}$ not empty
9      Draw the seed $\hat{\mathbf{q}} \in \mathcal{Q}$ with the best score $\hat{\mathbf{q}}.s$
10     Set $a = \hat{\mathbf{q}}.a$ and $b = \hat{\mathbf{q}}.b$
       (In the following, $\mathcal{I}_a$ and $\mathcal{I}_b$ define new seeds)
11     for each new match $\mathbf{q}^i$ nearby $\hat{\mathbf{q}}$ which satisfies the
       disparity gradient limit $\epsilon$ and the epipolar constraint
12         Set $\mathbf{q}^i.s_{ab} = -\infty$
13         if $\mathcal{J}_a(\texttt{round}(\mathbf{q}^i.\mathbf{x}_a)) = 0$ & $\mathcal{J}_b(\texttt{round}(\mathbf{q}^i.\mathbf{x}_b)) = 0$
14             $[\mathbf{q}^i.s_{ab}, \mathbf{q}^i.v] = \texttt{sim}(\mathbf{q}^i, \mathcal{I}_a, \mathcal{I}_b)$
15     end for
16     Sort matches $\mathbf{q}^i$ according to the scores $\mathbf{q}^i.s_{ab}$
17     for each $\mathbf{q}^i$ satisfying $\mathbf{q}^i.s_{ab} \geq z$ and $\mathbf{q}^i.v \geq t$
18         Set $n = n + 1$ and update $\mathbf{q}^i.\mathbf{A}_{ab}$
19         Do trifocal transfer for $\mathbf{q}^i$ and compute $\mathbf{q}^i.s_{ac}$, $\mathbf{q}^i.s$
20         Set $\mathcal{Q} = \mathcal{Q} \cup \{\mathbf{q}^i\}$ and $\mathcal{M} = \mathcal{M} \cup \{\mathbf{q}^i\}$
21         Set $\mathcal{J}_a(\texttt{round}(\mathbf{q}^i.\mathbf{x}_a)) = n$, $\mathcal{J}_b(\texttt{round}(\mathbf{q}^i.\mathbf{x}_b)) = n$
22         if $\mathcal{J}_c(\texttt{round}(\mathbf{q}^i.\mathbf{x}_c)) = 0$ & $\mathbf{q}^i.s_{ac} \geq z$
23             Set $\mathcal{J}_c(\texttt{round}(\mathbf{q}^i.\mathbf{x}_c)) = n$
24     end for
25 end while
```

Three-view matching. This paper proposes modifications to the method in [8] in order to improve the robustness of matching for view triplets when the trifocal tensor is known. The main difference to [8] is that the ordering of seeds in the priority queue $\mathcal{Q}$ is based on a total score s which combines two pairwise similarity scores, s_{ab} and s_{ac}, between the reference view a and the other two views. Hence, at each propagation step, the seed with the best total score is grown.

The proposed algorithm is shown in pseudo-code in Alg. 1. It mainly follows the method proposed in [8] but there are certain additional steps (i.e. lines 3, 4, 5, 19, 22, and 23), which are detailed in the following. The data structure for three-view matches is given in Def. 1.

First, since the input to Alg. 1 is a set of two-view seeds, as in [8], the seeds have to be transformed to three-view seeds by using trifocal transfer [6]. That is, we define a function called $\texttt{transfer}$, which transforms a match $(\mathbf{x}_a, \mathbf{x}_b)$ to the third view, indexed by c, and also computes the local affine transformation $\mathbf{A}_{ac}$ between the reference view a and the view c (line 3 in

Alg. 1). The function is implemented so that it uses $\mathbf{x}_a$, $\mathbf{x}_b$ and $\mathbf{A}_{ab}$ to define three corresponding points in the views a and b and then transforms this local affine basis to the third view by trifocal transfer [6]. Thereafter, $\mathbf{A}_{ac}$ may be solved from the three point correspondences between a and c. Finally, given $\mathbf{x}_c$ and $\mathbf{A}_{ac}$, one may also evaluate the local similarity s_{ac} between views a and c.

Given a three-view seed $\mathbf{s}$, the total score $\mathbf{s}.s$, on which the ordering in $\mathcal{Q}$ is based, combines the pairwise similarities between a and the other two views. This allows the three-view method to perform better than a combination of pairwise propagations. The scoring function is defined by

$$\texttt{score}(s_{ab}, s_{ac}, z) = \sum_{j \in b,c} \max\left(0, 1 - \frac{(s_{aj} - 1)^2}{(z-1)^2}\right), \quad (1)$$

which is a positive function on $[-1, 1]^3$. It is nonzero if either s_{ab} or s_{ac} exceed $z \leq 1$, and it obtains the largest values when they both exceed z and are close to 1.

Most of the computation time of Alg. 1 is spent in the while loop. The first for loop is the same as in [8]. Thus, most of the bad candidate matches are rejected already on the basis of the pairwise score s_{ab} and the trifocal transfer and the evaluation of the total score are not necessary for them. In fact, only a fraction of the candidates survive to the second for loop, and hence, the three-view method is only slightly slower than [8].

Finally, the last modification to [8] is the update of the matching table for the third view. A match is accepted if the similarity $s_{ab} \geq z$. However, as shown on lines 22 and 23 in Alg. 1, the match is added to the matching table of the third view only if also $s_{ac} \geq z$ and the corresponding pixel is not already reserved.

As described in Alg. 1, our current implementation is restricted to three views. However, this is not a major limitation since there are several methods for combining multiple depth maps, e.g. [9]. Moreover, due to efficiency and scalability requirements [7], it might be even advantageous to limit the number of images that are processed concurrently and use triplets of nearby views for depth map estimation. Alternatively, one may also extend Alg. 1 for more than three views. In fact, given the camera matrices, it would be straightforward to transfer the two-view matches to arbitrary number of views by the trifocal transfer (lines 3 and 19 in Alg. 1) and define the score (1) as the sum of several pairwise terms.

3. Experiments

Planar scene with repeated texture. In the first experiment we compared our approach with [8] by using several view triplets of a planar calibration pattern. That is, we selected a triplet of views, extracted seed matches from two of them [11], and performed both the two-view growing [8] and the three-view growing with the same seeds. This was repeated for 100 triplets. Usually there were many incorrect seeds because the epipolar constraint was not effective in removing them due to repeated texture. This is illustrated in Fig. 1 where only four seeds are correct. However, also in this case, the three-view method produced almost errorless matching whereas the method in [8] made many errors. The results for all 100 triplets are in Table 1, where the total number of matches is approximately the same for both methods. However, as the quartiles of the error distribution indicate, the matches by [8] have a larger error. Thus, the method in [8] failed in many cases whereas our method was able to produce accurate matches for most of the cases. The running time of the proposed method was about 1.2 times the time of [8]. Hence, the three-view result is usually better than the result of two pairwise propagations and can be computed faster.

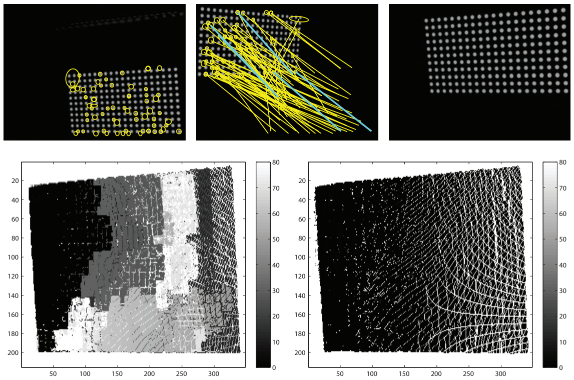

Figure 1: Two-view and three-view matching of a plane with repeated texture. Top: The three views and the seed regions for the first pair in yellow. The rays are plotted between the centers of the regions, and green rays indicate correct correspondences. Down: The grown matches for the algorithm in [8] (left) and our new algorithm (right) are colored with the Euclidean distance to the correct correspondences (in pixels).

Comparison with the state of the art. Our second experiment was done with the Middlebury Temple and Dino datasets [17]. As shown in Fig. 2, we took one triplet of views from both datasets and compared our method to the state-of-the-art multi-view stereo system [4], which has currently the best overall results in accuracy and completeness on the Middlebury data [17].

The method in [4] solves the multi-view stereo problem by growing and filtering 3D patches. In order to make the methods comparable, we set the density pa-

Table 1: The results for the calibration pattern.

Method	No. points	1st quartile	Median	3rd quartile
[8]	5664285	0.33	0.74	11
Ours	5676160	0.079	0.27	0.65

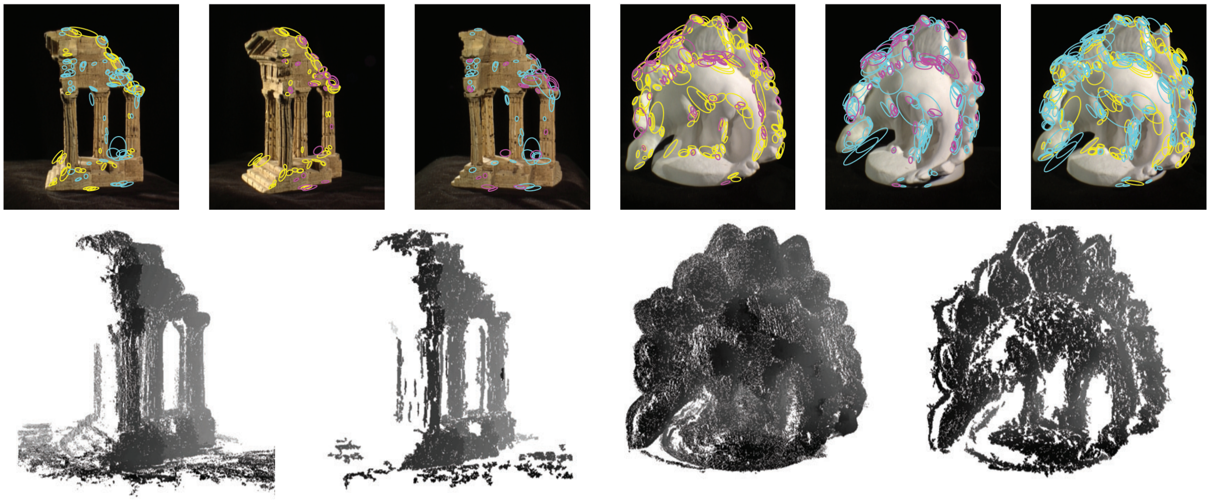

Figure 2: Experiments with view triplets from Temple and Dino datasets. Top: The seed matches $\mathcal{S}_{12}$, $\mathcal{S}_{13}$ and $\mathcal{S}_{23}$ are illustrated with different colors. Bottom: The depth maps produced by our method (left) and the method in [4] (right).

rameter in [4] so that the program tries to reconstruct a patch in every pixel. In addition, the window sizes for computing the ZNCC:s and the ZNCC thresholds were the same for both methods. Further, we let the method in [4] to detect seed features in every pixel in the first stage of the algorithm. The seed matches used in our method were obtained by [11] and are shown in Fig. 2.

The depth maps produced by the methods are illustrated in Fig. 2, which shows that our algorithm produces a denser point cloud. This is expected because our algorithm accepts also two-view matches whereas for [4] we required the 3D patches to be visible in all three images, as recommended by the authors. We tried the method in [4] also so that it accepted patches visible in only two images but in this case the reconstructions were worse than those in Fig. 2. This might be due to the fact that the best-first strategy is not used in [4].

The results in Fig. 2 also suggest that accurate growing is possible without expensive iterative optimization for the pose of matched patches. Hence, although iterative surface refinement is probably necessary at the final stage for the best reconstruction results [4], it might be unnecessary at the depth map estimation stage in approaches that are based on depth map fusion.

4. Conclusion

In this paper, we have proposed a three-view matching method which can be used for depth map estimation in multi-view stereo. The proposed approach employs the best-first matching strategy, is easily expandable to multiple views and provides results that are comparable to the state of the art within correspondence growing methods. Importantly, the matching result is achieved with a single growth stage and without iterative refinement for the matched patches during growing.

References

[1] N. Ayache and F. Lustman. Trinocular stereo vision for robotics. *TPAMI*, 13(1):73–85, 1991.

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

[3] J. Čech and R. Sára. Efficient sampling of disparity space for fast and accurate matching. In *CVPR*, 2007.

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

[5] D. Gallup et al. Real-time plane-sweeping stereo with multiple sweeping directions. In *CVPR*, 2007.

[6] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge, 2000.

[7] M. Jancosek et al. Scalable multi-view stereo. In *3DIM*, 2009.

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

[9] P. Labatut et al. Efficient multi-view reconstruction of large-scale scenes using interest points, Delaunay triangulation and graph cuts. In *ICCV*, 2007.

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

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

[12] M. Pollefeys et al. Detailed real-time urban 3D reconstruction from video. *IJCV*, 78(2-3):143–167, 2008.

[13] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[14] R. Tyleček and R. Sára. Depth map fusion with camera position refinement. In *CVWW*, 2009.

[15] J. Xiao et al. Learning two-view stereo matching. In *ECCV*, 2008.

[16] H. Zhang et al. A linear method for trinocular rectification. In *BMVC*, 2003.

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