Fast and Accurate Multi-View Reconstruction by Multi-Stage Prioritized Matching

Markus Ylimäki[1], Juho Kannala[1], Jukka Holappa[1],
Sami S. Brandt[2] and Janne Heikkilä[1]

[1]Department of Computer Science and Engineering,
P.O.Box 4500, 90014 University of Oulu, Finland
firstname.lastname@ee.oulu.fi
[2]Department of Computer Science, University of Copenhagen,
Universitetsparken 1, 2100 Copenhagen, Denmark
sbrandt@diku.dk

Abstract

In this paper, we propose a multi-view stereo reconstruction method which creates a three-dimensional point cloud of a scene from multiple calibrated images captured from different viewpoints. The method is based on a prioritized match expansion technique, which starts from a sparse set of seed points, and iteratively expands them into neighboring areas by using multiple expansion stages. Each seed point represents a surface patch and has a position and a surface normal vector. The location and surface normal of the seeds are optimized using a homography-based local image alignment. The propagation of seeds is performed in a prioritized order in which the most promising seeds are expanded first and removed from the list of seeds. The first expansion stage proceeds until the list of seeds is empty. In the following expansion stages, the current reconstruction may be further expanded by finding new seeds near the boundaries of the current reconstruction. The prioritized expansion strategy allows efficient generation of accurate point clouds and our experiments show its benefits compared with non-prioritized expansion. In addition, a comparison to the widely used patch-based multi-view stereo software (PMVS) shows that our method is significantly faster and produces more accurate and complete reconstructions.

1 Introduction

Three-dimensional reconstruction of an object or a scene from multiple photographs is a classical problem in computer vision. However, the problem is still very topical and many approaches towards an automated model acquisition have been proposed by the research community during recent years [1, 2, 3, 4]. Although it is now possible to get accurate reconstructions purely from images, the problem is still actively studied and further improvements are being continuously proposed [5, 6, 7].

To date, there are several approaches to scene modeling. For example, Microsoft Kinect provides depth images at a video frame rate by utilizing structured light patters from infrared light source. However, Kinect has limited range and can be used only indoors. Also, one can make very accurate reconstructions using advanced laser scanner technology but such devices are still quite expensive. Hence, passive image-based methods, which reconstruct objects and scenes from multiple photographs using multi-view stereo techniques, have retained their attractiveness. In particular, photographs maintain the textures of surfaces and therefore results of image-based modeling methods are suitable for applications such as virtual tourism and navigation, where the color and texture are important parts of the model.

The development of image-based modeling and multi-view stereo reconstruction methods [8, 9] has been intensive during the recent years. The methods can be roughly categorized to three groups: (a) global approaches which represent shapes on a dense volumetric grid [10, 11, 12, 13, 14], (b) depth map based approaches, which generate a surface model by fusing multiple depth maps [15, 16, 17, 18, 19, 20, 21, 22], and (c) surface expansion methods [23, 24, 3, 25, 26] which build a quasi-dense point cloud from a dense set of matches. The global approaches based on a volumetric grid usually minimize a global cost function for shapes, e.g. by finding a minimum cut on a graph which represents a grid of voxels. The accuracy of models depends on the density of the grid, and therefore, the computational and memory costs of volumetric grids are very high and they are not suitable for modeling large-scale scenes, like cities. Instead, the depth map based approaches and surface expansion methods are suitable for modeling a wide variety of scenes. The depth map based approaches first generate multiple depth maps from a set of images and then fuse them into a complete surface model. The methods based on surface expansion iteratively expand a sparse set of matches into a point cloud. Both of these approaches can produce good results [27, 4]. In fact, often the final model produced from a point cloud or depth maps is represented using a triangle mesh, and there are specific methods for converting point clouds and depth maps into meshes [28, 29] and for refining the resulting meshes [27, 30, 4].

In this paper, we propose a multi-view stereo reconstruction method based on surface expansion which iteratively expands a sparse set of initial matches into a quasi-dense point cloud representing the surfaces of the scene. Our work is inspired by the methods in [31, 25, 3] and builds upon the techniques presented in [32, 26]. While the algorithms in [31, 25, 32] use the match propagation between two views, the method in [26] utilizes three views. Adding more views makes the matching more robust for repeating textures, and therefore, in this work, we do not constrain the number of views that are matched simultaneously. A preliminary conference version of our method was published in [33]. In this paper we give a detailed description of the algorithm and a new experimental comparison between prioritized and non-prioritized match expansion strategies. In addition, we propose two extensions which were not used in [33]: refinement of reconstructed points and multi-stage expansion. Our experiments show that both of these extensions are important as they improve both the accuracy and completeness of reconstructions.

The input to our method is a set of images, the corresponding camera projection matrices and a sparse set of matched interest points called *seeds*. At first, the seeds are ordered into a priority queue based on their similarity score. Then the seeds are expanded from the queue one by one so that the best seed is always expanded first. In the expansion, the purpose is to find new matches nearby the current seed and each new match whose quality exceeds a threshold is put to the queue and used as a seed in further iterations. The expansion continues until the queue is empty. The vast majority of the points of the final reconstruction are acquired during the first expansion step. In order to fill the holes which possibly remain after the first expansion step, we repeat the propagation with a set of new seeds located near holes or outer boundaries of the current reconstruction. That is, we find matches which have unreconstructed pixels nearby and use them as new seeds.

Compared to our preliminary conference version [33], the two extensions, point refinement and multi-stage expansion, have a significant positive impact on results. That is, the refinement of points and the associated surface normal vectors improves the accuracy of reconstructions and makes the expansion less dependent on the quality of seeds, and the multi-stage expansion is able to fill unreconstructed areas that may remain after the first expansion stage. Overall, we are able to make more accurate and more complete reconstructions than [33] but still maintain a good efficiency.

The rest of this paper is organized as follows: Section 2 gives an overview of correspondence growing methods. Section 3 describes the key elements of the proposed algorithm. The experimental results are presented and discussed in Sections 4 and 5 and Section 6 concludes the paper.

2 Correspondence Growing

Correspondence growing, also known as match propagation or match expansion, is a widely used approach for finding point correspondences between multiple images in multi-view stereo reconstruction methods [31, 25, 32, 23, 34, 3, 26]. In this work, we use a similar greedy best-first match expansion as in [31, 32, 26] but, instead of limiting to the case of two [31, 25, 32, 35, 36] or three views [26], we extend the method to the generic multi-view stereo setting with arbitrary number of images. There are also other correspondence growing methods that can process more than three views simultaneously [23, 34, 3] but many of them do not use prioritized expansion [23, 3]. Although the principle of prioritized expansion has been used before in [34], the apparently most successful and widely used match expansion method [3] (PMVS) is non-prioritized. According to the comparison in [34] the multi-view stereo system based on [3] gave better performance with standard benchmark datasets than [34]. However, this difference may be due to the expansion technique of [34], which differs from ours and still contains repeated expansion and filtering stages like [3], or other aspects of the systems of [34] and [3] than matching (e.g. meshing). In fact, one of the key contributions of our work is to show that with prioritized matching one can produce more accurate and complete point clouds than [3], and also significantly faster. In addition, by comparing prioritized and non-prioritized versions of the same correspondence growing method we are able to verify the impact and benefits of priority-ordered expansion. To the best of our knowledge, our work is the first one to show the significance of prioritized correspondence growing in the generic multi-view stereo setting. Furthermore, since the usefulness of [3] as a core algorithm for large-scale reconstruction systems has been extensively demonstrated, e.g.[27, 37], and is widely used, also in recent work [38, 5], we believe that this finding may have significant implications and our method has potential to be widely useful as a more efficient alternative to [3].

The overall structure of our match expansion technique is somewhat similar to [31, 32, 26], but different from [34], and therefore we briefly review the basic propagation concept from [31, 32] in the following.

Generally, the propagation starts from a sparse set of seeds between two images, $\mathcal{I}_a$ and $\mathcal{I}_b$, and produces a quasi-dense set of matches [31]. The seeds are a set of ($\mathbf{x}_a$, $\mathbf{x}_b$) pairs, where $\mathbf{x}_a$ and $\mathbf{x}_b$ are corresponding pixels in the two images. The seeds can be expanded in arbitrary order or sequentially or in order which is based on their quality measured by similarity of corresponding image patches in the two images. The latter approach, i.e. prioritized expansion based on quality, performs often well because in that case inaccurate and incorrect matches are less likely to prevent expansion of correct matches, and therefore partial propagation is usually not a problem. One commonly used similarity measure is the Zero-Mean Normalized Cross Correlation (ZNCC), which is calculated between two patches of pixels around $\mathbf{x}_a$ and $\mathbf{x}_b$. Also some consistency measure is needed to prevent propagation into too uniform areas. Generally, the prioritized propagation proceeds by iterating the following steps [32]:

1. the seed ($\mathbf{x}_a$, $\mathbf{x}_b$) with the best ZNCC score is taken from the list of seeds
2. new candidate matches are searched around the spatial neighbourhood of the seed ($\mathbf{x}_a$, $\mathbf{x}_b$)
3. the candidate matches whose ZNCC score and consistency measure exceed certain thresholds are added to the final list of matches and to the list of seeds.

Each pixel can be matched only once, and therefore, the list of seeds becomes finally empty and the propagation stops. Further, the resulting matches can be triangulated to form a quasi-dense set of points in the 3-D space.

In *two-view stereo* case, the match expansion process may often lead to incorrect correspondences having high similarity due to repeating textures. *Multi-view stereo* based approaches tackle these problems by calculating the similarity between several views [26, 33, 3]. In the following section we describe our multi-view extension of the above two-view propagation strategy. Since the details of the implementation have a significant impact on both efficiency of the method and quality of reconstructions, the representation in the following sections is relatively detailed including pseudocode descriptions.

3 Algorithm

3.1 Overview

The proposed approach builds upon two-view and three-view match propagation methods described in [32, 26]. The method consists of two steps: an initial feature extraction and matching step and an iterative match expansion step.

An overview of the method is presented in Figure 1 and in Algorithm 1. The additional lines, which describe extensions with respect to [33], have been underlined in Algorithm 1. The method first sets initial seeds by matching points of interest between image pairs. The seeds are represented as patches as is commonly done in many stereo reconstruction methods. The data structure for the seed points is shown in Table 1. After acquiring the seeds, their quality is evaluated and they are stacked into a priority queue $\mathcal{Q}$ according to their quality. The quality is based on ZNCC score which is a relatively fast measure and robust against illumination variations [39]. The seeds are expanded from the queue to form a quasi-dense point cloud that represents the surfaces of the scene. The expansion proceeds in a prioritized manner so that the most promising seed is expanded first. Next, the method searches for unreconstructed areas in the current reconstruction and generates a set of new seeds which can be used to cover these regions. This may help to fill some partially occluded regions. The expansion step is iterated until all seeds are used and no more empty regions are found.

3.2 Initial Seeds

The initial seeds are obtained by matching interest points between each image pair of the dataset. In our

Table 1. Data structure for multi-view seed points

Variable	Description
a, b	Indices to the primary and secondary reference images
V	Vector of binary variables of the visibilities of the seed, i.e. if $V_i = 1$ the seed is visible in view i
$\mathbf{x}_a, \mathbf{x}_b$	Corresponding pixel coordinates in images $\mathcal{I}_a$ and $\mathcal{I}_b$
$\mathbf{X}$	Coordinates of a point in 3D space
$\mathbf{n}$	Surface normal at $\mathbf{X}$
s	Total similarity score of a seed
s_{ab}	Similarity score between between patches in images $\mathcal{I}_a$ and $\mathcal{I}_b$
v	Minimun intensity variance of the patches around the seed in all images where the seed is visible
v_{ab}	Minimum intensity variance of patches in images $\mathcal{I}_a$ and $\mathcal{I}_b$

experiments, we use Hessian-Affine regions [40] which are matched using SIFT (Scale Invariant Feature Transform) descriptors [41] but in general any affine covariant region matching technique could be used. Hessian-Affine detector is relatively fast and has been shown to outperform several other region detectors [40]. However, e.g. ASIFT [42] could have been used as well. The descriptor matching step (line 4 in Algorithm 1) uses one-to-many matching method, i.e. a point in an image may be matched with several points in another image, which satisfy the epipolar constraint, i.e. whose distance to the corresponding epipolar line is below a certain threshold. In the following sections, the notation $a.b$ points to the variable b in the structure a.

Now, given a pair of regions in views $\mathcal{I}_a$ and $\mathcal{I}_b$ and local affine transformation between them, the region centroids $\mathbf{x}_a$ and $\mathbf{x}_b$ are triangulated to get a point $\mathbf{X}$ in 3D space. The surface normal $\mathbf{n}$ at $\mathbf{X}$ can be computed by triangulating two additional point correspondences which can be easily synthesized given the corresponding local affine frames at $\mathbf{x}_a$ and $\mathbf{x}_b$. These tentative normal vectors are further refined during the propagation as described in Section 3.4. The local affine frames are pixel patches around the seed matches. In the view $\mathcal{I}_a$ the patch is a square and in the view $\mathcal{I}_b$ the patch is an affine transformed version of that, depending on the geometry between the views. The image pair from which $\mathbf{X}$ is triangulated defines the reference views for the seed (i.e. $\mathbf{s}.a$ and $\mathbf{s}.b$ for seed $\mathbf{s}$). In addition, each seed point s contains variables s, s_{ab}, v, v_{ab}, and V (calculated at lines 10-15 in Algorithm 1). Here s is the total quality score computed by combining pairwise similarity scores between local image patches in the reference view $\mathbf{s}.a$ and the other views, s_{ab} is the similarity score between two image patches, v is the minimum intensity variance of patches in all images where the seed is visible, v_{ab} is the minimum intensity variance of two image

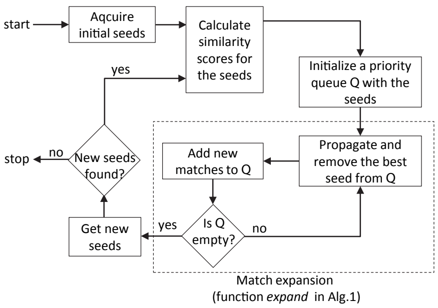

Figure 1. A simplified overview of the proposed algorithm.

Table 2. Pseudocode description of the algorithm illustrated in Figure 1. Function `expand` **at line 16 refers to Algorithm 2 in Table 3.**

Algorithm 1: Overview of the method

```
Input:  images $\mathcal{I}_j$, camera matrices $\mathbf{P}_j$,
        thresholds $\epsilon_{\mathrm{d}}$, $\epsilon_{\mathrm{e}}$, $t$, $t_{\mathrm{u}}$, $z$, $K$
Output: list of points $\mathcal{M}$, matching tables $\mathcal{J}_j$
1  Initialize seed point list $\mathcal{S} = \emptyset$
2  for each pair of views
3      Find affine covariant image regions
4      Match the regions using local image descriptors
5      Triangulate the matches and evaluate the surface normals
6      Append the triangulated matches to $\mathcal{S}$
7  end for
8  Initialize $\mathcal{M} = \emptyset$, $\mathcal{J}_j(\mathbf{p}) = 0$ for all $j$, $\mathbf{p}$, where $\mathbf{p}$ means pixels
9  while $\mathcal{S}$ not empty
10     for each seed point $\mathbf{s}$ in $\mathcal{S}$
11         for each view $k$ where $\mathbf{s}$ is in the field of view
12             Compute pairwise similarity score $s_{ak}$ and minimum
               intensity variance $v_{ak}$ between the reference view $\mathbf{s}.a$ and
               the view $k$, i.e. $[s_{ak}, v_{ak}] = \mathtt{sim}(\mathbf{s}, \mathcal{I}_{\mathbf{s}.a}, \mathcal{I}_k, \mathbf{P}_{\mathbf{s}.a}, \mathbf{P}_k)$
13         end for
14         Combine all pairwise scores $s_{ak}$ to get the total score $\mathbf{s}.s$
           and set $\mathbf{s}.v = \mathtt{min}(v_{ak})$
15     end for
16     Propagate the seeds in $\mathcal{S}$, i.e
       expand( $\mathcal{S}$, $\epsilon_{\mathrm{d}}$, $\epsilon_{\mathrm{e}}$, $t$, $t_{\mathrm{u}}$, $z$, $K$, $\mathcal{M}$, $\{\mathcal{I}_j, \mathbf{P}_j, \mathcal{J}_j\}_{\forall j}$ )
17     Get new seeds and initialize $\mathcal{S}$ with them
18 end while
```

patches, and V is a vector of binary variables defining the views where the seed is visible. The seed is marked as visible in view k (i.e. $V_k = 1$) only if it is in the field of view of view k and the pairwise correlation s_{ak} between the reference view a and the view k exceeds the ZNCC threshold z. The seed is in the field of view if $\mathbf{s}.\mathbf{X}$ is in front of the camera and the angle between the normal $\mathbf{s}.\mathbf{n}$ and the ray between the camera centre and $\mathbf{s}.\mathbf{X}$ is less than 90°. The variables are summarized in Table 1. Seeds are represented globally as illustrated in Figure 2.

The pairwise similarity score s_{ak} between the local patches in the reference view $\mathbf{s}.a$ and view k is defined with a similarity measure `sim` (line 12 in Algorithm 1), which computes the ZNCC (Zero-mean Normalized Cross Correlation) s_{ak} between the local patches in images $\mathcal{I}_{\mathbf{s}.a}$ and $\mathcal{I}_k$ as well as the minimum value v_{ak} of intensity variances of the two patches. The patch P in $\mathcal{I}_{\mathbf{s}.a}$ is a W pixels wide square, and the patch in $\mathcal{I}_k$ is an affine transformed version of P. We use $W = 7$ in all our experiments. The affine transformation is determined using camera matrices $\mathbf{P}_a, \mathbf{P}_k$ and the surface position and orientation at $\mathbf{s}$, i.e. $\mathbf{s}.\mathbf{X}$ and $\mathbf{s}.\mathbf{n}$. For getting a total quality score for the seed, the 3-D point $\mathbf{s}.\mathbf{X}$ is projected to all other views where it is in the field of view. Then, multiple pairwise correlations are calculated between a patch in the reference view $\mathbf{s}.a$ and patches in those views. The combination of the pairwise correlations s_{ak} is carried out with equation

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

where z is a parameter with a given fixed value (the same value is used also as a ZNCC threshold elsewhere in Algorithms 1 and 2) [26]. That is, the total quality of a seed depends on both the pairwise correlations and the number of images where the correlations exceed the threshold z.

Minimum intensity variance v_{ab} is used to prevent the propagation from spreading in too uniform areas. Using V, we are able to control the minimum number of views where each match should be visible.

Table 3. Pseudocode algorithm of the match expansion function called at line 16 in Algorithm 1 in Table 2.

```
Algorithm 2: Multi-view match expansion
Input : images $\mathcal{I}_j$, camera matrices $\mathbf{P}_j$, seed points $\mathcal{S}$,
        thresholds $\epsilon_{\mathrm{d}}$, $\epsilon_{\mathrm{e}}$, $t$, $t_{\mathrm{u}}$, $z$, $K$,
        pointers to the list of points $\mathcal{M}$ and matching tables $\mathcal{J}_j$

Initialize $n = size(\mathcal{M})$
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 $b = \hat{\mathbf{q}}.b$
  if $\hat{\mathbf{q}}.v > t_{\mathrm{u}}$
    Refine $\hat{\mathbf{q}}.\mathbf{X}$ and $\hat{\mathbf{q}}.\mathbf{n}$, i.e. $[\hat{\mathbf{q}}.\mathbf{X}, \hat{\mathbf{q}}.\mathbf{n}] = \mathtt{ref}(\hat{\mathbf{q}}, \mathcal{I}_a, \mathcal{I}_b, \mathbf{P}_a, \mathbf{P}_b)$
  for each new match $\mathbf{q}^i$ nearby $\hat{\mathbf{q}}$ which satisfies the
  disparity limit $\epsilon_{\mathrm{d}}$ and the epipolar constraint $\epsilon_{\mathrm{e}}$
    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 $n = n + 1$, $\mathbf{q}^i.\mathbf{n} = \hat{\mathbf{q}}.\mathbf{n}$, $\mathbf{q}^i.a = a$ and $\mathbf{q}^i.b = b$
    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 $\mathbf{q}^i.\mathbf{X}$ is in the field of view
      Project $\mathbf{x}_k = \mathbf{P}_k(\mathbf{q}^i.\mathbf{X})$, set $s_{ak} = -\infty$
      if $\mathcal{J}_k(\mathtt{round}(\mathbf{x}_k)) = 0$
        $[s_{ak}, v_{ak}] = \mathtt{sim}(\mathbf{q}^i, \mathcal{I}_a, \mathcal{I}_k, \mathbf{P}_a, \mathbf{P}_k)$
        if $s_{ak} \geq z$
          Set $\mathbf{q}^i.V_k = 1$
    end for
    if $\mathtt{sum}_j(\mathbf{q}^i.V_j) \geq K$
      Combine pairwise scores $s_{ak}$ to get $\mathbf{q}^i.s$ and
      set $\mathbf{q}^i.v = \mathtt{min}(\mathbf{q}^i.v, v_{ak})$
      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
```

3.3 Match Expansion

An overview of the expansion process is illustrated in Figure 2. Let $\mathbf{s}$ denote the current seed taken from the priority queue of the initial seeds. First, the algorithm finds a new candidate correspondence pair in the spatial neighbourhood of the current seed in its reference views (red crosses). Then the new match is triangulated to the space to get the corresponding 3-D point (blue star) and the surface normal. The normal is inherited from the current seed. Finally, the point in the space is projected to all other views where it can be visible (green plus signs) and the quality of this new match is evaluated among all those views including reference views. If the

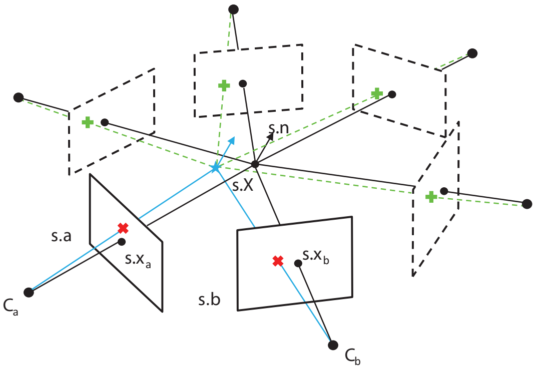

Figure 2. Example of a seed and expansion process in a set of views. A seed consists of reference views $\mathrm{s}.a$ and $\mathrm{s}.b$, correspondence pair $\mathrm{s}.\mathbf{x}_a$ and $\mathrm{s}.\mathbf{x}_b$, corresponding point in 3D space $\mathrm{s}.\mathbf{X}$ and surface normal $\mathrm{s}.\mathbf{n}$. $\mathbf{C}_a$ and $\mathbf{C}_b$ are the viewpoints of the reference views $\mathrm{s}.a$ and $\mathrm{s}.b$, respectively. The seed s is projected to every dashed view where $\mathrm{s}.\mathbf{X}$ is in the field of view and its quality is evaluated among all those views including reference views. The expansion process is illustrated with colored symbols (i.e. crosses, a star and plus signs). See the text for details.

quality exceeds a threshold, the match is added to the final reconstruction and to the priority queue to be used as a seed in further iterations. The actual prioritized match expansion phase begins at line 16 in Algorithm 1 with the function `expand`. The details of the expansion function are presented in Algorithm 2 and they are explained in the following.

As an input the function gets the images $\mathcal{I}_j$ and corresponding camera matrices $\mathbf{P}_j$, the initial seeds $\mathcal{S}$, the thresholds ϵ_{d}, ϵ_{e}, t, t_{u}, z and K and pointers to the list of reconstructed points $\mathcal{M}$ and to the matching tables $\mathcal{J}_j$. The matching tables $\mathcal{J}_j$ are index matrices which can be used to locate the particular 3-D point of a pixel in the list of matches. The disparity threshold ϵ_{d} defines an area of possible matches for a point inside the spatial neighbourhood of the current seed [31]. Threshold ϵ_{e} is the epipolar constraint, i.e. the maximum distance between a point in one image and the epipolar line of a corresponding point in another image, t is the threshold for intensity variances preventing the propagation from spreading in too uniform areas, t_{u} is the threshold for intensity variances for refinement, z limits the pairwise ZNCC scores and K defines the minimum for the number of views where matches should be visible.

Starting from line 4 in Algorithm 2, the propagation function expands the seeds in the priority queue $\mathcal{Q}$. The queue is initialized at line 3 and at the beginning of every while loop iteration, the seed with the best score is taken from the queue. At the end of each iteration, new matches are added to the list of matches $\mathcal{M}$ and to the priority queue as seeds to be further expanded.

Every new match inherits the surface normal from its parent seed. Therefore, before searching for new candidate matches, the position and orientation of the current seed in the 3D space are refined. The refinement process is described in Section 3.4. Then, (lines 9-13) new candidate matches are searched in the spatial neighbourhood of the current seed in the reference views. The process is identical to [31, 32, 26, 33]. The candidate matches whose similarity scores and minimum variances exceed the thresholds z and t, respectively, are then triangulated to 3D space and their similarity is evaluated globally. In the global evaluation process (lines 19-25), the triangulated 3D point is projected to each view where it is in the field of view and the pairwise similarity score between patches in the current view and the reference view a is calculated. If the score exceeds the threshold z, the seed is marked to $\mathrm{s}.V$ as visible in that view. If a new seed is visible in at least K views, the pairwise similarity scores, including the score between reference views, are combined using (1) and the seed is added to the list of points $\mathcal{M}$ and to the priority queue $\mathcal{Q}$ and the matched pixels are marked as reserved to $\mathcal{J}_j$ for those seeds where s is visible. The expansion process continues as long as there are seeds in the priority queue $\mathcal{Q}$.

The vast majority of the reconstructed points is acquired during the first expansion but, due to the fixed reference views of seeds, some areas, which are not visible in the reference views due to occlusions, cannot be reconstructed. However, in some cases these areas can be visible in another pair of views. Therefore, at the line 17 in Algorithm 1, new seeds are searched from the current reconstruction and used as initial seeds $\mathcal{S}$ for the next iteration. In the searching process, the matching tables $\mathcal{J}_j$ are used to find matches which can be further propagated. That is, if the ratio of empty pixels in the pixel neighbourhood of a match is between 0.4 and 0.9 (value 1 means a totally empty region, which do not contain possible seeds) in at least three images, the match is added to the list of seeds so that the original reference views of the match are replaced by new ones from the set of views where the match is visible. The while loop (lines 9-18 in Algorithm 1) is repeated until no new seeds can be found at line 17. Figure 3 shows how the point cloud evolves during the match expansion phase. Surfaces which are seen in many images

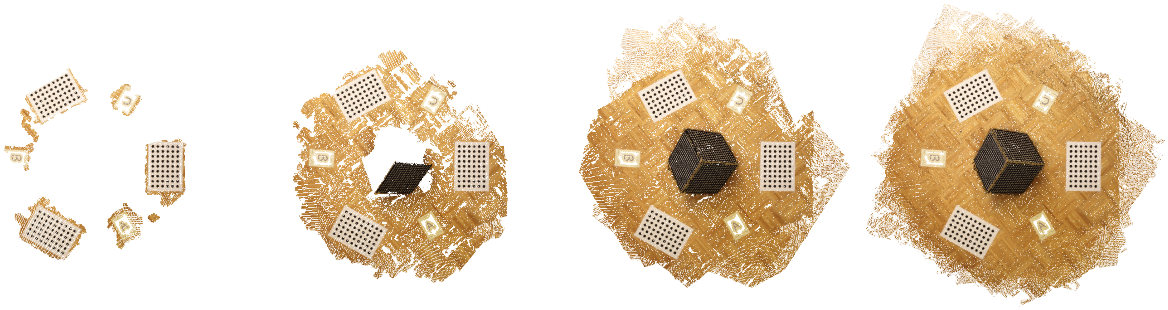

Figure 3. Evolution of the point cloud during the match expansion phase. From left to right: first 50k points that were reconstructed, 200k points, 400k points and the final reconstruction with 585785 points. Areas with highly varying texture are reconstructed first.

and which have rich texture are reconstructed first.

3.4 Match Refinement

Each seed drawn from the priority queue (line 5 in Algorithm 2) whose minimum variance exceeds the threshold t_{u} is refined (line 8 in Algorithm 2). The variance threshold for refining is necessary because seeds inside too uniform areas, i.e. areas with low variance, can not be refined very accurately. The function refines the position and orientation of the seed point in the 3D space. As illustrated in Figure 4, the refinement is based on Lucas-Kanade image alignment procedure where a template patch, situated at $\hat{\mathbf{q}}.\mathbf{x}_a$ in $\mathcal{I}_a$, is aligned into $\mathcal{I}_b$ so that the sum of squared difference is minimized [43], i.e. we minimize E with respect to $\Delta\hat{\mathbf{p}} = \begin{pmatrix}\Delta\mathbf{p} & \Delta\lambda_1 & \Delta\lambda_0\end{pmatrix}^T$:

$$E = \sum_{\mathbf{x}} [\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}{+}\Delta\mathbf{p})) - (\lambda_1{+}\Delta\lambda_1)\mathbf{T}(\mathbf{x}) - (\lambda_0{+}\Delta\lambda_0)]^2, \tag{2}$$

where $\mathbf{T}(\mathbf{x})$ is the template patch (see Fig. 4), $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p}))$ is the image $\mathbf{I}$ warped onto the coordinate frame of the template, $\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})$ denote the set of parameterized warps, $\mathbf{x}$ contains the pixel coordinates and $\mathbf{p}$ denotes the vector of warping parameters. Variables λ_0, λ_1 are used to shift and scale the intensity values of the template to minimize the influence of intensity differences between the template and the image. The algorithm assumes that an initial estimate of $\hat{\mathbf{p}} = \begin{pmatrix}\mathbf{p} & \lambda_1 & \lambda_0\end{pmatrix}^T$ is known and then solves the increments $\Delta\hat{\mathbf{p}}$ iteratively.

In our case, the warps are defined, as in [8], using the homography

$$\mathbf{H_{ab}} = \mathbf{K_b}(\mathbf{R_b} - \mathbf{t_b}\hat{\mathbf{q}}.\mathbf{n}^T/d)\mathbf{K_a}^{\mathbf{-1}}, \tag{3}$$

which is parametrized by $\mathbf{p} = (\hat{\mathbf{q}}.\mathbf{n}^T, d)^T$, i.e. a world plane situated at $\hat{\mathbf{q}}.\mathbf{X}$ and oriented perpendicular to $\hat{\mathbf{q}}.\mathbf{n}$. That is, for the points $\hat{\mathbf{X}}$ on the plane $\hat{\mathbf{q}}.\mathbf{n}^T\hat{\mathbf{X}} + d = 0$. Other variables are camera internal and external parameters from camera matrices $\mathbf{P_a} = \mathbf{K_a}[\mathbf{I}|\mathbf{0}]$ and $\mathbf{P_b} = \mathbf{K_b}[\mathbf{R_b}|\mathbf{t_b}]$.

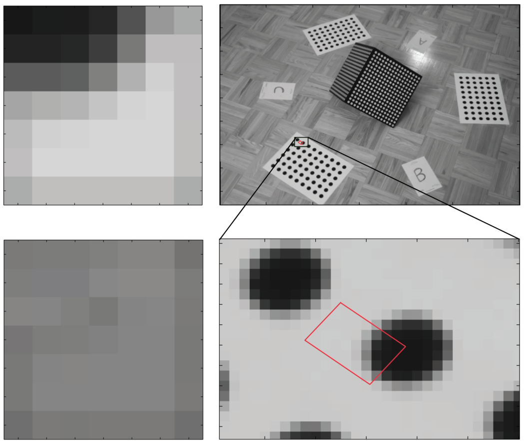

Figure 4. Match refinement based on image alignment. The template patch (top left) extracted from image a is aligned to the image b (right) using a homography $\mathbf{H_{ab}}$ so that the sum of squared pixel values of the difference patch (bottom left) is minimized.

The expression in Eq. (2) can be approximated using

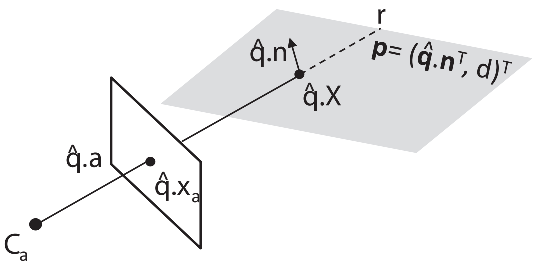

Figure 5. Match refinement adjusts the warping parameters p, which define a plane in the space. After the refinement, the position and the surface normal of the seed $\hat{\mathrm{q}}$ are updated. The normal is updated directly from the plane equation and the new position is at the intersection point between the ray r and the plane.

the first order Taylor expansion on $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p}))$:

$$\begin{aligned} E &\approx \textstyle\sum_{\mathbf{x}}[\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla\mathbf{I}\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\Delta\mathbf{p} \\ &\quad -\lambda_1\mathbf{T}(\mathbf{x}) - \lambda_0 - \Delta\lambda_1\mathbf{T}(\mathbf{x}) - \Delta\lambda_0]^2 \\ &= \textstyle\sum_{\mathbf{x}}[\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) - \lambda_1\mathbf{T}(\mathbf{x}) - \lambda_0 \\ &\quad + \begin{pmatrix}\nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{p}} & -\mathbf{T}(\mathbf{x}) & -1\end{pmatrix}\begin{pmatrix}\Delta\mathbf{p}\\ \Delta\lambda_1 \\ \Delta\lambda_0\end{pmatrix}]^2, \end{aligned} \tag{4}$$

where $\nabla\mathbf{I}$ is the gradient of image $\mathbf{I}$ evaluated at $\mathbf{W}(\mathbf{x};\mathbf{p})$ and $\frac{\partial\mathbf{W}}{\partial\mathbf{p}}$ is the Jacobian of the warp. By marking $\begin{pmatrix}\nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{p}} & -\mathbf{T}(\mathbf{x}) & -1\end{pmatrix} = \mathbf{M}(\mathbf{x})$ and setting the derivative of E with respect to $\Delta\hat{\mathbf{p}}$ to zero, we get the increments for the next iteration, i.e.

$$\begin{aligned} \Delta\hat{\mathbf{p}} &= [-\textstyle\sum_{\mathbf{x}}\mathbf{M}^T(\mathbf{x})\mathbf{M}(\mathbf{x})]^{-1} \\ &\quad \times \textstyle\sum_{\mathbf{x}}\mathbf{M}^T(\mathbf{x})[\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) - \lambda_1\mathbf{T}(\mathbf{x}) - \lambda_0]. \end{aligned} \tag{5}$$

Then, before the next iteration, the parameters need to be updated to $\hat{\mathbf{p}} \leftarrow \hat{\mathbf{p}} + \Delta\hat{\mathbf{p}}$.

The number of needed refinement iterations may depend on the quality of the initial estimate of $\hat{\mathbf{p}}$. However, in our experiments, the refinement was carried out only once per match.

After refining, the position of the seed, i.e. $\hat{\mathbf{q}}.\mathbf{X}$, is updated so that new position is the intersection point of the plane defined by $\mathbf{p}$ and the ray $\mathbf{r}$ defined by the center of the reference camera $\hat{\mathbf{q}}.a$ and the match position $\hat{\mathbf{q}}.\mathbf{x}_a$ in the corresponding image. The surface normal $\hat{\mathbf{q}}.\mathbf{n}$ is updated directly from the normal of the refined plane $\mathbf{p}$. See the illustration in Figure 5.

4 Experiments

This section is divided into five subsections where we (1) present an overview of the experiments, (2) test the influence of the most critical parameters on accuracy and exection time, (3) validate the importance of best-first growing strategy, (4) compare the accuracy and (5) computation time of our method with those of the publicly available patch-based multi-view stereo software (PMVS)[1] [3].

4.1 Overview

The proposed method was tested against the publicly available PMVS program [3] using publicly available benchmark datasets Fountain-P11, Herz-Jesu-P8 and Castle-P30 [2] and a simple dataset of a cube on a mosaic parquet floor. The datasets are presented in Table 4. The importance of the best-first growing strategy was validated by comparing the results of two different versions of the proposed method: one using the best-first expansion strategy and another using a non-prioritized expansion strategy where seeds were expanded in the order they were found.

Although the whole reconstruction pipeline from images to a watertight mesh is described in [3], only the part which creates a point cloud from a set of images is freely available online. Therefore, the comparisons are made between point clouds instead of meshes. Further, the evaluation in [2] is no longer available, and therefore, the accuracy of the results were compared by calculating depth maps both from the point clouds and the ground truth meshes. The point clouds created from castle-P30 dataset are evaluated only visually because the ground truth is not available [2]. The relative error tells how much the assigned depth of a pixel differs from the ground truth depth, i.e.:

$$error = \frac{|d_{assigned} - d_{groundtruth}|}{d_{groundtruth}}. \tag{6}$$

If multiple depth values were assigned to the same pixel in the depth map, the depth with the smallest error was chosen.

In addition to the quality of the reconstructions, we also compared the execution times of the programs. All experiments were carried out on the same computer with Intel Xeon E5-2650 v2 @ 2.6GHz processor. The processor has eight cores but programs used only a single core.

All the parameters were fixed in the experiments. The PMVS program was used with default parameters

---

[1] `http://www.di.ens.fr/pmvs/`

[2] The benchmark [2] is no longer available.

Table 4. Datasets used in our experiments

Dataset	Images	Image size
Cube	17	644×429
Fountain-P11	11	3072×2048
Herz-Jesu-P8	8	3072×2048
Castle-P30	30	3072×2048

with the following exceptions: *csize* was fixed to value 1 so that the program tries to reconstruct a patch in every pixel like our program does and *CPU* was fixed to value 1 to make the computation times comparable to our approach. In our method, the parameters were W=7, N=11, z=0.8, t=0.001, t_{u}=0.05, ϵ_{d}=1, ϵ_{e}=1, K=3. In addition, the size of the template patch in the refinement stage was fixed to 29×29 pixels. The PMVS program was also tested with the same ZNCC threshold (0.8) but the default value (0.7) gave better results and was used instead.

4.2 Parameter tests

Similarly to [3] and other patch-based matching techniques, the main parameters affecting the resulting point cloud are the ZNCC threshold z and the size of the correlation window W. It is well-known that a higher ZNCC threshold leads to less outliers but more sparse point clouds and a larger window size may increase density and robustness but also computational cost and boundary artifacts at depth discontinuities. Other parameters of our method do not involve similar trade-offs and can be easily set to fixed values that generally perform well. In order to find good default values for z and W, we experimented with various choices. The results, presented in Figure 6, were acquired by varying one parameter at a time and keeping the others constant. The tests were carried out both with the proposed method and PMVS [3] using the Herz-Jesu-P8 dataset. The heights of the curves in the top subfigures show the proportion of reconstructed pixels whose error with respect to the ground truth depth map is less or equal than the corresponding value on the x-axis [2]. The steeper the curve is at the beginning, the more accurate the reconstruction is. Further, the higher the value at the end, the more pixels were reconstructed. Maximum occupancy (100%) is the case where each pixel in every image has been reconstructed, and therefore, it can not be reached in practice. The jumps at the ends of the curves contain all pixels whose error is greater than the largest value on the x-axis.

The methods behave mostly similarly but with the PMVS the execution time per point seems to decrease a bit when using a bigger ZNCC threshold. In general, the smaller the ZNCC threshold or bigger the ZNCC window the more points were reconstructed. On the other hand, with smaller ZNCC thresholds the proportion of inaccurate points is higher and with larger window sizes the computation time per point increases. According to these results, we decided to use values 0.8 and 7 as the ZNCC threshold and the ZNCC window size, respectively. Thus, all the parameters had the same fixed values in all of our experiments.

4.3 Best-first growing strategy

The importance of the prioritized matching principle was tested by comparing the results of our method with and without the best-first principle. The test was carried out using the Cube dataset and, in addition to the error measure described in Section 4.1, the accuracies of the point clouds were also evaluated by calculating the distances between the points and the nearest triangles in the ground truth mesh. The results are presented in Figure 7. In addition to [3] and the two versions of the proposed method, the figure presents also the accuracies of the point clouds produced by the baseline method [33] and the baseline method with the multistage expansion. Both subfigures show the importance of the proposed extensions, and in addition, the importance of the best-first growing strategy. The curves in the figure follow the same representation as the curves in Figure 6 but the left subfigure uses absolute values and errors are distances between points and the ground truth mesh. Figure 8 illustrates the point clouds made with the PMVS program and our program without and with the prioritized matching. The execution times of the different methods with the Cube dataset are listed in Table 5.

The comparisons clearly show the importance of the best-first strategy. Especially the left subfigure in Figure 7 shows that, compared with the point cloud created with the proposed method, the point cloud produced with non-prioritized expansion has somewhat less precise points and much more outliers i.e. points whose error is equal or greater than 8 mm. Outliers can also be observed from the point clouds in Figure 8. The inaccuracies of the dot patterns are due to outliers above the actual surface. From Table 5 we can see that the best-first growing strategy also has an impact on the executions times. The method using the strategy is about 30% faster with the cube dataset and 10% and 15% faster with the Fountain and Herz-Jesu datasets, respectively.

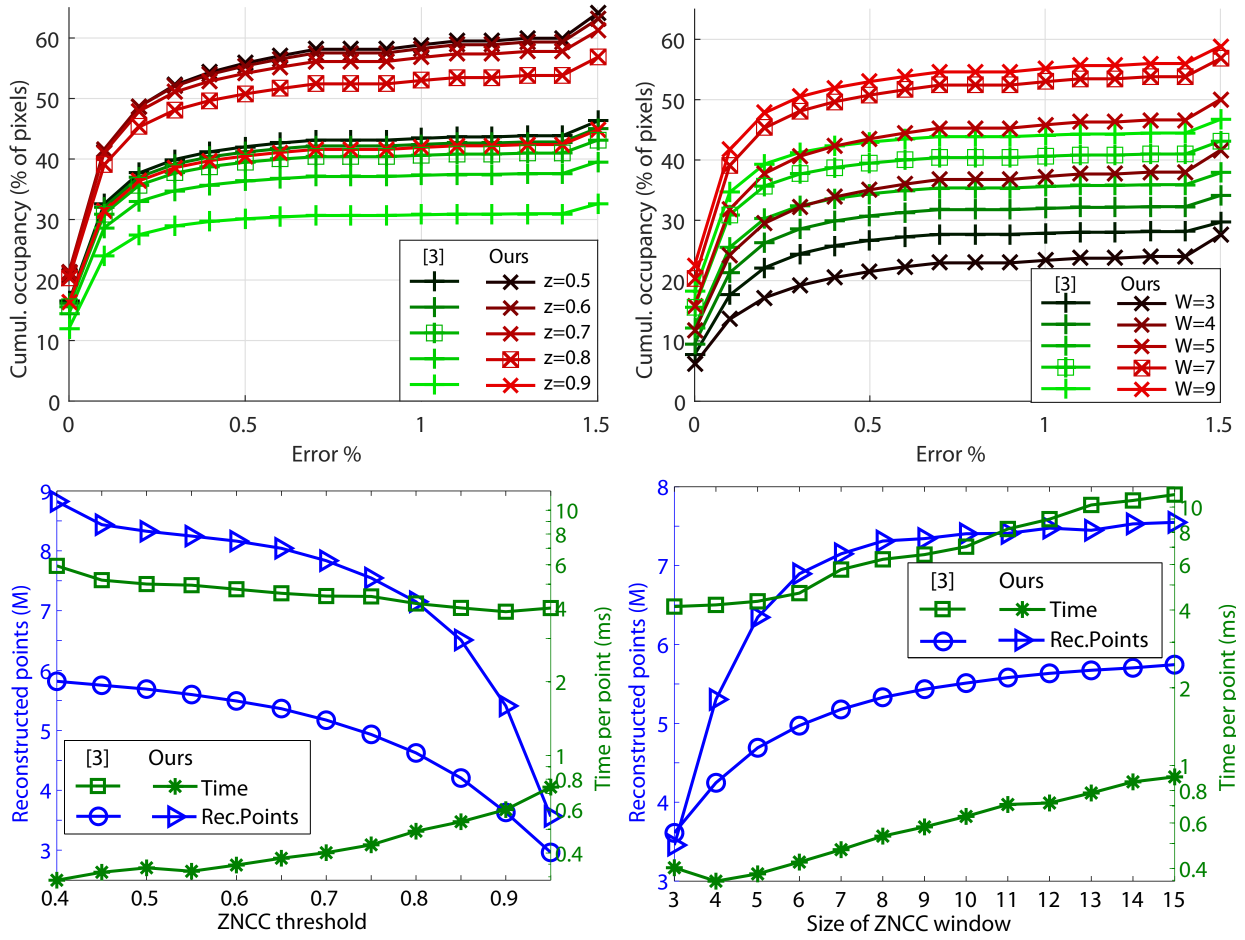

Figure 6. Influence of z (left) and W (right) on the accuracy of point clouds (top) and the performance of the method (bottom) tested with Herz-Jesu-P8 dataset. The curves at top, marked with additional □ symbols, correspond to the chosen parameter values.

4.4 Evaluation of accuracy

The point clouds obtained from Fountain-P11 and Herz-Jesu-P8 datasets by PMVS program [3], our baseline method [33] and the proposed method are presented in Figure 9. The comparisons of the point clouds are illustrated in Figure 10. The curves follow the same representation as the curves in the right subfigure in Figure 7. The point clouds obtained from Castle-P30 dataset by PMSV and the proposed method are presented in Figure 11.

As the results show, the point clouds produced by the proposed method are denser than the PMVS point clouds and cover more areas than our baseline method. Further, compared with the PMVS and our baseline point clouds, the point clouds produced by the proposed method have better accuracy in all cases.

As Figure 10 shows, the multi-stage expansion has significant impact on the amount of reconstructed points (red vs green) whereas the refinement provides further improvements on the accuracy (green vs purple).

4.5 Computational Cost

The computational efficiency, namely execution time and number of reconstructed points, was also recorded in our experiments. For our methods the seed matches were acquired by matching Hessian-Affine regions [40] using SIFT descriptors [41]. Both in Furukawa's program and in our methods, seed matches were detected from every image pair of each dataset. The total execution times contain the initial seed extraction, matching and expansion steps. Table 5 presents the results achieved using Furukawa's method [3], our baseline

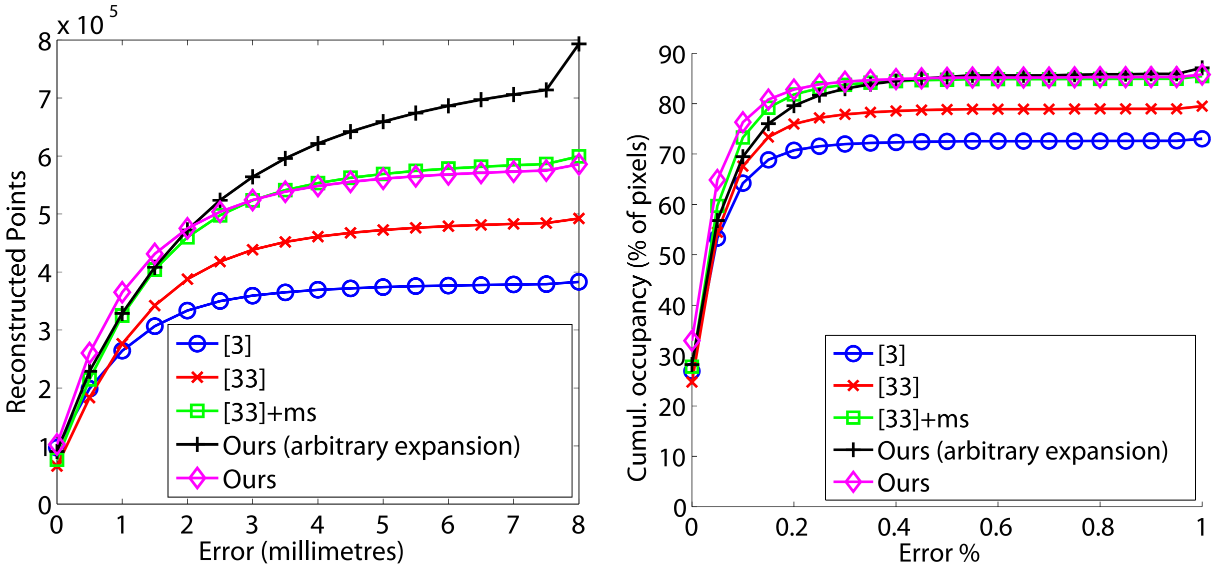

Figure 7. Accuracy of the Cube point clouds against ground truth mesh (left) and ground truth depth maps (right).

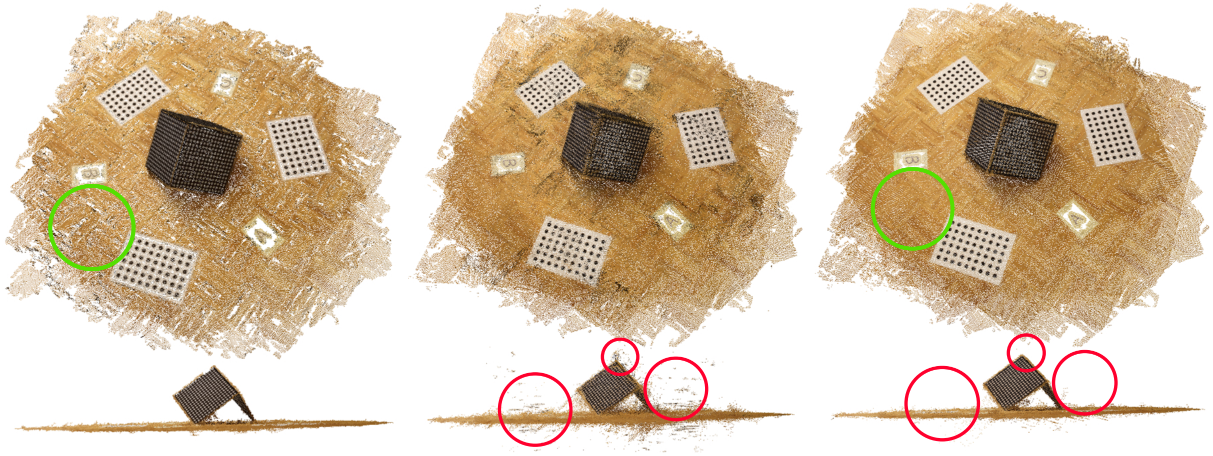

Figure 8. Cube reconstructions build by the PMVS program [3] (left), the proposed method with non-prioritized seed expansion (center) and the proposed method (right). In the middle, the outliers due to non-prioritized expansion are clearly visible (red circles). Our reconstruction is denser than that of PMVS (green circle).

method [33] and the proposed method. In addition, the table shows the results achieved with the baseline method together with the multi-stage extension and also the results of our approach with non-prioritized seed expansion. The Castle reconstructions were made only by the PMVS program and the proposed method.

As one can see from Table 5, the total execution time of the PMVS method is even 7.3 times higher than the execution time of our method. That difference is almost doubled if the number of reconstructed points is also taken into an account.

Note that while the provided extensions improve quality of results they do not decrease the efficiency significantly compared with the baseline method and with the Cube dataset the proposed method is even faster than the baseline method with the multi-stage expansion.

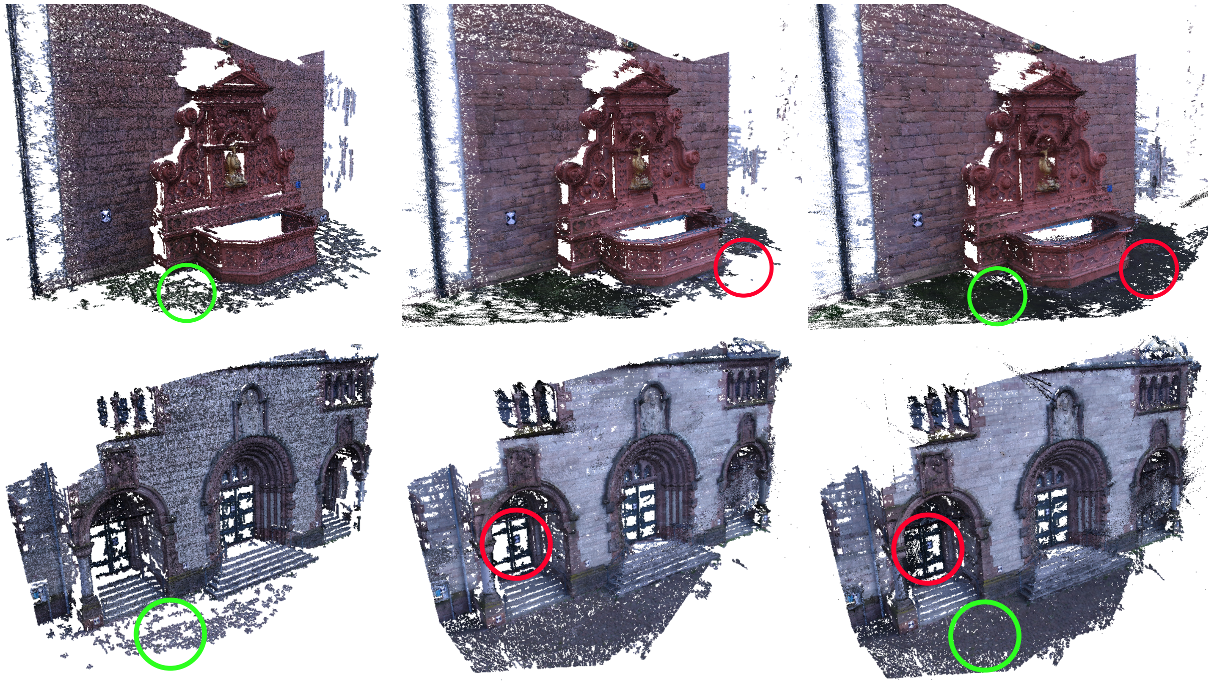

Figure 9. Fountain (top) and Herz-Jesu (bottom) reconstructions build by the PMVS program [3], our baseline method [33] and the proposed method. Our reconstructions are denser than those of PMVS (green circles) and the proposed method fills some holes that appear in the baseline reconstructions (red circles).

5 Discussion

The results in Section 4.3 showed that the expansion becomes more robust against the outlier seeds when the seeds are expanded in the prioritized order. That is, when the best seeds are expanded first, the outlier seeds usually can not expand because the areas that they cover have already been matched. The priorization of the seeds also speeds up the method because less time is wasted on expanding the outlier seeds.

Expanding the matches in the reference images may cause our baseline method to end up in a situation where the final reconstruction has holes. The multi-stage expansion in the proposed method was used to cover that shortcoming with a different pair of reference images, and therefore, it is clear that the proposed method produces more complete reconstructions.

The refinement in the proposed method is based on the texture patches around the seed matches, and therefore, it works better on areas with high texture variations. Those areas can usually be reconstructed relatively well in any case, so the improvement, provided by the refinement, is quite moderate but consistent.

The difference in the computational efficiency between the PMVS program and the proposed method is mainly due to the repeated expansion and filtering steps of the PMVS method. Greedily expanded seeds may produce bad matches which are later removed by the filtering stage. However, this may lead to unnecessary computations. In contrast, in our approach the best-first strategy quarantees the quality of matches and makes the filtering step unnecessary. Hence, a method using the best-first strategy is robust to incorrect initial seeds and reduces the amount of computation. Note that, the initial seed extraction is not optimized in our program, because it is partially a MATLAB implementation while other parts are implemented with C++. Therefore, even further speed-up could be achieved.

Finally, we would like to emphasize that in this work we focus only on the point cloud generation stage that is a core component in many recent best-performing multi-view stereo systems [3, 4]. However, besides point cloud generation, the state-of-the-art systems [3, 4] contain also other components, such as surface mesh generation and refinement. These other components have a large effect on the quality of the result if it is measured using the final refined mesh, as is typically done in standard benchmarks [1, 2]. Nevertheless,

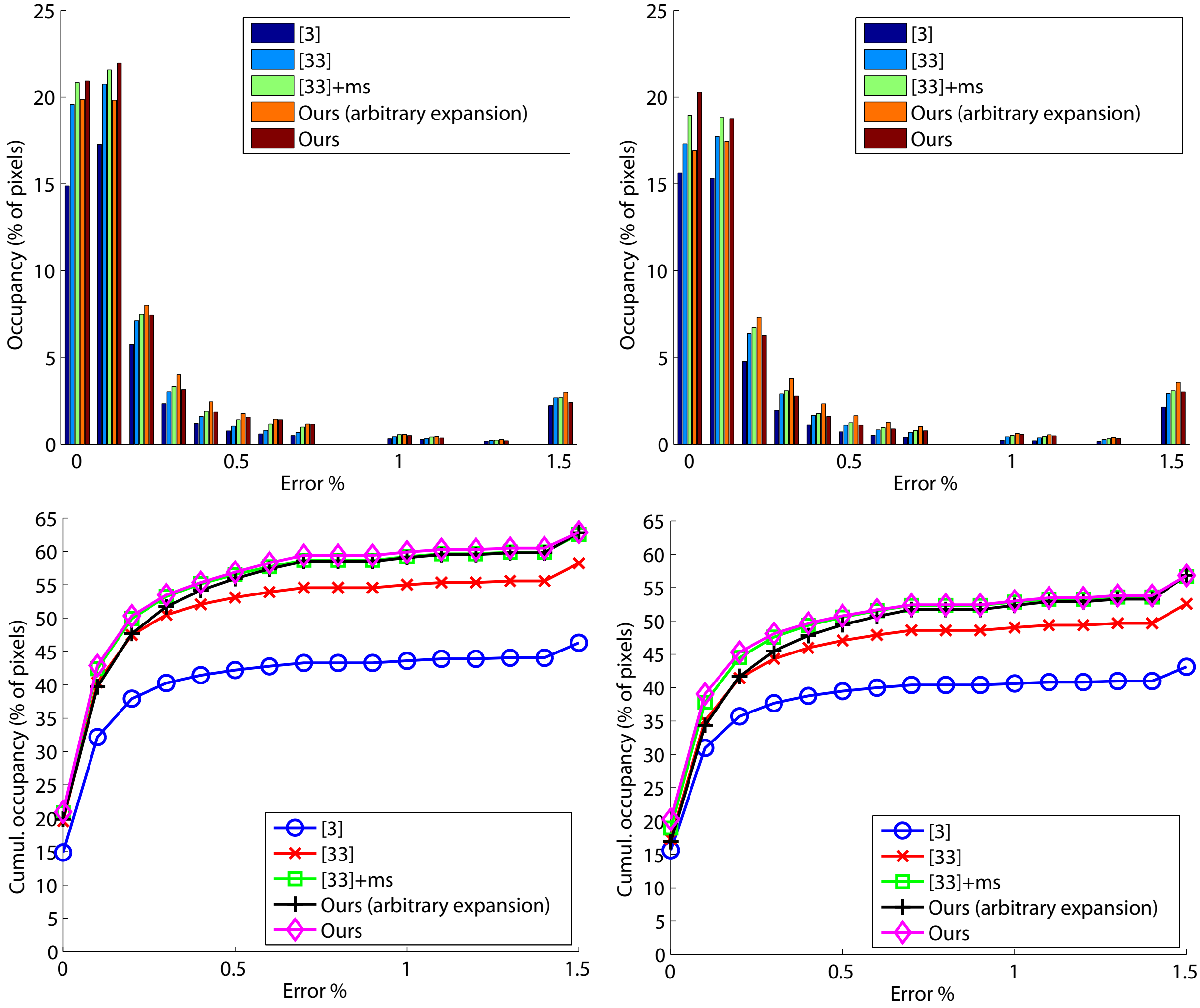

Figure 10. Error distribution (top) and cumulative error distribution (bottom) of Fountain (left) and Herz-Jesu (right) reconstructions. The proposed method produces both more accurate and denser point clouds from these datasets.

because the mesh generation and refinement stages of the state-of-the-art systems [3, 4] are not publicly available it is hard to evaluate the different components of these systems using standard benchmarks[3]. This same problem of fair evaluation was also discussed in [19] and similar to their work we make comparisons between depth maps that are obtained by projecting both the reconstructed point clouds and the ground truth mesh models onto the images.

Furthermore, in order to facilitate future research and ensure the reproducibility of our results, we will publish an open source implementation of our software upon the publication of the article. It is important to note that the software of the recent methods [19] and [4] is not freely available. In fact, the authors of these papers have established companies, Pix4D and Acute3D, which develop closed commercial solutions for image-based three-dimensional reconstruction. However, we believe that in order to allow new researchers to contribute to the field and to expedite the development, it would be important to have open source implementations of recent methods publicly available for researchers.

---

[3] Also, the benchmark [2] is no longer available and the ground truth data has been released only for those two scenes which we used in our evaluations.

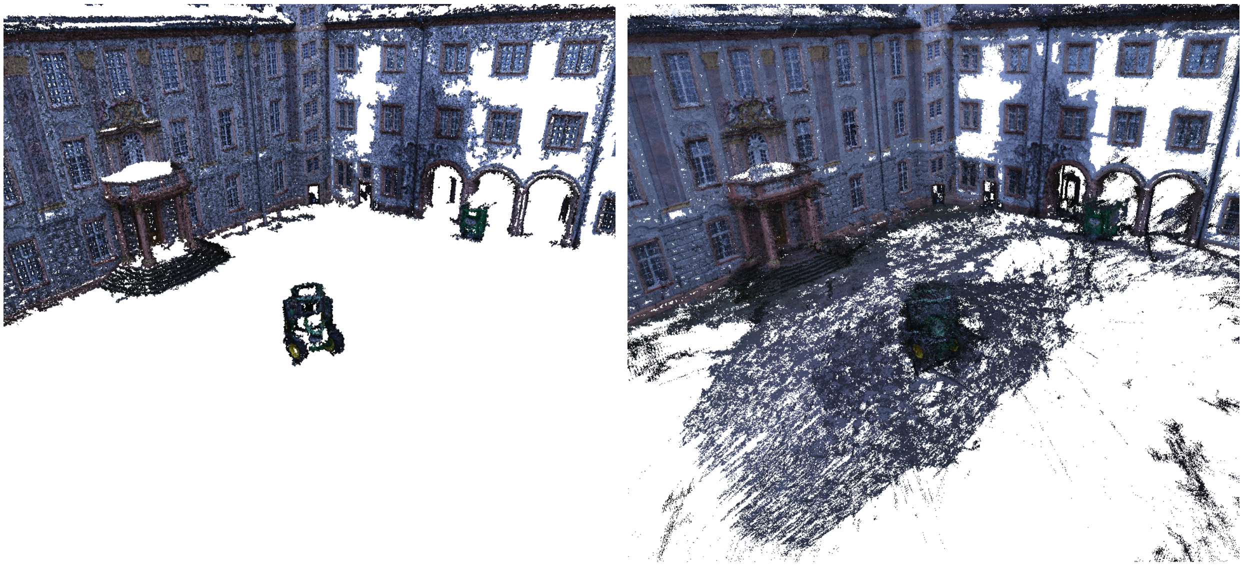

Figure 11. Castle reconstructions build by the PMVS program [3] (left) and the proposed method (right). Our reconstruction has lots of points on ground which is totally missing in the PMVS reconstruction.

6 Conclusion

In this paper, we have proposed a multi-view stereo reconstruction method based on prioritized matching. Generally, the method takes an arbitrary number of images and a sparse set of correspondences as input and iteratively expands them to neighboring regions in all images. The approach uses the best-first principle, where the most promising seed is always expanded first. During the expansion the seeds are refined using image alignment. After the first expansion, the reconstruction is improved by expanding a set of new seeds near the holes or outer boundaries of the current reconstruction. The expansion is repeated until all possible areas have been reconstructed.

Table 5. Comparison of efficiency

	Method	**Dataset**			
		Cube	Fountain-P11	Herz-Jesu-P8	Castle-P30
Number of points	[3]	382 865	6 141 328	5 175 051	10 674 253
	[33]	492 329	8 594 333	6 206 911	-
	[33] + ms	599 413	9 766 125	7 009 032	-
	Ours (Arb.exp.)	793 444	11 149 988	8 000 098	-
	Ours	585 785	9 842 320	7 148 507	17 493 563
Total time (min)	[3]	32.5	670.9	370.5	1270.8
	[33]	6.0	69.6	46.1	-
	[33] + ms	9.0	84.4	54.9	-
	Ours (Arb.exp.)	12.3	100.0	72.9	-
	Ours	8.6	91.5	61.9	260.8
Time per point (ms)	[3]	5.09	6.55	4.30	7.14
	[33]	0.73	0.49	0.45	-
	[33] + ms.	0.90	0.52	0.47	-
	Ours (Arb.exp.)	0.93	0.54	0.55	-
	Ours	0.88	0.56	0.52	0.89

The comparison to the widely used PMVS software showed that the proposed method produces denser reconstructions with better accuracy and is several times faster. In some cases it is even more than 10 times faster when the processing time is measured per reconstructed point. The method also improves the reconstruction results achieved with our baseline method.

An open source implementation of our approach will be released upon the publication of the article.

References

[1] Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: 'A comparison and evaluation of multi-view stereo reconstruction algorithms', IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006), pp. 519–528

[2] Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: 'On benchmarking camera calibration and multi-view stereo for high resolution imagery', IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008), pp. 1–8

[3] Furukawa, Y., Ponce, J.: 'Accurate, dense, and robust multi-view stereopsis', IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2010, 32 (8), pp. 1362–1376

[4] Vu, H.H., Labatut, P., Pons, J.P., Keriven, R.: 'High accuracy and visibility-consistent dense multiview stereo', IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2011, 34 (5), pp. 889–901

[5] Shan, Q., Curless, B., Furukawa, Y., Hernandez, C., Seitz, S.M.: 'Occluding contours for multi-view stereo', IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)

[6] Delaunoy, A., Pollefeys, M.: 'Photometric bundle-adjustment for dense multi-view 3d modeling', IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)

[7] Sinha, S.N., Scharstein, D., Szeliski, R.: 'High-resolution stereo matching using local plane sweeps', IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)

[8] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision (Cambridge University Press, 2000)

[9] Quan, L.: Image-Based Modeling (Springer, 2010)

[10] Sinha, S., Mordohai, P., Pollefeys, M.: 'Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh', IEEE International Conference on Computer Vision (ICCV) (2007)

[11] Vogiatzis, G., Hernández, C., Torr, P., Cipolla, R.: 'Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency', IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2007, 29 (12), pp. 2241–2246

[12] Kolev, K., Klodt, M., Brox, T., Esedoglu, S., Cremers, D.: 'Continuous global optimization in multiview 3d reconstruction', International Journal of Computer Vision (IJCV), 2009, 84 (1), pp. 80–96

[13] Kutulakos, K.N., Seitz, S.M.: 'A theory of shape by space carving', International Journal of Computer Vision (IJCV), 2000, 38 (3), pp. 199–218

[14] Seitz, S.M., Dyer, C.R.: 'Photorealistic scene reconstruction by voxel coloring', International Journal of Computer Vision (IJCV), 1999, 35 (2), pp. 151–173

[15] Goesele, M., Curless, B., Seitz, S.M.: 'Multi-view stereo revisited', IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2006)

[16] Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.M., Yang, R., *et al.*: 'Real-time visibility-based fusion of depth maps', IEEE International Conference on Computer Vision (ICCV) (2007)

[17] Zach, C., Pock, T., Bischof, H.: 'A globally optimal algorithm for robust TV-L^1 range image integration', IEEE International Conference on Computer Vision (ICCV) (2007)

[18] Li, J., Li, E., Chen, Y., Xu, L., Zhang, Y.: 'Bundled depth-map merging for multi-view stereo', IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010)

[19] Tola, E., Strecha, C., Fua, P.: 'Efficient large-scale multi-view stereo for ultra high-resolution image sets', Machine Vision and Applications, 2012, 23 (5), pp. 903–920

[20] Gales, G., Crouzil, A., Chambon, S.: 'A region-based randomized voting scheme for stereo matching', Advances in Visual Computing (2010)

[21] Niquin, C., Prévost, S., Rémion, Y.: 'An occlusion approach with consistency constraint for multiscopic depth extraction', International Journal of Digital Multimedia Broadcasting (IJDMB), 2010, pp. 1–8

[22] Prévost, S., Niquin, C., Chambon, S., Gales, G.: Multi and Stereoscopic matching, depth and disparity (John Wiley & Sons, Inc., 2013), pp. 137–156

[23] Habbecke, M., Kobbelt, L.: 'A surface-growing approach to multi-view stereo reconstruction', IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007), pp. 1–8

[24] Lhuillier, M., Quan, L.: 'A quasi-dense approach to surface reconstruction from uncalibrated images', IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2005, 27 (3), pp. 418–433

[25] Čech, J., Sára, R.: 'Efficient sampling of disparity space for fast and accurate matching', BenCOS Workshop in conjunction with CVPR (2007)

[26] Koskenkorva, P., Kannala, J., Brandt, S.S.: 'Quasi-dense wide baseline matching for three views', International Conference on Pattern Recognition (ICPR) (2010)

[27] Furukawa, Y., Curless, B., Seitz, S., Szeliski, R.: 'Towards internet-scale multi-view stereo', IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010), pp. 1434–1441

[28] Labatut, P., Pons, J.P., Keriven, R.: 'Robust and efficient surface reconstruction from range data', Computer Graphics Forum (CGF), 2009, 28 (8), pp. 2275–2290

[29] Jancosek, M., Pajdla, T.: 'Multi-view reconstruction preserving weakly-supported surfaces', CVPR (2011)

[30] Vu, H.H., Keriven, R., Labatut, P., Pons, J.P.: 'Towards high-resolution large-scale multi-view stereo', CVPR (2009)

[31] Lhuillier, M., Quan, L.: 'Match propagation for image-based modeling and rendering', IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2002, 24 (8), pp. 1140–1146

[32] Kannala, J., Brandt, S.S.: 'Quasi-dense wide baseline matching using match propagation', IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007)

[33] Ylimäki, M., Kannala, J., Holappa, J., Heikkilä, J., Brandt, S.S.: 'Robust and accurate multi-view reconstruction by prioritized matching', International Conference on Pattern Recognition (ICPR) (2012)

[34] Jancosek, M., Shekhovtsov, A., Pajdla, T.: 'Scalable multi-view stereo', 3DIM (2009)

[35] Gales, G., Chambon, S., Crouzil, A., McDonald, J.: 'Reliability measure for propagation-based stereo matching', International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) (2012)

[36] Bleyer, M., Rhemann, C., Rother, C.: 'Patchmatch stereo - stereo matching with slanted support windows', British Machine Vision Conference (BMVC) (2011)

[37] Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., *et al.*: 'Building rome in a day', Communications of the ACM, 2011

[38] Shan, Q., Adams, R., Curless, B., Furukawa, Y., Seitz, S.: 'The visual turing test for scene reconstruction', International Conference on 3D Vision (3DV) (2013)

[39] Chambon, S., Crouzil, A.: 'Similarity measures for image matching despite occlusions in stereo vision', Pattern Recognition, 2011, 44 (9), pp. 2063–2075

[40] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., *et al.*: 'A comparison of affine region detectors', International Journal of Computer Vision (IJCV), 2005, 65 (1-2), pp. 43–72

[41] Lowe, D.: 'Distinctive image features from scale invariant keypoints', International Journal of Computer Vision (IJCV), 2004, 60 (2), pp. 91–110

[42] Morel, J., Yu, G.: 'Asift: A new framework for fully affine invariant image comparison', SIAM Journal on Imaging Sciences, 2009, 2 (2), pp. 438–469

[43] Baker, S., Matthews, I.: 'Lucas-kanade 20 years on: A unifying framework', International Journal of Computer Vision (IJCV), 2004, 56 (3), pp. 221–255