MARMVS: Matching Ambiguity Reduced Multiple View Stereo for Efficient Large Scale Scene Reconstruction

Zhenyu Xu,Yiguang Liu*, Xuelei Shi, Ying Wang, Yunan Zheng
College of Computer Science, Sichuan Uiversity, China
sanxu@outlook.com,liuyg@scu.edu.cn,sxlsnow@outlook.com,yingwang@stu.scu.edu.cn,1121955210@qq.com

Abstract

The ambiguity in image matching is one of main factors decreasing the quality of the 3D model reconstructed by PatchMatch based multiple view stereo. In this paper, we present a novel method, matching ambiguity reduced multiple view stereo (MARMVS) to address this issue. The MARMVS handles the ambiguity in image matching process with three newly proposed strategies: 1) The matching ambiguity is measured by the differential geometry property of image surface with epipolar constraint, which is used as a critical criterion for optimal scale selection of every single pixel with corresponding neighbouring images. 2) The depth of every pixel is initialized to be more close to the true depth by utilizing the depths of its surrounding sparse feature points, which yields faster convergency speed in the following PatchMatch stereo and alleviates the ambiguity introduced by self similar structures of the image. 3) In the last propagation of the PatchMatch stereo, higher priorities are given to those planes with the related 2D image patch possesses less ambiguity, this strategy further propagates a correctly reconstructed surface to raw texture regions. In addition, the proposed method is very efficient even running on consumer grade CPUs, due to proper parameterization and discretization in the depth map computation step. The MARMVS is validated on public benchmarks, and experimental results demonstrate competing performance against the state of the art.

1. Introduction

Multi-view stereo (MVS) is a hot research area in computer vision, which offers a cheap and convenient way to capture the 3D geometry of scenes and objects, and serves as a main ingredient for many CV applications, *e.g.* [40, 26, 25, 24, 8]. A lot of MVS methods with increasingly high performance have been proposed in last decades, thanks to the public available benchmarks [31, 34, 1, 14]. Due to the successes of SfM algorithms [28, 38, 33] that camera pose can be well estimated, the 3D reconstruction from multiple images can be viewed as a problem of dense matching across images. The PatchMatch method [4] is a powerful tool to solve the dense matching problem with high accuracy and efficiency. Built on the PatchMatch schema, several methods rank at the top of recent challenging benchmark [30]. However, the PatchMath method does not perform well when matching ambiguity occurs, in such situation many hypotheses can generate high matching scores, and leaving uncertainty for better depth and normal choosing.

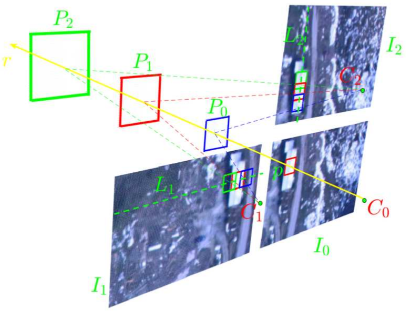


Figure 1. The matching ambiguity can be reduced by proper scale and neighbouring image selection. Finding the optimal corresponding match for the patch p in reference image I_0, if image I_2 is chosen as the neighbour image of p, it will be matched to several ambiguous patches along the epipolar line L_2, however an unambiguous patch will be matched if image I_1 is chosen. On the other hand, p is ambiguous on both images in smaller scale.

In this paper, we propose a new MVS method to handle the matching ambiguity issue, and consequently, the accuracy and completeness are increased. Firstly, for a given pixel the matching ambiguity can be possibly reduced by carefully selection of pixel's scale and the neighbouring images. The motivation behind is that the matching ambiguity can be analysed using the geometry of image surface (the height is the pixel gray-scale intensity), and it varies with different directions and scales, see Fig. 1. The matching stability measured by surface normal curvature, is computed in different scales and directions guided by epipolar constraint, which is used as one of important factors for the pixel level scale selection. Secondly, by initializing the depth value in a smaller range using the depth information of the surrounding feature points of each pixel, it suppresses the ambiguity introduced by repeated image texture and also accelerates the speed of depth map convergency. And thirdly, the ambiguity is further reduced by giving higher priority for planes with higher matching stability in the following propagation step that propagating correctly reconstructed surfaces to raw texture regions.

For large scale scene reconstruction with high resolution images, the efficiency is quite an important factor to be taken into account. Many methods become impractical due to heavy computation burden and huge memory requirement. We propose an efficient way to handle the most time consuming part, the PatchMatch stereo for depth map computation. By proper parametrization and parameter discretization of the homography, the homography matrices mapping an image patch on the reference image to a warped patch on the neighbour image can be pre-computed and stored. This is time saving especially for high resolution images, since now we do not need to compute the homography mapping at every single pixel. This strategy gains a speed up by a factor of more than 10, as a result the proposed method is very efficient even implemented on consumer-grade CPUs.

In summary, the main contributions of this paper are in three folds:

- A new method for optimal pixel level scale selection is proposed by the analysis of image surface using differential geometry and epipolar geometry.

- A PatchMatch based method for depth map computation is proposed, by using the proposed matching stability as a priority, it alleviates the ambiguity introduced by image raw texture regions.

- An efficient strategy is proposed for depth map computation, which increases the computational speed by 10 times.

2. Related Work

According to Ref. [31], MVS methods can be sorted into four classes depend on how the 3D scenes are formulated, voxel based methods, surface evolution based methods, feature point growing based methods, and depth-map merging based methods. The proposed method belongs to the last class, which takes the output of common SfM softwares [28, 38, 33] as input and produces dense point cloud by merging individual depth maps. A typical depth map merging based MVS method generally follows the pipe line of image selection, stereo matching, depth map filtering and fusion. In each step, it may varies from one to another.

Computing the depth map of a given reference image is generally by performing a dense stereo matching between reference image and neighbouring images. Nowadays, the most popular method for dense matching must be the PatchMatch algorithm, based on which, many MVS methods rank at the top of several benchmarks [31, 34, 1, 30], even with some deep learning based methods emerged in recent years [43, 36, 17, 23, 42]. The PatchMatch algorithm is primarily designed to compute the nearest neighbour field between images [4], which shows great power in efficiently solving pixel labeling problems when the label set is very large. Bleyer *et al.* [7] introduce this method into bilocular stereo with impressive results obtained, since with the PatchMatch schema the disparity and normal can be modeled in continuous space and solved efficiently. Shen [32] extends this frame work into multi-view stereo for depth map computation, then merges the depth maps into a single point cloud. Many other works [45, 13, 29] with impressive results have been proposed by more sophisticated image choosing or depth propagation.

Despite the success of these methods, they share the same drawback inherited from the PatchMatch algorithm, when compared with some other excellent works [18, 19]. Since the original PatchMatch method only has a data term, which is unable to handle the matching uncertainty in large raw texture regions. Several multi-scale approaches [37, 39, 41] have been proposed to deal with the large raw texture regions by down-sampling the source images typically into 3 scales and merging the depth images computed in these scales. This problem can also be fixed in some extent by using super pixel approaches [27, 20], that enforce the pixel within the same super pixel share the same pixel label. However, this can bring some artifacts, since the supper pixels on 2D image do not always consist with the geometry in 3D space, such as curved surfaces. Compared to the previous image-level multi-scale approaches, our method choose the optimal scale for each pixel. As the multi-scale approach in our method is not done by image down-sampling, thus it is able to cover more dense scales and it does not surfer from the inaccuracy introduced by image down-sampling. Moreover, as the scale for every pixel is determined before depth map computation, that for each pixel we only compute the depth in the optimal scale, which is more time saving than computing the depth in several scales and merging them into a single one.

The computational efficiency is quite an important feature for MVS method, since the images captured by devices nowadays are with increasingly higher resolution. So, GPU compatible methods [2, 3, 45] have been proposed by modifying the sequential propagation direction of the original PatchMatch stereo algorithm. To further utilize the capabilities of GPUs, Galliani *et al.*[13] propose a diffusion-like propagation schema, and which is improved by Xu and

Tao [41] by sampling more candidates in each group of the checkerboard pattern and only propagating those with low matching cost. Different to these methods accelerating the computation speed by using more powerful hardware devices, our method is friendly to those with lower computational capability, and efficient even on consumer grade cpus.

3. MARMVS: The Method

Our method is detailed in 3 parts. In Section 3.1, the ambiguity in image matching is analysed with the measures to compute the similarity between two image patches, and we also propose corresponding method to reduce this ambiguity by proper scale and neighbour image selection for every single pixel. In Section 3.2, the pipe line to compute the depth map is given, besides conventional techniques, strategies to reduce the ambiguity are given in detail. In addition, we also design a solution to efficiently compute the depth maps. In Section 3.3, we briefly talk about our method for depth map filtering and fusion, for every 3D point the consistency is measured across views and between its neighbour pixels, which is a flexible strategy for controlling the trade off between accuracy and completeness for the reconstructed model, especially for those parts only visible in two views.

3.1. Pixel Level Scale Selection

It is routinely required to compute the depth map of every reference image for depth map based MVS methods. To compute the depth of a given pixel x in the reference image, similar to existing methods, we follow the typical procedure of PatchMatch stereo. The space propagation is performed after random depth and random normal initialization, and the propagation is also performed after depth map refinement via random optimization. In the propagation and optimization steps, a 3D plane will replace the current plane if it is supposed to be a more accurate approximation of the corresponding local object surface, which is determined by whether a higher matching score can be generated by the new 3D plane. Based on the epipolar geometry, the former operation is equivalent to find the optimal corresponding match along the epipolar line. Denote $\boldsymbol{p}$ a patch centered at x, in raw texture and repetitive texture regions it occasionally happens that several candidate patches in the neighbouring image all posses high similarity scores to the patch $\boldsymbol{p}$, which leaves uncertainty in optimal depth decision. But the location of the correct patch $\boldsymbol{p}^{'}$ in the neighbour image is prior unknown, it is impracticable to check whether $\boldsymbol{p}$ is similar to the patches along the epiploar line close to $\boldsymbol{p}^{'}$. However, based on the epipolar constraint, we can check that if the patch $\boldsymbol{p}$ is similar to its own surrounding patches on the epipolar line related to $\boldsymbol{p}^{'}$.

The similarity between two image patches can be measured by many methods, such as NCC [44, 16, 21], SSD [9, 11, 10], or key-point descriptors [35]. Among them, the methods possessing the ability of being robust for small image patch matching and being invariant to different light conditions are preferred. As the invariance for different light conditions is usually required, that high similarity score can be obtained between two image patches if they are approximately linear dependent. Let $p(X, \sigma)$ be an image patch with X the center and σ the scale which is proportional to the window size. Let $L(X, I_n)$ be the epipolar line on the reference image passing through X, and I_n is corresponding neighbouring image. We can turn the problem of reducing the matching ambiguity at pixel X to searching a proper scale σ and neighbour image I_n, so that the patch $p(X, \sigma)$ is less linear correlated to its surrounding patches along the epipolar line $L(X, I_n)$. This substantially requires the patch $p(X, \sigma)$ curved in the epipolar line direction, if we view this patch as a part of image surface (the intensity is the height) in corresponding image scale. It is naturally to use *normal curvature* to estimate how the image surface is curved in a specific epipolar line direction.

Based on the epipolar geometry, all the epipolar lines on the reference image I_r corresponding to the neighbouring image I_n intersect at the epipole $e(I_r, I_n)$, thus the direction of epipolar line $L(X, I_n)$ passing through X and corresponding to neighbour image I_n can be determined by X and $e(I_r, I_n)$. According to [15], homogeneous coordinates of the epipole $e = (x, y, z)$ is the right null-vector of the fundamental matrix F:

$$Fe = 0. \tag{1}$$

Since F is a 3×3 matrix and has rank 2, there only exists one group of linearly correlated solutions, thus the position of the epipole $e(I_r, I_n)$ is fixed and can be computed by solving Eq. 1. The fundamental matrix F defined from image I to $I^{'}$ can be computed with the internal and external camera parameters obtained from SfM method:

$$F = [P^{'}C]_{\times} P^{'} P^{+}, \tag{2}$$

where P^{+} is the pseudo-inverse of P, C is the camera center related to I and $[\cdot]_{\times}$ defines the skew-symmetric matrix corresponding to cross product.

After the direction of epipolar line through each pixel is computed, the corresponding normal curvature in different image scales can be derived from differential geometry [12] and scale space theory [22].

Let $I(x, y)$ represent pixel intensity at (x, y), and write the image surface in form of vector equation:

$$\boldsymbol{r}(x, y) = (x, y, I(x, y)), \tag{3}$$

Taking first and second partial derivatives on Eq. 3, we have $\boldsymbol{r}_x = (1, 0, I_x)$, $\boldsymbol{r}_y = (0, 1, I_y)$, $\boldsymbol{r}_{xx} = (0, 0, I_{xx})$, $\boldsymbol{r}_{yy} = (0, 0, I_{yy})$, $\boldsymbol{r}_{xy} = (0, 0, I_{xy})$. The surface normal at (x, y) is $\boldsymbol{n} = \frac{\boldsymbol{r}_x \times \boldsymbol{r}_y}{||\boldsymbol{r}_x \times \boldsymbol{r}_y||}$. With these derivatives and the surface normal, the coefficient matrix of first and second fundamental form can be computed:

$$I: \begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_x \cdot \boldsymbol{r}_x & \boldsymbol{r}_x \cdot \boldsymbol{r}_y \\ \boldsymbol{r}_x \cdot \boldsymbol{r}_y & \boldsymbol{r}_y \cdot \boldsymbol{r}_y \end{bmatrix} = \begin{bmatrix} 1 + I_x^2 & I_x I_y \\ I_x I_y & 1 + I_y^2 \end{bmatrix}$$

$$II: \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_{xx} \cdot \boldsymbol{n} & \boldsymbol{r}_{xy} \cdot \boldsymbol{n} \\ \boldsymbol{r}_{xy} \cdot \boldsymbol{n} & \boldsymbol{r}_{yy} \cdot \boldsymbol{n} \end{bmatrix} = \frac{\begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}}{\sqrt{1 + I_x^2 + I_y^2}} \tag{4}$$

Denote $\vec{n} = (u, v)$ a unit vector in the direction of the epipolar line $L(X, I_n)$, denote (a, b) the coordinate of X on reference image, the epipolar line can be represented by the parametric equations:

$$x(t) = ut + a \tag{5}$$

$$y(t) = vt + b \tag{6}$$

Combining the Eqs. 3, 5, and 6, a curve on the image surface is defined $\boldsymbol{r}(x(t), y(t))$, whose orthogonal projection on the image plane is the epipolar line $L(X, I_n)$. The direction of the tangent line of the curve $\boldsymbol{r}(x(t), y(t))$ is $\omega = u\boldsymbol{r}_x + v\boldsymbol{r}_y$. Since $\boldsymbol{r}_x$ and $\boldsymbol{r}_y$ are base vectors of the tangent plane of the image surface at X, denote $\delta = [u, v]$, the normal curvature at X with direction ω is

$$k(X, \omega) = \frac{\delta \begin{bmatrix} L & M \\ M & N \end{bmatrix} \delta^T}{\delta \begin{bmatrix} E & F \\ F & G \end{bmatrix} \delta^T}. \tag{7}$$

Note, in Eq. 7 the image scale is ignored, but we need to compute the normal curvatures in different scale to determine the matching window size of each pixels. Actually, this can be done by calculating the first and second derivatives of the image in different scale, then the first and second fundamental form are obtained via Eqs.4 and the normal curvature can be computed by Eq. 7. For computational efficiency, instead of computing the derivatives after convoluting the image with different gaussian kernel, it could be done by convolution of the origin image with the derivatives of the gaussian kernel based on the property that:

$$\frac{\partial^{i+j}}{\partial x^i \partial y^j}[G(x, y, \sigma) * I(x, y)] = I(x, y) * \frac{\partial^{i+j}}{\partial x^i \partial y^j} G(x, y, \sigma), \tag{8}$$

where $G(x, y, \sigma)$ is the gaussian kernel with σ the scale, and $*$ represents the convolution operator. Ref. [6] presents an efficient technique to approximate the first and second order Gaussian derivatives by using box filters and integral images, which is a contributory factor for the efficiency of the benchmark detector and descriptor. With this computation strategy, the time used for normal curvature computation is negligible, for example, the normal curvature is able to be calculated in 10 scales within 1 second for an image with resolution of 3072 × 2048.

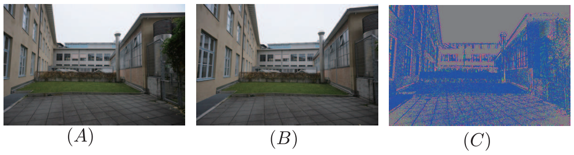

(A) (B) (C)

Figure 2. Illustration of scale selection for each pixel in the reference image. Image (A) is the reference, image (B) is a neighbouring image of (A). The results are drawn in image (C), the color of each pixel represents a specific selected window size ranging from 7×7 to 63×63, where for the smallest window size are depicted in blue and the biggest are depicted in gray.

Table 1. The side length of the window for stereo matching in our method.

l_0	l_1	l_2	...	l_n
7	11	15	...	4n+7

It is known that the smaller scale we choose the better local detail is preserved, on the other hand, the larger scale often yields increased completeness but over smoothed local structure. Hence, to the purpose of preserving fine details of rich texture regions and increasing completeness of raw texture regions, for each pixel in the reference image, the size of the matching window is set according to the smallest scale available to generate a stable match. This is achieved by searching the normal curvature from the first scale to the largest scale until it reaches a threshold T (T=0.01 and the pixel intensity is within [0,1] in our method). The sizes of matching windows in our method are ranged from $l_0 \times l_0$ to $l_n \times l_n$, as shown in table 1, l_0 defines the smallest matching window which is set to 7 and is fixed in our method, l_n may varies depends on the scene to be reconstructed. A typical example of pixel level scale selection is illustrated in Fig. 2.

3.2. Depth Map Computation

The depth maps in our method are computed following the procedure of existing PatchMatch stereo based MVS methods, the differences are: 1)The depth range for every single pixel in a reference image is initialized randomly within smaller range determined by depth information of surrounding feature points output by SFM method, leading to faster depth convergence. 2)The window size related to a neighbour image is chosen for every individual pixels based on the criteria introduced in section 3.1, thus the proposed method can preserve fine structure in rich texture regions while increase the completeness in raw texture regions. 3)In the final propagation, the normal curvature is taken in-

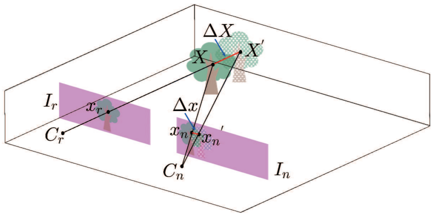


Figure 3. Suppose we require the reconstructed error of pixel x_r to be less than ΔX, we should select a desired neighbouring image for x_r, on which the projection Δx of ΔX should be larger than 0.5 pixel.

to consideration, which further propagates an unambiguous plane to raw texture region with even higher completeness achieved.

3.2.1 Neighbouring Images Selection

To compute the depth map of a reference image, several neighbour images should be selected for stereo matching. The two principle rules of existing methods for neighbour image selection are: 1) a good neighbour image should have enough overlap area with the reference image; 2) there should exist enough base line to generate a stable triangulation, or it will affect the accuracy of the reconstructed model. Based on the first rule, for a given image, it will be chosen as a neighbouring image if it shares sufficient feature points with the reference image. And for the second rule, instead of computing the average triangulation angle of the feature point, we give a more accurate measurement. Suppose the relative error of the reconstructed 3D model is required to be less than ϵ ($\epsilon = 0.001$ in our method). Let X be a 3D point from SfM, and move X a bit to $X^{'}$ along the ray passing through camera center C of reference image I_r, then we have: $X^{'} = (1+\epsilon)X - \epsilon C_r$. Denote $x_n^{'}$ and x_n as the projections of $X^{'}$ and X on the neighbouring image I_n respectively, A is the point set containing the common 3D points of the reference image and the neighbour image. The threshold τ rejecting the candidates above is computed as:

$$\tau = \frac{1}{|A|} \sum_{X \in A} \left|\left| x_n^{'} - x_n \right|\right| . \tag{9}$$

The neighbouring image candidates with $\tau < 0.5$ are removed, and in the remaining images, we keep top 8 images ranked by the number of common 3D points as neighbouring images.

3.2.2 Parameters Initialization

In PatchMatch based stereo matching methods, it usually assigns a random depth d and a random normal vector $\vec{n}$ for every pixel in the reference image, then the stereo matching is turned out to be an optimization problem of finding d and $\vec{n}$ that minimizing the matching cost between reference image and its neighbouring images. In our method, besides the parameter d and $\vec{n}$, a random number i is also assigned to every pixel in the reference image, which represents the ID of the selected neighbouring image for stereo matching. More precisely, we assign a parameter set U with four random parameters to each pixel $p = (a,\ b)$ in the reference image:

$$U := \{d,\ \varphi,\ \theta,\ i\}, \tag{10}$$

where d is the depth of pixel, φ and θ define the normal of a plane, i is the neighbouring image ID. Let $V = [x, y, z]$, the first three parameters in Eq. 10 actually define a plane π in the camera coordinate system of the reference image:

$$\vec{n}V^T + D = 0, \tag{11}$$

where $\vec{n} = [sin(\varphi)cos(\theta), sin(\varphi)sin(\theta), cos(\varphi)]$, and $D = -d\vec{n}K_r^{-1}[a\ b\ 1]^T$. To get uniformly distributed random normal vectors, we pick two random variates $u \in [0, 1)$ and $v \in [0.5, 1)$, then

$$\begin{cases} \varphi = cos^{-1}(2v-1) \\ \theta = 2\pi u \end{cases} \tag{12}$$

is used to generate normal vectors which are uniformly distributed on hemisphere surface.

As for the parameter d, different to existing method that assign a random value between the depth range of the reference image, we shrink this range to the max and min depth of four closest feature points in four regions, as depicted in Fig.4. It is very efficient to find the most closest feature point in an area by using the method in chamfer matching [5], which calculates the shortest distance for every non-edge points to the edge points. With dynamical programming, it only needs to scan the image once to find the closest feature point for all non-feature points in a specific area.

The last parameter i for every pixel is initialized with a ID number randomly chosen from the neighbour images candidates which are selected by the method in Section 3.2.1.

3.2.3 Plane Propagation and Refinement

After the random initialization of the parameters, plane propagation and refinement are performed alternately to obtain the final depth map. In both steps, for every pixel, a new plane will replace current plane, if it is able to generate a higher score:

$$S = ((1-\alpha)S_{pho} + \alpha S_{sta})S_{tri}, \tag{13}$$

where S_{pho} is the photometric consistency estimated by Zero-mean Normalized Cross correlation (ZNCC), S_{sta}

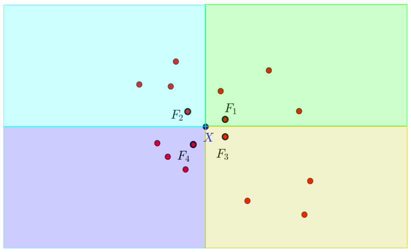


Figure 4. The depth value of point X are randomly assigned in the range determined by the depth of the four most closest feature points F_1, F_2, F_3, and F_4 in four region respectively.

represents the matching stability of current patch, which is measured by the normal curvature k and defined as $S_{sta} = min(e^{\frac{k}{0.01}-1}, 1)$, S_{tri} is used to reject those unable to generate a stable triangulation:

$$S_{tri} = \begin{cases} 1 \ \ if \ \Delta x > 0.5 \\ 0 \ \ otherwise \end{cases}, \tag{14}$$

in which, Δx represents the projection of the displacement of the 3D point, as depicted in Fig.3, and it is computed as:

$$\Delta x = \left| \overrightarrow{C_n X} - \frac{\overrightarrow{C_n X} \cdot \vec{P}_{axis}}{\overrightarrow{C_n X'} \cdot \vec{P}_{axis}} \overrightarrow{C_n X'} \right| \frac{f_x + f_y}{2\overrightarrow{C_n X} \cdot \vec{P}_{axis}}. \tag{15}$$

The parameter α controls the influence of the matching stability. In the first two propagations and refinements we set $\alpha = 0$. In the last propagation, we set $\alpha = 0.1$ which is an empirical value from experiments, it is able to propagate correctly reconstructed surfaces to raw texture regions, and also it avoids propagating them into surface discontinue area.

3.2.4 Efficient Implementation

For PatchMatch based depth map computation, the most time consuming part is the calculation of the homograph matrix mapping of the image patches from the reference image to its neighbouring images, as it is conducted on every pixel with generally more than ten times. Denote $[a, b, 1]$ the homogeneous coordinates of a pixel with normal $\vec{n}$ and depth d, according to [32, 29], the homography matrix induced by the plane defined in Eq. 11 is computed as:

$$\begin{aligned} H =& K_{I'}(R_{I'}R_I^{-1} + \frac{R_{I'}(C_{I'} - C_I)\vec{n}}{D})K_I^{-1} \\ =& K_{I'}R_{I'}R_I^{-1}K_I^{-1} + K_{I'}R_{I'}(C_{I'} - C_I)\vec{n}K_I^{-1} \\ & \frac{1}{-d\vec{n}K_I^{-1}[a\ b\ 1]^T}. \end{aligned} \tag{16}$$

Note, for a pair of images I and I', the term $K_{I'}R_{I'}R_I^{-1}K_I^{-1}$ is a constant matrix for every pixel in I, the term $K_{I'}R_{I'}(C_{I'} - C_I)\vec{n}K_I^{-1}$ only depends on the surface normal $\vec{n}$, and the last term depends on the coordinates of specific pixel, the surface normal $\vec{n}$ and a depth value. For computational efficiency, the first term is pre-computed and cached as M_1. For the second term, since at every pixel, the $\vec{n}$ is randomly sampled on the half spherical surface of a unit sphere, if we make a uniform discrete sampling of the hemisphere surface densely enough, then the $\vec{n}$ can be sampled in a discrete vector set N rather than on the continuous spherical surface. In our implementation, the set N contains about 0.5 million vectors. This is achieved by uniform sampling of $u \in [0, 1)$ and $v \in [0.5, 1)$ with the sampling step length set to 0.001, then we get φ and θ by Eq. 12, each pair of φ and θ defines a vector on the sphere surface. For each $\vec{n} \in S$, we compute and store the second term in Eq. 16, then we get a set of matrices $M_2(\vec{n})$. The $\vec{n}K_I^{-1}$ in the third term of Eq. 16 can also be computed and stored as $M_3(\vec{n})$. The Eq. 16 is rewritten as

$$H = M_1 + \frac{M_2(\vec{n})}{-dM_3(\vec{n})[a\ b\ 1]^T}, \tag{17}$$

where M_1 and $M_2(\vec{n})$ are 3×3 matrices, $M_3(\vec{n})$ is 1×3 matrix, and they are all pre-computed. Compared to Eq. 16 the computational complexity is largely reduced.

3.3. Depth Map Fusion

The aim of depth map fusion is to merge all the depth maps into a single point cloud while filtering out the incorrect and redundant points by consistency checking. In our method, the consistency is checked not only crossing different views, but also with the neighbouring pixels, which is based on the idea that local object surface is continuous and can be viewed in more than one view. Suppose the tangent plane is defined by a 3D point and a normal, the discrepancy between two tangent planes is computed as:

$$d = \frac{1}{2}\left(\left|(X - X') \cdot \vec{n}\right| + \left|(X - X') \cdot \vec{n'}\right|\right). \tag{18}$$

In our method, two tangent planes are consistent only if d smaller than the error related to the distance to the camera center $e|X - C_r|$, where e is the relative error. To check the consistency at pixel x_r in a depth map, we first compute the discrepancy to its 8 neighbouring pixels, and similar operation is done on its neighbouring images, as illustrated in Fig. 5. If there are no less than N_1 pixels of the neighbouring image consisting with x_r, we say this neighbouring image is consistent with x_r. Only there are N_2 neighbouring images and N_3 neighbouring pixels of x_r passing the consistency checking, the related 3D points and the normals are averaged and saved to the point cloud. The set of thresholds $\Omega := (e, N_1, N_2, N_3)$ are used to balance the accuracy and completeness of the reconstructed model.

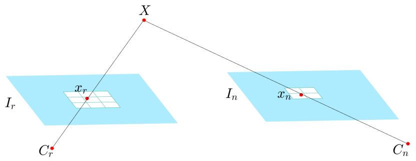


Figure 5. Illustration of consistency checking between neighbouring pixels and across neighbouring images. The consistency checking is first conducted on x_r and its eight neighbouring pixels. And then, since the coordinates of x_n are floating point numbers, so there are 4 closest pixels to x_n, then the consistency checking is done between x_r and the 4 pixels on the neighbouring image.

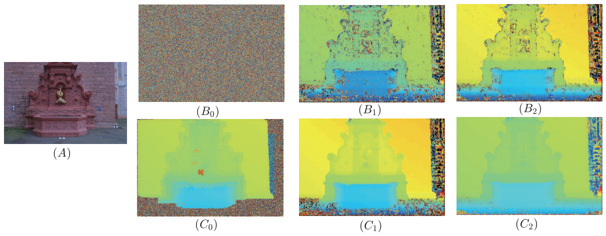


Figure 6. Depth maps after initialization, first propagation, first refinement and second propagation by using (C_0, C_1, C_2) and without using (B_0, B_1, B_2) the proposed depth initial method.

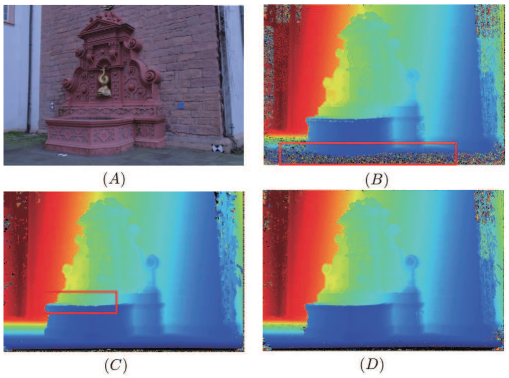


Figure 7. Depth maps corresponding to the matching scale set to minima (B), maxima (C), and adaptively by the proposed method (D).

4. Experimental Results and Discussion

Experiments are carried out on a PC with Intel Core i7 9700k CPU and 32G RAM. The proposed method is implemented in C++ and executed in 8 parallel threads on CPU. Qualitative and quantitative evaluation are performed on ETH3D benchmark [30] and Strecha benchmark [35]. We first test the effectiveness of each critical individual part of our method, and then the overall evaluation is done by comparing the proposed method with the state of the art. The criteria proposed in [30] are used for evaluation: the completeness, the accuracy and the F_1 score which is harmonic mean of completeness and accuracy.

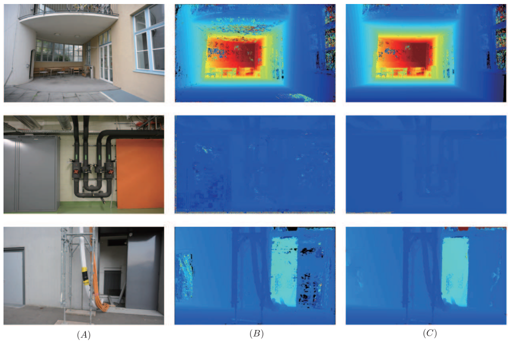


Figure 8. Depth maps obtained before (column (B)) and after (column (C)) the proposed propagation schema.

4.1. Evaluation of Individual Parts

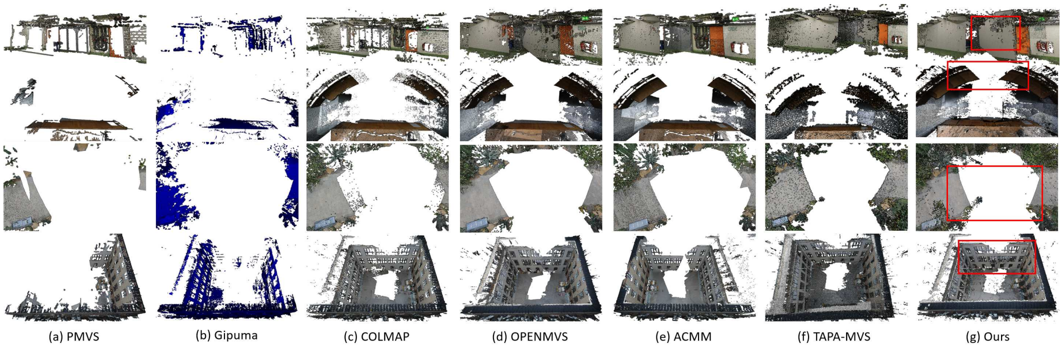
We have tested three individual parts of the proposed method, the pixel level scale selection, the depth initialization, and the strategy of plane propagation, which are supposed to be the core contributions of our method. To evaluate one part of our method, parameters of all other parts are set to defaults.

4.1.1 Pixel Level Scale Selection

The choosing of proper scale in image matching affects the balance between accuracy and completeness. To test the effectiveness of the pixel level scale selection, we compared our results obtained with the results by setting the scale to minima and maxima respectively. As the qualitative comparison in Fig. 7, when the scale is set to minima, it preserves the thin structure of the object but also brings many noises in large textureless regions, on the other hand, when the scale is set to maxima, it removes the noises in textureless regions but also over blurs the object surfaces and edges. It is apparent that the proposed pixel level scale selection preserves the thin structure well and also reduces the noise in textureless regions. Quantitative comparison is given in table 2, the $MaxScale$ and $MinScale$ denote the results obtained by setting the matching scale to maxima and minima respectively. They both suffers from inferior accuracy and completeness.

4.1.2 Pixel Level Depth Initialization

The method for depth value initialization proposed in Section 3.2.2 has the ability of accelerating convergency speed of the depth image and reducing the matching ambiguity introduced by repetitive image structure. For visual inspection, as shown in Fig. 6, a quite clean depth image is obtained by our method only after two times of propagations and refinements. On the other hand, when the depth value



(a) PMVS (b) Gipuma (c) COLMAP (d) OPENMVS (e) ACMM (f) TAPA-MVS (g) Ours

Figure 9. Comparison to state of the arts.

Table 2. Results (F_1 score, accuracy, and completeness) on the ETH3D high-resolution multi-view training datasets by removing or replacing individual parts of our method.

Method	1cm			2cm			5cm		
	F1	A	C	F1	A	C	F1	A	C
MARMVS	**68.04**	70.00	**66.85**	**79.21**	**81.98**	**77.19**	**88.39**	91.86	**85.65**
MaxScale	63.84	60.54	66.16	72.83	69.39	76.52	81.31	78.76	83.96
MinScale	56.29	69.16	47.19	65.36	81.54	54.25	72.79	**91.92**	60.31
WithoutI	62.05	63.36	59.74	70.97	73.29	69.56	79.34	82.30	76.81
WithoutP	60.54	**70.31**	53.12	69.93	81.77	60.98	78.29	90.76	68.87

is randomly initialized in the range of min and max depth of the reference image, the computed depth image is rather noisy, especially in the areas that possess raw texture or symmetrical structure. Quantitative comparison is given in table 2, our method without using the depth initialization strategy is denoted as $WithoutI$, it can be seen that the accuracy is decreased.

4.1.3 Plane Propagation

The plane propagation is aiming at propagating a better plane to neighbouring pixels, but differently from traditional methods, we combine the matching stability and matching score to evaluate whether a plane is better than the current one. As illustrated in Fig. 8, many false depth values are corrected by propagating stable matched planes to texureless area. A quantitative comparison is given in table 2, our method without using the propagation strategy is denoted as $WithoutP$, we can see that without this strategy the completeness is reduced.

4.1.4 Runtime Evaluation

In our experiment, for a reference image whose resolution is 6000 × 4000 pixels, it usually takes about 500 seconds to compute the depth map in a single CPU thread running at 4.0GHZ. Though this is not as fast as some GPU based method [41, 13], but it offers an option for devices with low computational power.

w

Table 3. F_1 score, accuracy, and completeness, on the ETH3D high-resolution multi-view test dataset, the best results are marked in bold.

Method	1cm			2cm			5cm		
	F1	A	C	F1	A	C	F1	A	C
PMVS	36.22	81.70	25.07	44.16	90.08	31.84	52.22	94.97	39.19
Gipuma [13]	34.77	69.55	27.47	45.18	84.44	34.91	57.99	95.31	45.11
COLMAP [29]	61.27	**83.75**	50.90	73.01	**91.97**	62.98	83.96	**96.75**	75.74
OPENMVS	60.03	63.23	59.20	70.56	77.77	78.54	88.74	92.27	73.02
ACMM [41]	70.80	77.50	64.35	80.78	90.65	74.34	89.14	96.30	83.72
TAPA-MVS [27]	66.60	75.16	61.82	79.15	85.71	74.94	88.16	92.49	85.02
MARMVS	**71.51**	67.72	**76.76**	**81.84**	80.24	**84.18**	**90.30**	90.32	**90.63**

4.2. Overall Evaluation

Typical results on the ETH3D dataset is depicted in Fig. 9, our method possesses the highest completeness, and from table 3, we can see that it also has the highest F_1 score among the state-of-the-art methods.

5. Conclusion

In this paper, we have presented a novel MVS system named MARMVS that efficiently handles the ambiguity in image matching process. Without using any prior knowledge of the scene and without using powerful computing devices, the MARMVS produces competing results against the state of the art on large scene reconstruction with high resolution images, which offers a compelling choice for those with low computational capability platforms. In future works, we plan to improve the quality of the reconstructed model by learned prior knowledge of surface geometry information.

Acknowledgments

Thanks to the founders of ETH3D.net for providing the evaluation datasets, and to Editor and Reviewers for time and effort spent in handling this paper. This work is supported by NSFC under Grant U19A2071 and 61860206007.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016.

[2] Christian Bailer, Manuel Finckh, and Hendrik PA Lensch. Scale robust multi view stereo. In *European Conference on Computer Vision*, pages 398–411. Springer, 2012.

[3] Linchao Bao, Qingxiong Yang, and Hailin Jin. Fast edge-preserving patchmatch for large displacement optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3534–3541, 2014.

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, volume 28, page 24. ACM, 2009.

[5] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. Technical report, SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER, 1977.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[7] M Bleyer, C Rhemann, and C Rother. Patchmatch-stereo matching with slanted support windows. In *British Machine Vision Conference, pp.–11*, 2011.

[8] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):1052–1067, 2007.

[9] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1486–1493, 2014.

[10] Amaël Delaunoy and Emmanuel Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility. *International journal of computer vision*, 95(2):100–123, 2011.

[11] Amaël Delaunoy, Emmanuel Prados, Pau Gargallo I Piracés, Jean-Philippe Pons, and Peter Sturm. Minimizing the multi-view stereo reprojection error for triangular surface meshes. In *BMVC 2008-British Machine Vision Conference*, pages 1–10. BMVA, 2008.

[12] Manfredo P Do Carmo. *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Courier Dover Publications, 2016.

[13] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[15] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[16] Vu Hoang Hiep, Renaud Keriven, Patrick Labatut, and Jean-Philippe Pons. Towards high-resolution large-scale multi-view stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1430–1437. IEEE, 2009.

[17] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018.

[18] Michal Jancosek and Tomás Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *CVPR 2011*, pages 3121–3128. IEEE, 2011.

[19] Andreas Kuhn, Heiko Hirschmüller, Daniel Scharstein, and Helmut Mayer. A tv prior for high-quality scalable multi-view stereo reconstruction. *International Journal of Computer Vision*, 124(1):2–17, 2017.

[20] Andreas Kuhn, Shan Lin, and Oliver Erdler. Plane completion and filtering for multi-view stereo reconstruction. In *German Conference on Pattern Recognition*, pages 18–32. Springer, 2019.

[21] Florent Lafarge, Renaud Keriven, Mathieu Brédif, and Vu Hoang Hiep. Hybrid multi-view reconstruction by jump-diffusion. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 350–357. IEEE, 2010.

[22] Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

[23] Keyang Luo, Tao Guan, Lili Ju, Haipeng Huang, and Yawei Luo. P-mvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10452–10461, 2019.

[24] Lena Maier-Hein, Peter Mountney, Adrien Bartoli, Haytham Elhawary, D Elson, Anja Groch, Andreas Kolb, Marcos Rodrigues, J Sorger, Stefanie Speidel, et al. Optical techniques for 3d surface reconstruction in computer-assisted laparoscopic surgery. *Medical image analysis*, 17(8):974–996, 2013.

[25] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.

[26] Minwoo Park, Jiebo Luo, Andrew C Gallagher, and Majid Rabbani. Learning to produce 3d media from a captured 2d video. *IEEE Transactions on Multimedia*, 15(7):1569–1578, 2013.

[27] Andrea Romanoni and Matteo Matteucci. Tapa-mvs: Textureless-aware patchmatch multi-view stereo. *arXiv preprint arXiv:1903.10929*, 2019.

[28] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[29] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[30] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017.

[31] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.

[32] Shuhan Shen. Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. *IEEE transactions on image processing*, 22(5):1901–1914, 2013.

[33] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[34] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. Ieee, 2008.

[35] Engin Tola, Christoph Strecha, and Pascal Fua. Efficientlarge-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012.

[36] Kaixuan Wang and Shaojie Shen. Mvdepthnet: real-time multiview depth estimation neural network. In *2018 International Conference on 3D Vision (3DV)*, pages 248–257. IEEE, 2018.

[37] Jian Wei, Benjamin Resch, and Hendrik PA Lensch. Multi-view depth map estimation with cross-view consistency. In *BMVC*, 2014.

[38] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013.

[39] Pengfei Wu, Yiguang Liu, Mao Ye, Zhenyu Xu, and Yunan Zheng. Geometry guided multi-scale depth map fusion via graph optimization. *IEEE Transactions on Image Processing*, 26(3):1315–1329, 2017.

[40] Xian Xiao, Changsheng Xu, Jinqiao Wang, and Min Xu. Enhanced 3-d modeling for landmark image classification. *IEEE Transactions on Multimedia*, 14(4):1246–1258, 2012.

[41] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5483–5492, 2019.

[42] Youze Xue, Jiansheng Chen, Weitao Wan, Yiqing Huang, Cheng Yu, Tianpeng Li, and Jiayu Bao. Mvscrf: Learning multi-view stereo with conditional random fields. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4312–4321, 2019.

[43] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[44] Andrei Zaharescu, Edmond Boyer, and Radu Horaud. Transformesh: a topology-adaptive mesh-based approach to surface evolution. In *Asian Conference on Computer Vision*, pages 166–175. Springer, 2007.

[45] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1517, 2014.