Accurate Depth Map Estimation from Small Motions

Hossein Javidnia
National University of Ireland, Galway
University Road, Galway, Ireland, H91 TK33
{h.javidnia1}@nuigalway.ie

Peter Corcoran
National University of Ireland, Galway
University Road, Galway, Ireland, H91 TK33
{peter.corcoran}@nuigalway.ie

Abstract

With the growing use of digital lightweight cameras, generating 3D information has become an important challenge in computer vision. Despite several attempts presented in the literature to solve this challenge, it remains an open problem when it comes to the structural accuracy of the depth map and the required baseline (distance between the first and the last frames) to capture a sequence of images. In this paper, a novel approach is proposed to compute a high quality dense depth map together with a semi-dense/dense 3D structure from a sequence of images captured on a narrow baseline. Computing the depth information from small motions has been a challenge for decades because of the uncertain calculation of depth values when using a small baseline – up to 12mm. The proposed method can, in fact, perform on a much wider range of baselines from 8 mm up to 400 mm while respecting the structure of the reference frame. The evaluation has been done on more than 10 sets of recorded small motion clips and for the wider baseline, on 7 sets of stereo images from Middlebury benchmark. Preliminary results indicate that the proposed method has a better performance in terms of structural accuracy in comparison with the current state of the art methods. Also, the performance of the proposed method remains stable even when only a low number of frames are available for processing.

1. Introduction

The use of consumer cameras, specifically smartphones is growing continuously nowadays and the level of expectation around what these cameras can do is increasing year by year. Consumers and photographers generally prefer to have advance features such as shallow depth of field in their images. This effect requires a large aperture like the ones used in DSLR cameras. Lightweight cameras like those in smartphones are equipped with small apertures which are not capable of reproducing this effect.

Because these types of cameras are equipped with only one lens, this feature is commonly implemented by using a focal stack to compute the depth map [1, 2, 3, 4]. An alternative approach is to compute the 3D structure of the scene and the corresponding depth map.

The 3D structure can be computed using the frame-to-frame movements of the handheld camera. Movements of the camera can occur for several reasons, such as natural hand-shake, or when the user moves the camera slightly to capture a better scene. Generally, this effect is considered as an issue to be solved with image stabilization methods or stabilization gear such as tripods. However these types of movements can be used advantageously in a variety of applications for instance synthetic defocus [5, 6].

The baseline between sequences of frames captured as a sudden motion is considered to be small if it's less than ~8 mm. This restricts the viewing angle of a 3D point to less than 0.2° [7]. Due to this limitation the general Structure from Motion (SfM) methods fails [8, 9, 10] and the computed depth map will be highly penalized.

Several works addressed the challenges of the Structure from Small Motion (SfSM) [5, 7, 11, 12] and proposed a number of algorithms. But there are still a couple of open challenges remaining for these methods such as:

1- These methods fail for baselines wider than ~12 mm. In wide-baseline motions, local image deformations cannot be realistically approximated by translation or translation with rotation and a full affine model is required. Also larger baselines increase the observed disparities, but increase the difficulty of finding corresponding points due to a larger change in viewpoint. This statement is specifically targeting the close scenes with shorter depth ranges.
2- These methods return false results when the number of the input frames is less than 15 frames.
3- The structure of the depth map is not respected properly based on the reference frames. More specifically the depth maps generated by these methods suffer from the lack of accuracy along the edges and corners of structures within the imaged scene.
4- Some of these methods suffer from missing/undefined patches in the depth map, especially along the boundaries of the image.

In this paper, we propose an approach to estimate the depth from small motion clips that addresses each of the challenges mentioned above. In addition to its ability to provide high structural accuracy and occlusion handling, the proposed method has 2 important additional advantages:

1- It is able to process a sequence of image frames with baselines as large as 400 mm.
2- There is no restriction on the minimum number of frames in the proposed method. The evaluation shows that it can perform accurately for $frames \geq 2$.

In the next section, we review the previous works done in this area. Section 3 presents the details of the proposed approach and the evaluation and comparison results are presented in section 4.

2. Related Works

The first step in the process of the SfSM is to build a dense 3D model from the sequence of images. This step is widely studied in several SfM research works [13, 14, 15].

In SfM, bundle adjustment [16] is used to find the optimal estimation of the sparse 3D structure of the scene and positions of camera poses. Nonlinear least square is used as the basic cost function to evaluate the reprojection error from undistorted to distorted image domain. There are several issues that must be solved for this method to be successful:

1. The accuracy of the estimated 3D structure is highly dependent on proper initialization of the cost function. To solve this problem factorization methods have been widely adopted in SfM literature as a means for initializing the bundle adjustment [17, 18, 19].
2. When encountering continues texture-less surface, the method is not capable of producing 3D points due to the lack of features and the failure of the feature tracking.
3. The feature tracking is also an issue in case of rapid movements.
4. Complex computation of the reprojection error for inverse depth representation because of mapping the projected 3D points from the undistorted image domain to the distorted image domain [20]. This issue makes the normal bundle adjustment improper for small motions.

To overcome the problems of the common bundle adjustments with small baseline motions, a modified bundle adjustment is presented in [11]. In this case the reprojection error is calculated from distorted to undistorted image domain [21]. This solves the inverse depth representation problem. The method presented in [11] also employs the idea that in small motion clips the cost function can be initialized better as long as the camera poses or the distance between frames are closer to each other. The idea used in [11] was initially introduced in [7] to find the trajectory of the camera from small motion. The density profile in [7] is created by random depth initialization and plane sweeping based image matching [22, 23]. It employs Markov Random Field [24] to regularize the estimated depth effectively.

The method presented in this paper appears to be the first to deal successfully with wider baselines and low frame-rate motion clips. This work presents evaluation and comparisons with other methods in both small and large baseline motions. The results demonstrate that the method proposed here performs better in terms of accuracy of the depth estimation and respecting the structure of the reference image frame.

Fig. 1 illustrates the general overview of the proposed SfSM approach.

3. Proposed Method

The main steps of the proposed SfSM approach are detailed and explained in this section.

The feature detection of the 3D reconstruction block in the proposed method is equipped with ORB features [25]. The correspondence features location to the initial features is found by Kanade-Lukas-Tomashi (KLT) method [26].

The bundle adjustment presented in [11] is used for 3D structure optimization based on the Huber loss function [27]. The reason for employing this bundle adjustment is the different way of measuring the reprojection error than the usual SfM methods.

The reprojection error is computed by mapping the points in the distorted domain to the points in the undistorted domain. The point of this change is to make the reprojection error computation less complex for inverse depth estimation. Using this technique enables the proposed method to perform on uncalibrated motion clips.

Fig. 2 shows the 3D reconstruction by our method and Hyowon Ha *et al.* [11].

3.1. Dense Matching Profile

The basic idea of the dense matching in the current paper is based on the Plane Sweeping method [22]. Different from plane sweeping based stereo matching methods, we estimate the $\boldsymbol{k}$-th plane directly from the set of ORB matches. If $(\boldsymbol{u_i}, \boldsymbol{v_i})$ and $(\boldsymbol{u_i} - \boldsymbol{k}, \boldsymbol{v_i})$ represents the pixel i in the left image and the correspondence match in the right image respectively, then the set of $\mathbb{M} = \{\boldsymbol{u_i}, \boldsymbol{v_i}, \boldsymbol{k}\}$ denotes the match of the two pixels.

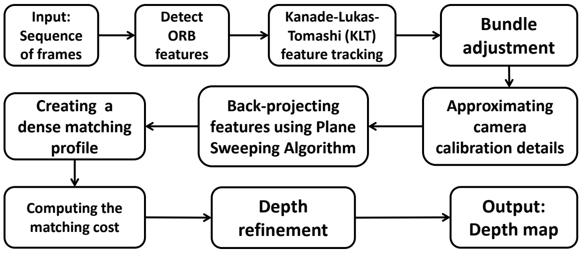

Input: Sequence of frames
Detect ORB features
Kanade-Lukas-Tomashi (KLT) feature tracking
Bundle adjustment
Approximating camera calibration details
Back-projecting features using Plane Sweeping Algorithm
Creating a dense matching profile
Computing the matching cost
Depth refinement
Output: Depth map

Figure 1. General overview of the proposed SfSM

At the pixel $(\boldsymbol{u_i}, \boldsymbol{v_i})$, the disparity is $\mathfrak{D}(\boldsymbol{u}, \boldsymbol{v}) = \mathfrak{p} \dotplus (\boldsymbol{u_i}, \boldsymbol{v_i}, \mathbf{1})$, where $\mathfrak{p} = (\mathfrak{p}_1, \mathfrak{p}_2, \mathfrak{p}_3)$ represents the plane.

To compute the sequence of disparity planes, a segmentation tree [28] is used. The overall objective function in the proposed method which is being minimized by the segmentation tree is:

$$\sum_i^n \mathcal{E}(\boldsymbol{m}_i, \mathfrak{p}_{j_i}) \qquad (1)$$

where $\boldsymbol{m}_i \in \mathbb{M}$ and $\boldsymbol{m}_i$ is part of the plane $\boldsymbol{j}_i$. The goal of this function is to measures the error between the true disparity at $\boldsymbol{m}$ and the disparity generated by the plane. $\boldsymbol{k}$-th disparity plane is computed by minimizing this function using a graph $\boldsymbol{G}$. The graph $\boldsymbol{G}$ is constructed by connecting each node $\boldsymbol{m}$ to its ten nearest neighbours computed by Euclidean distance.

3.2. Matching Cost and Plane Sweeps

At the first step, the frame $\boldsymbol{k}$ is resampled into an $[\boldsymbol{x}, \boldsymbol{y}]$ area from frame $\boldsymbol{k} + \mathbf{1}$ using B-Spline interpolation. The correlation score of $\boldsymbol{\mathcal{N}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{k})$ is obtained over 5×5 patches. The score is turned into the pixel-wise matching cost as:

$$\boldsymbol{\mathcal{C}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{k}) = \mathbf{1} - \boldsymbol{\mathcal{N}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{k}) \qquad (2)$$

where $\boldsymbol{\mathcal{N}}$ refers to Normalized Cross Correlation and $\boldsymbol{\mathcal{C}}$ refers to the matching cost.

The raw cost is converted from pixel cost to the aggregated volume cost using adaptive cost aggregation [29].

As it is common in most of the stereo algorithms, the cost volume is computed as:

$$\boldsymbol{\mathcal{C}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{k}) = \sum_{(x,y)} \boldsymbol{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{k}) \qquad (3)$$

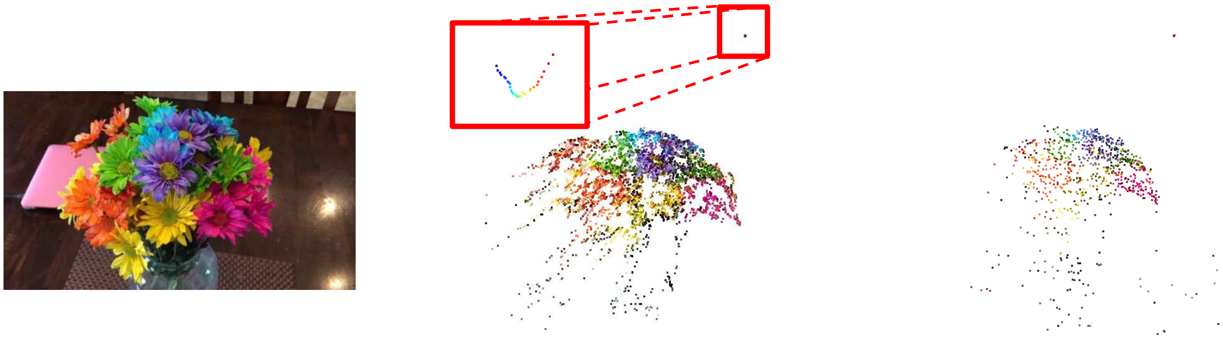
But this assumption has a requirement that the surface has to be facing the camera and this makes the pixels surrounding a patch to have almost the same disparity value. The restriction for this assumption arises from the common and important challenge of handling the occlusions along the boundaries in stereo matching methods. To resolve this issue, the cost volume is computed by aggregating the cost based on the color and similarity features. The matching cost from the resampled image is weighted by a similarity feature, in this paper the $\boldsymbol{\Delta\mathcal{C}}$ and the color difference between $\boldsymbol{p} = (\boldsymbol{u}, \boldsymbol{v})$ and $\boldsymbol{r} = [\boldsymbol{x}, \boldsymbol{y}]$ as $\boldsymbol{\Delta}\mathbb{C}$.

The weighting function $\boldsymbol{w}$ can be defined as:

$$\boldsymbol{w}(\boldsymbol{p}, \boldsymbol{r}) = \mathbf{exp}\left(\frac{-\boldsymbol{\Delta}\mathbb{C} - \boldsymbol{\Delta\mathcal{C}}}{\boldsymbol{t}}\right) \qquad (4)$$

where $\boldsymbol{t}$ is the weighting constant. The basic idea in Eq. 4 is to aggregate the matching cost based on color and feature similarity (geometric proximity). Considering a pixel $\boldsymbol{p}$ and pixel $\boldsymbol{r}$, the matching cost from $\boldsymbol{r}$ is weighted by the color difference between $\boldsymbol{p}$ and $\boldsymbol{r}$, and the Euclidian distance between $\boldsymbol{p}$ and $\boldsymbol{r}$ on the image plane.
The computed aggregated cost from the pixel-wise cost is:

$$\mathfrak{C}(\boldsymbol{p}, \boldsymbol{k}) = \frac{\sum_{r,r'} \boldsymbol{w}(\boldsymbol{p},\boldsymbol{r})\boldsymbol{w}'(\boldsymbol{q},\boldsymbol{r}')\boldsymbol{\mathcal{C}}(\boldsymbol{p},\boldsymbol{k})}{\sum_{r,r'} \boldsymbol{w}(\boldsymbol{p},\boldsymbol{r})\boldsymbol{w}'(\boldsymbol{q},\boldsymbol{r}')} \qquad (5)$$



a. A frame of the sequence

b. **Our** reconstructed 3D point cloud and the estimated camera trajectory – Side view

c. Reconstructed 3D point cloud by **Hyowon Ha *et al.***[11] – Side view

Figure 2. Comparison of our 3D reconstruction with Hyowon Ha *et al.* [11]

where $\boldsymbol{p}$ and $\boldsymbol{q}$ are the matching pixels and $\boldsymbol{r'}$ is the support region of $\boldsymbol{q}$. Eq. 5 represents the weighted sum of per pixel which is used as cost aggregation.

Once the cost volume is computed, the initial disparity map $\boldsymbol{\mathcal{D}}$ is obtained by parametrizing the plane equation in pixel level with local disparity values. The condition for choosing the local disparity is minimizing the total aggregation cost.

Although $\boldsymbol{\mathcal{D}}$ has a quite reasonable depth values but it still can be noisy and the structure of the depth map can suffer from inaccurate edges and corners. To solve this issue and handle the probable occlusions, we define 2 terms as the smoothness term and the data term.

The smoothness term for pixels $\boldsymbol{p}$ and $\boldsymbol{q}$ and the displacement vector $\boldsymbol{v}$ is defined as:

$$\boldsymbol{\mathcal{S}}(\boldsymbol{v}_p, \boldsymbol{v}_q) = \boldsymbol{w}_{pq} \min\left(\left\|\boldsymbol{v}_p - \boldsymbol{v}_q\right\|_2^2, \boldsymbol{t}\right) \quad (6)$$

where $\boldsymbol{w}_{pq}$ is the weighting variable computed by the color similarity of the patch surrounding $\boldsymbol{p}$ and pixel $\boldsymbol{q}$. $\boldsymbol{t}$ is the reduction threshold.

This term has the most influence on occlusion handling by propagating the cost from the non-occluded pixels to occluded pixels based on their similarity.

Following the smoothness term, the data term is defined as:

$$\boldsymbol{d}_p(\boldsymbol{v}_p) = \begin{cases} \left\|\boldsymbol{v}_p - \boldsymbol{v}_p^*\right\|_2^2 & \boldsymbol{p \text{ is non} - occluded} \\ \boldsymbol{0}, & \boldsymbol{otherwise} \end{cases} \quad (7)$$

where $\boldsymbol{v}_p^*$ is the initial displacement vector.

Defining these 2 terms handle more than 96% of the occlusions but still, there are some missing parts, specifically around the boundaries of the objects which cause an inaccurate edge structure in the depth map. Considering a pixel $\boldsymbol{p}$ located in the occluded area. We try to estimate its disparity value by using a small patch $\boldsymbol{\mathcal{H}}(\boldsymbol{p})$ with known disparity values, centered at $\boldsymbol{p}$.The disparity value of $\boldsymbol{p}$ can be estimated by the following equation:

$$\boldsymbol{D}_p = \boldsymbol{D}_q + \langle \boldsymbol{\mathcal{g}} \boldsymbol{D}_q, \boldsymbol{p} - \boldsymbol{q} \rangle \quad (8)$$

where $\in \boldsymbol{\mathcal{H}}(\boldsymbol{p})$, $\boldsymbol{D}_q$ and $\boldsymbol{\mathcal{g}}\boldsymbol{D}_q$ are the disparity value and gradient respectively. $\langle , \rangle$ represents the inner product operation.

This estimation is done for all the pixels in $\boldsymbol{\mathcal{H}}(\boldsymbol{p})$ and at the end the final disparity map of $\boldsymbol{p}$ is obtained by:

$$\boldsymbol{D}_p = \frac{\sum_{q \in \mathcal{H}(p)} \omega_{pq} [\boldsymbol{D}_q] + \langle \boldsymbol{\mathcal{g}} \boldsymbol{D}_q, \boldsymbol{p} - \boldsymbol{q} \rangle}{\sum_{q \in \mathcal{H}(p)} \omega_{pq}} \quad (9)$$

where $\boldsymbol{\omega}_{pq} = \boldsymbol{w}_{pq}$ is the weighting function and it is defined as:

$$\boldsymbol{\omega}_{pq} = \boldsymbol{\omega}_{ds(pq)} \boldsymbol{\omega}_{cl(pq)} \quad (10)$$

where $\boldsymbol{\omega}_{ds(pq)}$ denotes the distance term and $\boldsymbol{\omega}_{cl(pq)}$ color similarity term.

$$\boldsymbol{\omega}_{ds(pq)} = \mathbf{exp}\left(-\frac{\|\boldsymbol{p} - \boldsymbol{q}\|^2}{2\boldsymbol{\alpha}^2}\right) \quad (11)$$

$$\boldsymbol{\omega}_{cl(pq)} = \mathbf{exp}\left(-\frac{\left\|\boldsymbol{r}_p - \boldsymbol{r}_q\right\|^2}{2\boldsymbol{\beta}^2}\right) \quad (12)$$

where $\boldsymbol{r}_p$ and $\boldsymbol{r}_q$ are the color values of the pixels $\boldsymbol{p}$ and $\boldsymbol{q}$ respectively. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are constant values specified experimentally.

When corresponding matching pixels have dissimilar colors because of illumination variations, the inaccurate disparity map is generated. Adding the color similarity term to the weighting function helps to handle this issue.

To treat the probable artifacts caused by plane sweeping algorithm due to the over/under sampling, the inter frame motion estimation problem is reformulated to be optimized over image intensity function for sequence of frames. The formulation computes the cost over all pixels of the reference frame. Through this formulation a geometrical fidelity is checked for patch of pixels. The fidelity check is based on consistency of the normal directions between neighboring pixels to make sure they have similar surface normal vector. The correlation between the normal vectors of the center pixel and neighboring pixels can lead optimization to refine the depth map.

3.3. Final Depth Refinement

After computing the final depth map from the previous step, it is refined by the guided joint filter presented in [30]. The filter in [30] is based on the mutual information. The mutual information guides the weighted median filter to follow the structure of the RGB image while filtering the correspondence depth map. To keep the valid depth values and just filtering the false ones, window selection step of the median filter is designed to be adaptive using the joint histogram. The probable remaining artifacts after the adaptive weighted median filter are being eliminated by normalized interpolated convolution in diffused image domain.

Beside the performance of this filter in occlusion handling, it helps the depth map to follow the image structure more precisely. Without defining any limitations, for small parallax including slow-enough motion, or far-enough objects, or fast-enough temporal sampling, occluded areas are small. Our experiments show that the mentioned filter guarantees intra-object occlusion handling accurately even in wide baseline motions. The failure of the filter might occur in the case of large inter-object occlusion. Generally in small motions the main occlusion to deal with is intra-object. Although it is worth pointing out that the filter is able to handle relatively good amount of inter-object occlusions unless there is a considerable displacement or off axis parallax.

4. Experiments and Evaluation

In this paper, the experiment and evaluation is done in 2 parts. First, the proposed method is evaluated for small motions and in the second part, it is evaluated for stereo image sets. The first comparison is done against Hyowon Ha *et al.* [11] and Kevin Karsch *et al.* [31] and the second comparison is done against 3DMST [32] and APAP-Stereo [33] stereo matching algorithms ranked in Middlebury stereo benchmark [34], training dense section.

For the first part, the dataset from [11] is used and we also provided 10 other small motion clips using the devices shown in Table 1. The motion clips are available to download at (goo.gl/m5QohE).

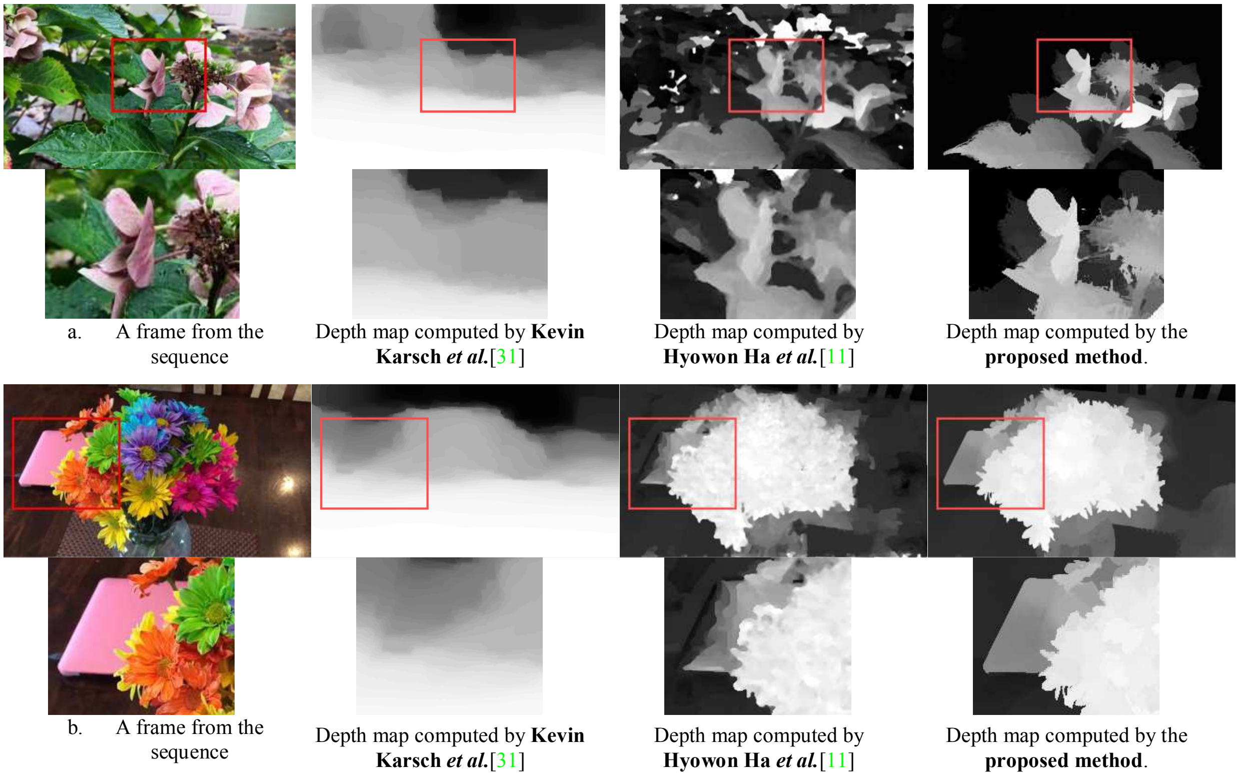
There is no ground truth in this form of evaluation, but the performance of the proposed method makes it possible to show the visual comparison with 2 other methods.

Fig. 3 shows the depth map computed by Hyowon Ha *et al.* [11], Kevin Karsch *et al.* [31] and our method. These images show the performance of the proposed method in terms of accuracy of the depth along edges and the depth values on the surface of objects in the case of small motions and small baseline.

The results by Hyowon Ha *et al.* [11] and Kevin Karsch *et al.* [31] have inaccurate depth values along the edges and corners of the objects as seen in Fig.3.a and Fig.3.b. Note that due to the very small baseline between the frames these methods distinguish foreground information better than background information.

In some cases as shown in Fig.3.b, the depth map estimated by these methods are suffering from inaccurate depth values on an object's surface or the depth values of the background and foreground objects are mixed together which cause inaccurate performance in segmentation and 3D reconstruction applications.

Fig. 4 shows how the inaccurate depth values along the edges can generate a faulty 3D structure. The highlighted patches show a part of the 3D textured mesh generated based on the reference frame and the corresponding depth map.



a. A frame from the sequence

Depth map computed by **Kevin Karsch *et al.***[31]

Depth map computed by **Hyowon Ha *et al.***[11]

Depth map computed by the **proposed method**.

b. A frame from the sequence

Depth map computed by **Kevin Karsch *et al.***[31]

Depth map computed by **Hyowon Ha *et al.***[11]

Depth map computed by the **proposed method**.

Figure 3. Comparison of the depth from small motion with state-of-the-art methods

Table 1. Devices used for making our own dataset

	Device	Resolution	fps
1	iPhone6 Plus	1080p	60
2	iPhone6 Plus	1080p	30
3	iPhone7	1080p	30
4	iPhone7	720p	30
5	iPhone7 Plus	1080p	30
6	iPhone7 Plus	4K	30

The 3D mesh generated based on the depth map by Hyowon Ha *et al.* [11] is suffering from missing parts on objects' surfaces which is caused by inaccurate depth values on reference patches.

For the second part of the comparison, we evaluated the performance of the proposed method for a set of stereo images. In this case, we considered the left and right images as a sequence of frames, 2 frames instead of processing 30 frames by considering the fact that the method is designed to perform on small baseline motions while the higher number of frames provides the higher number of inliers at the feature matching step. Note that more experiments are done on ordinal camera motions recorded by authors [11]. The depth map in Fig.1.d and Fig.1.c in the *Appendix_1* (goo.gl/fqqUxk) is generated using only 2 frames of the real camera motion which is captured by users. That's why the result of the method [11] in Fig.1.d and Fig.1.c in *Appendix_1* is different from what is published in the main paper [11]. The depth map in [11] is computed using 30 frames, but in this paper only 2 frames are used. That shows the superior performance of the proposed method.

To have an accurate evaluation at this part, we used 7 pairs of stereo images from Middlebury stereo benchmark with the corresponding ground truth depth maps. Fig. 5 represents the visual comparison of this evaluation. Fig.5.a and Fig.5.b show how the proposed method is capable of keeping the structure of the reference image in the depth map, especially important features like edges and corners in comparison with top stereo matching algorithms.

The accuracy of the estimated depth by each method has been evaluated against the ground truth which is provided by the benchmark and the numerical results are presented in Table 2. These results illustrate the competitive performance of the proposed method in terms of accuracy of the depth along edges and the depth values on the surface of the objects against top algorithms in Middlebury benchmark. Although there is still the potential for this method to be improved as it is not performing perfectly in some cases.

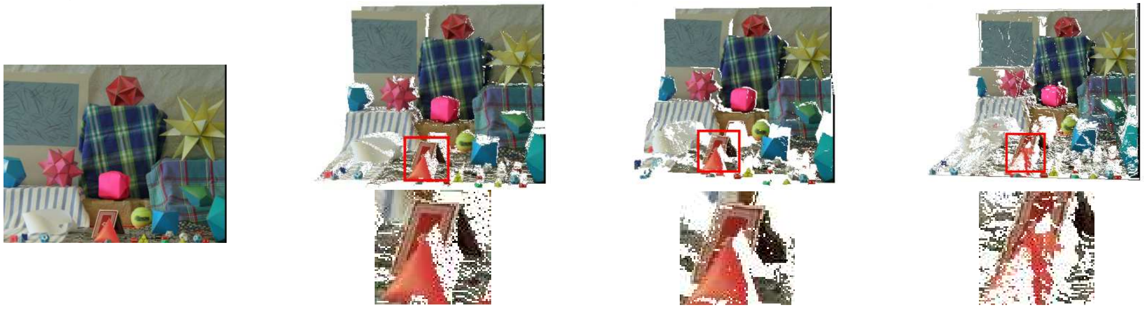
To find more visual/extended numerical results and the higher resolution version of the images presented in Fig. 3 and Fig. 5 please refer to *Appendix_1.*

For evaluation purposes, 4 metrics including PSNR, RMSE, Universal Quality Index (UQI) [35] and Structural Similarity Index (SSIM) [36] are used. Table 2 presents the average numerical comparison of the methods per metric on the chosen stereo sets from the benchmark. The extended numerical results are presented in *Appendix_1.*

Fig. 6 represents the SSIM and UQI maps of the depth map generated by each method from the images in Fig. 5. The SSIM map show how similar is the structure of the computed depth map to the ground truth. The lighter and darker pixel values show more and less structural similarity to the ground truth respectively.

The general quality of the generated depth maps in comparison with the ground truth is shown as UQI map. The lighter and darker pixel values show more and less similarity to the ground truth respectively.

As it is illustrated in Fig. 6, the proposed method is estimating depth maps relatively close to the ground truth in both structural and quality indices as there are larger areas covered with lighter values. The areas presented in dark show how far the depth values are from ground truth based on SSIM and UQI maps.



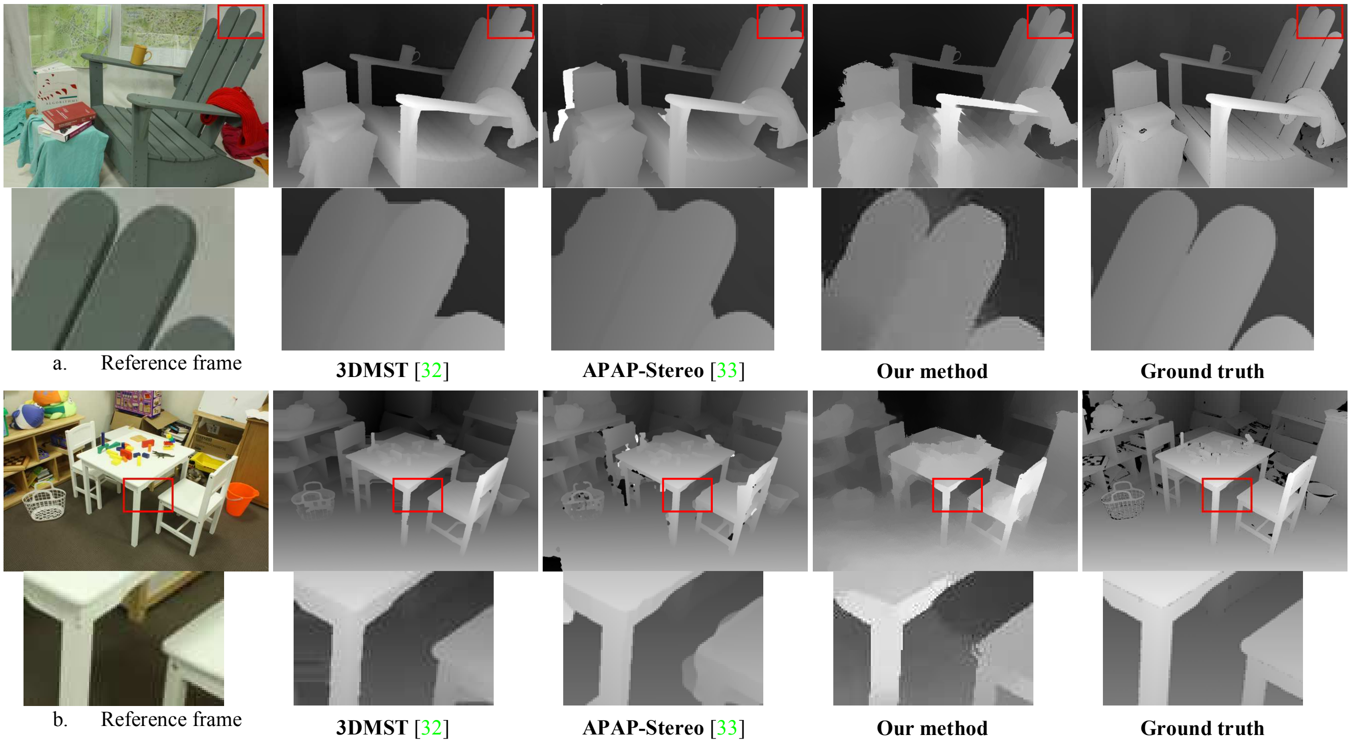
Reference frame

3D textured mesh – **Ground truth**

3D textured mesh – **Our method**

3D textured mesh – **Hyowon Ha *et al.*[11]**

Figure 4. Comparison of the 3D textured mesh based on the depth maps generated by the proposed method and Hyowon Ha *et al.* [11]



a. Reference frame

3DMST [32]

APAP-Stereo [33]

Our method

Ground truth

b. Reference frame

3DMST [32]

APAP-Stereo [33]

Our method

Ground truth

Figure 5. Comparison with 3DMST [32] and APAP-Stereo [33] based on Middlebury benchmark

There are still considerable parts in the depth maps generated by the proposed technique which look far from ground truth but the results are reasonably close to the top stereo matching algorithms.

SSIM maps in Fig. 6 show that the structure of the reference frames including the sharp edges and corners, is respected in the estimated depth map and this is one of the advantages of the proposed method.

Occluded regions are important features in depth extraction methods [37][38]. Unlike most of the current algorithms that are not able to handle this issue, the proposed method can estimate the information on invisible scene components. Fig. 6 illustrates another important advantage of the proposed technique which is the acceptable performance on lower fps motions such as 2 frame stereo images. The presented cost function makes the algorithm capable of processing motions with wider baseline.

The robustness of the proposed method is also evaluated by considering the magnitude of the baseline and number of the frames. The result illustrates that the algorithm can generate depth with the similarity of ~75% to the ground truth as long as the magnitude of the baseline is greater than ~6% of the nearest scene depth and the number of frames captured exceeds 2 frames.

Table 2. Numerical comparison of the methods/average per metric for seven stereo set

	PSNR	RMSE	UQI	SSIM
Ours	17.281	35.491	0.87	0.70
3DMST [32]	18.315	29.975	0.89	0.82
APAP-Stereo [33]	18.734	28.672	0.95	0.85

5. Conclusion

This paper has presented an accurate approach for computing the depth map from narrow baseline motion clips.

Six important contributions have been proposed in this work as follows:

General Contributions:

1. Generally in small motions, the feature tracker can obtain more inliers due to the small difference between the frames. However the number of inliers reduces when the baseline becomes wider and as the result the generated depth map becomes inaccurate. The modified cost function in the proposed method makes it capable of processing sequence of frames with the baseline up to 400 mm while most of the methods in this field fail for the baselines wider than ~12 mm.
2. Accurate performance for $frame \geq 2$

3. Occlusion handling by respecting the structure of the reference frame.

Technical Contributions:

1. New data and smoothness terms are defined to recondition cost volume and cost aggregation function.
2. Proposed cost propagation is formulated as energy minimizer function for depth on each pixel point.
3. The proposed method can approximate non-planar surfaces, while being robust against depth outliers and occlusion.

This practical application has the potential to be used in smartphone cameras. These cameras are designed to gather image frames before and after a user initiate a capture sequence. The 3D information obtained by this method can be used for synthetic defocus applications, object detection and segmentation purposes and scene analyses and understanding.

Unlike other techniques, the 3D points generated by the proposed method at the background of a scene don't have high uncertainty. This gives a uniform and continuous shape to the point cloud from the closest to the furthest point visible to the camera.

A range of different experiments on both wide and narrow baselines have been conducted which proved that the proposed method exhibits improved performance over state of the art methods. In addition this method is sufficiently robust to perform adequately at low frame rates and with a small number of input images.

With respect to the performance and accuracy of the studied method, there is still the computational time of this technique which has to be considered as a trade-off. The method has been tested on a device equipped with Intel i7-5600U @ 2.60GHz CPU and 16 GB RAM. The whole process of computing the 3D structure and depth map take about 6-8 minutes. The most expensive part of the method is the bundle adjustment optimization which is takes around 4-5 minutes on high resolution images and motivates our future research activities to make this method suitable for real-time applications. The full evaluation of this method requires a dataset of video sequences with valid ground truths which at the moment is not publicly available. As part of our future work on this topic we would like to provide a dataset of video sequences with the ground truths for close-range scenes using ToF cameras.

Acknowledgement

The research work presented here was funded under the Strategic Partnership Program of Science Foundation Ireland (SFI) and co-funded by SFI and FotoNation Ltd. Project ID: 13/SPP/I2868 on "*Next Generation Imaging for Smartphone and Embedded Platforms*".

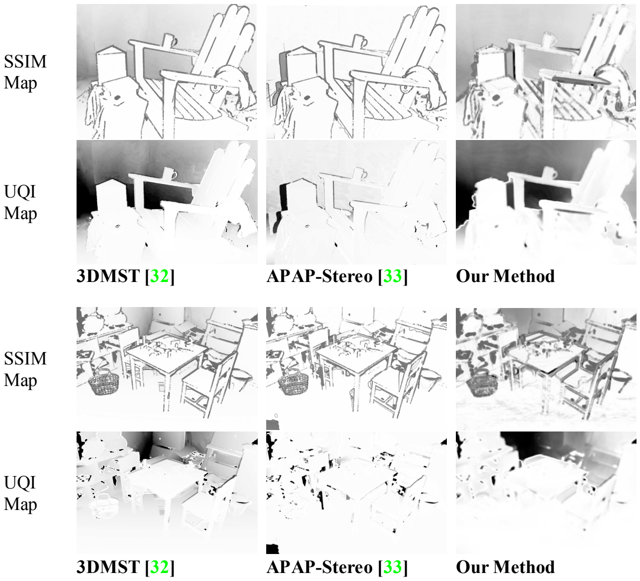


Figure 6. Comparison of SSIM and UQI maps

References

[1] David E. Jacobs, Jongmin Baek, Marc Levoy, "Focal Stack Compositing for Depth of Field Control", *Stanford Computer Graphics Laboratory Technical Report* 2012-1. October, 2012.

[2] Lin, Haiting, Can Chen, Sing Bing Kang, and Jingyi Yu, "Depth recovery from light field using focal stack symmetry", *In Proceedings of the IEEE International Conference on Computer Vision*, pp. 3451-3459, 2015.

[3] Suwajanakorn, Supasorn, Carlos Hernandez, and Steven M. Seitz, "Depth from focus with your mobile phone", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3497-3506, 2015.

[4] Bailey, Stephen W., Jose I. Echevarria, Bobby Bodenheimer, and Diego Gutierrez, "Fast depth from defocus from focal stacks", *The Visual Computer,* 31, no. 12, pp. 1697-1708, 2015.

[5] S. Im, H. Ha, G. Choe, H. G. Jeon, K. Joo, and I. S. Kweon, "High Quality Structure from Small Motion for Rolling Shutter Cameras," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 837-845, 2015.

[6] Barron, Jonathan T., Andrew Adams, YiChang Shih, and Carlos Hernández. "Fast bilateral-space stereo for synthetic defocus." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4466-4474. 2015.

[7] F. Yu and D. Gallup, "3D Reconstruction from Accidental Motion," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3986-3993, 2014.

[8] Tron, Roberto. "A Factorization Approach to Inertial Affine Structure from Motion." *arXiv preprint arXiv*:1608.02680, 2016.

[9] Agudo, Antonio, Francesc Moreno-Noguer, Begoña Calvo, and José María Martínez Montiel. "Sequential non-rigid structure from motion using physical priors." *IEEE transactions on pattern analysis and machine intelligence*, 38, no. 5 , pp. 979-994, 2016.

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

[11] H. Ha, S. Im, J. Park, H. G. Jeon, and I. S. Kweon, "High-Quality Depth from Uncalibrated Small Motion Clip," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5413-5421, 2016.

[12] N. Josh and L. Zitnick, "Micro-Baseline Stereo," *Microsoft Research Technical Report,* vol. MSR-TR-2014-73, May 2014.

[13] G. Zhang, H. Liu, Z. Dong, J. Jia, T. T. Wong, and H. Bao, "Efficient Non-Consecutive Feature Tracking for Robust Structure-From-Motion," *IEEE Transactions on Image Processing,* vol. 25, pp. 5957-5970, 2016.

[14] H. Guan and W. A. P. Smith, "Structure-From-Motion in Spherical Video Using the von Mises-Fisher Distribution," *IEEE Transactions on Image Processing,* vol. 26, pp. 711-723, 2017.

[15] H. Zhou, K. Ni, Q. Zhou, and T. Zhang, "An SfM Algorithm With Good Convergence That Addresses Outliers for Realizing Mono-SLAM," *IEEE Transactions on Industrial Informatics,* vol. 12, pp. 515-523, 2016.

[16] Triggs, Bill, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. "Bundle adjustment—a modern synthesis." *In International workshop on vision algorithms*, Springer Berlin Heidelberg, 1999, pp. 298-372. 1999.

[17] Y. Dai, H. Li, and M. He, "Projective Multiview Structure and Motion from Element-Wise Factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, pp. 2238-2251, 2013.

[18] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *Computer Vision — ECCV '96: 4th European Conference on Computer Vision Cambridge, UK, April 15–18, 1996 Proceedings Volume II*, B. Buxton and R. Cipolla, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 709-720, 1996.

[19] B. Triggs, "Factorization methods for projective structure and motion," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 845-851, 1996

[20] Ma, Lili, YangQuan Chen, and Kevin L. Moore. "Rational radial distortion models of camera lenses with analytical solution for distortion correction." *International Journal of Information Acquisition* 1, no. 02, pp. 135-147, 2004.

[21] Tamaki, Toru, Tsuyoshi Yamamura, and Noboru Ohnishi. "Unified approach to image distortion." In *Pattern Recognition, Proceedings. 16th International Conference on*, vol. 2, pp. 584-587. IEEE, 2002.

[22] R. T. Collins, "A space-sweep approach to true multi-image matching," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 358-363, 1996.

[23] S. N. Sinha, D. Scharstein, and R. Szeliski, "Efficient High-Resolution Stereo Matching Using Local Plane Sweeps," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1582-1589, 2014.

[24] N. Komodakis and N. Paragios, "Beyond pairwise energies: Efficient optimization for higher-order MRFs," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2985-2992, 2009.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, pp. 2564-2571, 2011.

[26] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," presented at the Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2, Vancouver, BC, Canada, 1981.

[27] P. J. Huber, "Robust Estimation of a Location Parameter," in *Breakthroughs in Statistics: Methodology and Distribution*, S. Kotz and N. L. Johnson, Eds., ed New York, NY: Springer New York, pp. 492-518, 1992.

[28] X. Mei, X. Sun, W. Dong, H. Wang, and X. Zhang, "Segment-Tree Based Cost Aggregation for Stereo Matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 313-320, 2013.

[29] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, "High-Quality Real-Time Stereo Using Adaptive Cost Aggregation and Dynamic Programming," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pp. 798-805, 2006.

[30] H. Javidnia and P. Corcoran, "A Depth Map Post-Processing Approach Based on Adaptive Random Walk With Restart," *IEEE Access,* vol. 4, pp. 5509-5519, 2016.

[31] K. Karsch, C. Liu, and S. B. Kang, "Depth Transfer: Depth Extraction from Video Using Non-Parametric Sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 36, pp. 2144-2158, 2014.

[32] L. Li, X. Yu, S. Zhang, X. Zhao, and L. Zhang, "3D cost aggregation with multiple minimum spanning trees for stereo matching," Applied Optics, vol. 56, pp. 3411-3420, 2017.

[33] vision.middlebury.edu/stereo/eval3/

[34] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal of Computer Vision,* vol. 47, pp. 7-42, 2002.

[35] W. Zhou and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters,* vol. 9, pp. 81-84, 2002.

[36] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing,* vol. 13, pp. 600-612, 2004.

[37] A. Humayun, O. Mac Aodha and G. J. Brostow, "Learning to find occlusion regions," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2161-2168, 2011.

[38] H. Fu, C. Wang, D. Tao and M. J. Black, "Occlusion Boundary Detection via Deep Exploration of Context," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 241-250, 2016.