2019 International Conference on 3D Vision (3DV)

Mobile photometric stereo with keypoint-based SLAM for dense 3D reconstruction

Remy Maxence
Keio University, Japan
maxence.remy@hvrl.ics.keio.ac.jp

Hideaki Uchiyama
Kyushu University, Japan
uchiyama@limu.ait.kyushu-u.ac.jp

Hiroshi Kawasaki
Kyushu University, Japan
kawasaki@ait.kyushu-u.ac.jp

Diego Thomas
Kyushu University, Japan
thomas@ait.kyushu-u.ac.jp

Vincent Nozick
Universite Paris-Est Marne-la-Vallee, France
vincent.nozick@u-pem.fr

Hideo Saito
Keio University, Japan
hs@keio.jp

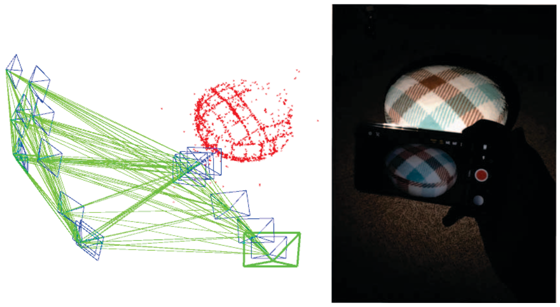
Abstract

The standard photometric stereo is a technique to densely reconstruct objects surfaces using light variation under the assumption of a static camera with a moving light source. In this work, we use photometric stereo to reconstruct dense 3D scenes while moving the camera and the light altogether. In such non-static case, camera poses as well as correspondences between pixels of each frame to apply photometric stereo are required. ORB-SLAM is a technique that can be used to estimate camera poses. To retrieve correspondences, our idea is to start from a sparse 3D mesh obtained with ORB SLAM and then densify the mesh by a plane sweep method using a multi-view photometric consistency. By combining ORB-SLAM and photometric stereo, it is possible to reconstruct dense 3D scenes with a off-the-shelf smartphone and its embedded torchlight. Note that SLAM systems usually struggle with textureless object, which is effectively compensated by the photometric stereo in our method. Experiments are conducted to show that our proposed method gives better results than SLAM alone or COLMAP, especially for partially textureless surfaces.

1. Introduction

Among all the techniques used for 3D reconstruction, photometric stereo shines by its ability to capture details, and to work even with textureless surfaces. This technique has many applications. In the archaeological field, photometric stereo can be considered to generate 3D models of ancient objects difficult to manipulate. In art, it can be used to recover barely visible details of bas-reliefs. Unfortunately, most of existing photometric stereo techniques require a static camera and a moving light, which prevents from a usage on a mobile device; it is a technique that remains only used in laboratories with specific setups. In this work, our objective is to make this technology usable for any smartphone users by moving around a close-up target scene with the flashlight on, which will be used as the moving light source. We are targeting lambertian surfaces in dark environments.

In our case, the camera and the light source are moving together since they are embedded on the same device. This makes the photometric stereo problem more challenging to



(a) Pose estimation

(b) Set-up

Figure 1. Example of one experiment for reconstruction. The experiment is conducted in a dark environment with a smartphone whose flashlight is turned on (b). The smartphone takes a video and we then compute the camera poses (blue rectangles on (a)), the trajectory (green lines on (a)) and a point cloud (red points on (a))

2475-7888/19/$31.00 ©2019 IEEE
DOI 10.1109/3DV.2019.00069

solve compared with the standard fixed camera set-up. Our proposed set-up adds two requirements to the standard photometric problem: (1) an accurate estimation of the camera pose for each input image (which is also the pose of the flashlight), (2) correspondences between pixels to create the photometric system. (1) will be acquired with SLAM. (2) will be avoided using a plane-sweep approach.

The Simultaneous Localisation and Mapping (SLAM) technology is efficient to quickly compute camera poses from a sequence of RGB images in various environments. We reason that the SLAM technology is the solution to overcome the above mentioned issues, which will allow to reconstruct dense 3D models of objects with a mobile device, even in the case of textureless objects.

We propose to use ORB-SLAM [16, 17] to compute camera poses and 3D features (ORB features). ORB-SLAM has the advantage to be usable with many kinds of movements and on any scale. Thanks to photometric stereo, we can compute normal vectors of the initial point cloud found by SLAM. Then, using the multi-view photometric consistency, we can densify this point cloud.

Our contributions are (1) the creation of an algorithm for dense 3D reconstruction with a smartphone using its embedded camera and flashlight, (2) a densification method to reconstruct 3D scenes with only few initial points, and (3) a method of close-up 3D reconstruction that works in dark environment and with partially textureless objects.

2. Related work

Structure-from-motion is the most popular method to get camera poses and sparse 3D features. It is particularly efficient in recovering large-scale structures but struggles with high-frequency details. [6, 18]. This standard technique uses point correspondences to get a trajectory of key points and thus, get the pose of the camera for each image. The state-of-the-art technique that is currently acknowledged is COLMAP [23]. It has the ability to extensively and accurately build a 3D scene from a sequence of RGB images. The drawbacks of this method are that it is highly time-consuming since it will extract from all the images as many features as possible. It also requires clearly visible textures to get a smooth result. Nevertheless, COLMAP is the referential work that has to be considered when estimating the efficiency of a technique for 3D reconstruction.

Photometric stereo is an acknowledged technology [24, 25, 10] that uses a pixel-wise approach to recover the normal vectors and the albedo of a scene captured with a specific setup (a camera and a system of varying lights). Based on the variation of the light intensity, it performs well with static conditions (static scene and camera). The user also usually needs to know the pose of the lights beforehand. Some works targeted the dynamic scenes [28, 4, 15] but they only partially used photometric clues and few considered a dynamic camera. In [11], Higo et al. proposed a mathematical model based on [1] for photometric stereo when using a moving camera. However, they use a heavy process to get their final normal map. They consider an extensive list of possible correspondences for every single pixel and apply the photometric equations for every correspondences.

These last two decades, researchers started to work on unifying Structure from Motion and photometric stereo using various techniques such as Optical flow [26]. In [21], photometric stereo and Structure from Motion are applied separetely; for photometric stereo, the camera remains static but moves for the Structure from Motion. In [8], the authors proposed to use multiple camera. Those solutions are not usable with a monocular camera.

SLAM technologies are used for odometry and can be used with a monocular camera. Recent methods [5, 16, 17] are able to reconstruct large scenes as long as some constraints are respected (apparent textures, no changing light). With only images taken by a moving camera, it is possible to get the camera pose in real-time. Various SLAM algorithms exist; some are dense (depth-map oriented) [5] and others are sparse (key-point oriented) [16, 17]. Depth-map oriented algorithms such as LSD-SLAM [5] usually focus on large scale scenes. Since our focus is on close-up scenes, we chose to use a key-point oriented approach. It also brings many benefits. First, ORB-SLAM [16] is a key-point oriented algorithm that works well on any scale. It is also fast and robust to many different kind of movements including pure rotations [13, 12]. Since we cannot avoid the user to be shaky, this is a key aspect for us. Additionally, the bundle adjustment introduced in [17] brings more accuracy in the pose estimation, which is something essential for photometric stereo. Nonetheless, ORB-SLAM, as a technique using feature-based approach, performs poorly when the textures are not completely obvious, the light changes [19], the scene is blurry or the environment is dark. If those bad conditions appear, the point cloud might be inaccurate and sparse.

In our proposed work, we use as a basis the point cloud found by ORB-SLAM, which gives us a large amount of reliable correspondences. The missing correspondences are then computed by triangulating these reliable points.

3. Proposed method

Starting from a video taken by a smartphone, our objective is to reconstruct a dense 3D point cloud. The pipeline of our proposed method is described in Figure 2. We use ORB-SLAM (section 3.2) to obtain the camera pose and an initial sparse 3D point cloud. With this information, we apply photometric stereo (section 3.3) to obtain the normal vectors. Finally, we densify (section 3.4) our point cloud using a recursive approach of multi-view photometric con-

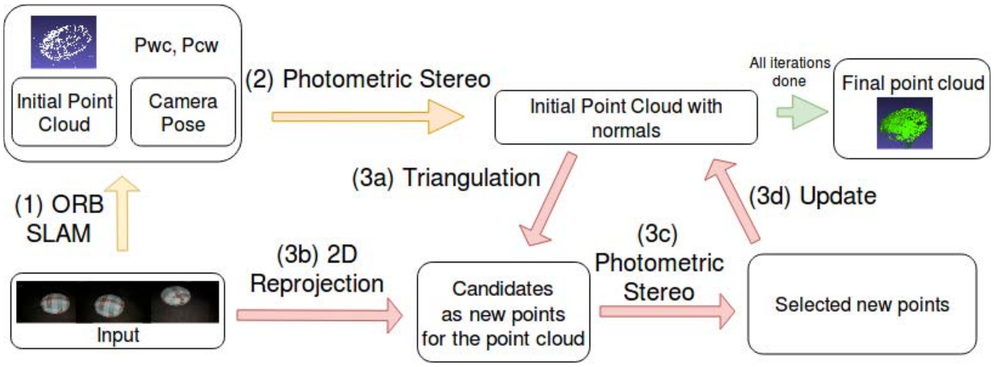


Figure 2. Pipeline of the proposed method. The input is a video taken by a smartphone. The output is a dense point cloud. More details will be provided for each step in the corresponding sections: (1) ORB-SLAM: Section 3.2, (2) Photometric Stereo: Section 3.3, (3) Densification: Section 3.4

sistency.

3.1. Image acquisition

Our proposed method starts with only one input: we provide a video where the camera moves around the object to reconstruct. We do not require any prior knowledge concerning the scene. The camera is calibrated beforehand [27] [3] to obtain camera internal parameters. The input video is divided into several frames. To apply photometric stereo, a large variation of movement is required in order to have a large panel of light orientations. During the acquisition of this video, the torchlight is turned on to create the condition of light variation. To increase the accuracy of our proposed method, we avoid using auto-focus and changing the white balance during the acquisition. We also avoided saturation.

3.2. ORB-SLAM-based algorithm

In our proposed method, we use ORB-SLAM in order to get the camera pose (and consequently the light pose). These poses are then used to build the photometric equations that we will detail in Section 3.3. ORB-SLAM also provides an initial point cloud that will be used as a starting point for densification that will be detailed in Section 3.4.

ORB-SLAM extracts ORB features in all images to find a set of correspondences between all the images as explained in [16]. These correspondences allow us to compute the camera pose of each input image. Since the camera and the torchlight of a smartphone are usually close to one another, we assume that the light pose corresponds to the camera pose. From now on, we will only talk about the *camera pose* but it also refers to the light pose. Among all the input images, a series of *key frames* is selected. This selection is based on the number of feature as explained below. The world coordinate system is set as the camera coordinate of the first key frame. The scale of the scene is arbitrarily define as the median scene depth during an initialization step [17]. To compute the camera pose, the algorithm uses homographies and the fundamental matrices as in [16]. If the scene is rather planar, the homography is used. On the other hand, if the scene is non-planar and has a low-parallax, the fundamental matrix is used. To evaluate the type of scene, a score function is computed for each of the two models. As in [16], we use a score function based on symmetric transfer errors [9].

Bundle adjustment is used to refine the results as explained in [17]. In the end, we get the transformation from world to camera coordinate system.

$$\boldsymbol{P}_{cw} = \begin{bmatrix} \boldsymbol{R}_{cw} & \boldsymbol{T}_{cw} \\ 0 & 1 \end{bmatrix}. \tag{1}$$

We note the inverse transformation $\boldsymbol{P}_{wc}$. In addition to the camera pose, we also get a sparse reconstruction of the scene in 3D with *key points*. For all those key points, we have correspondences between all the key frames. With the pose of the camera and the initial correspondences, we can apply the equations related to photometric stereo.

3.3. Photometric Equations

We propose to apply the equations of photometric stereo considering that (1) the camera is moving, and (2) the light source is a near-light source close to the object. With the equations corresponding to these assumptions, we can build a solvable system. This system is used twice in our pipeline as we have shown on Figure 2: (1) to get the normal vector of each point of our initial point cloud, and (2) to densify our point cloud using a multi-view photometric consistency.

3.3.1 Photometric stereo adapted to a moving camera

For each pixel p from the i^{th} image, the standard photometric equation (static camera, lambertian surface) in his simplest form as explained by Woodham in [24] links the pixel intensity $s_i(p)$ with the albedo $\rho(p)$, the light vector $\boldsymbol{l}_i(p)$ (3 dimensional vector corresponding to the direction from the pixel p to the light source) and the normal vector $\boldsymbol{n}(p)$. $\boldsymbol{n}(p)$ and $\boldsymbol{l}_i(p)$ are in the same coordinate system. We consider that the scale is the one computed by ORB-SLAM during its initialization step. Additionally, we add a bias $a(p)$ to remove from the pixel intensity the component that is due to the ambient light. The photometric equation is as below:

$$s_i(p) - a(p) = \rho(p)(\boldsymbol{l}_i(p) \cdot \boldsymbol{n}(p)). \tag{2}$$

Since the camera coordinate system is changing for each image, the natural coordinate systems to use is the world coordinate system (equivalent to the camera coordinate system of the first key frame).

For each image, the origin of the camera coordinate system is the position of the camera. Consequently, it is easier to compute the light vector in each camera coordinate system and then project it into the world camera coordinate system using the rotation matrix $\boldsymbol{R}_{wc}$. Considering the origin of the camera coordinate system as the position of the light source, the light vector in the camera coordinate system $\boldsymbol{l}_i(p)^c$ is then $\boldsymbol{l}_i(p)^c = -\boldsymbol{P}_{cw}.\boldsymbol{x}(p)$ where $\boldsymbol{x}(p)$ is the position in world coordinate system of pixel p (homogeneous coordinate). After projecting in the world coordinate system, we finally can write :

$$\boldsymbol{l}_i(p) = -\boldsymbol{R}_{wc}\boldsymbol{P}_{cw}\boldsymbol{x}(p). \tag{3}$$

3.3.2 Near-light photometric stereo

Equation 2 is valid only if we assume that the origin of the light source is at infinity from the object. In our work, the camera and the light are considered being at the same point and we focus on close-up scenes. As a consequence, we cannot make this assumption of infinite distance. We need to correct the light vector (2) to take this aspect into account. The optimization of the near-light photometric model is a problem that is still investigated and usually includes heavy optimization. The simplest model in terms of computation is a fall-off decreasing with the size of the light vector [20]. We correct the equation (1) and (2) and finally get:

$$s_i(p) - a(p) = \rho(p)(\boldsymbol{l}'_i(p) \cdot \boldsymbol{n}(p)), \tag{4}$$

with

$$\boldsymbol{l}'_i(p) = -\frac{\boldsymbol{R}_{wc}\boldsymbol{P}_{cw}\boldsymbol{x}(p)}{|\boldsymbol{R}_{wc}\boldsymbol{P}_{cw}\boldsymbol{x}(p)|^f}, \tag{5}$$

where f is the fall-off factor. If $f = 0$, we get back to the standard case as described in equation (2). The quadratic model ($f = 2$) and the cubic model ($f = 3$) are often used in the literature [20, 14].

3.3.3 Photometric system

We use the photometric equation (3) to get the normal vectors. We define $N(p)$ as the number of correspondences for the key point p. In theory, the equation (3) is verified for all images and all pixels. We can consider $\rho(p)\boldsymbol{n}(p)$ as a 3-dimensional unknown and separate $\rho(p)$ and $\boldsymbol{n}(p)$ by normalizing. Then, we create the system as in (6). The subscripts $_x$, $_y$ and $_z$ refer to the 3 components of 3D vectors.

$$\begin{bmatrix} \mathrm{s}_1(p) \\ \mathrm{s}_2(p) \\ \mathrm{s}_3(p) \\ \mathrm{s}_4(p) \end{bmatrix} = \begin{bmatrix} \boldsymbol{l}'_1(p)_x & \boldsymbol{l}'_1(p)_y & \boldsymbol{l}'_1(p)_z & 1 \\ \boldsymbol{l}'_2(p)_x & \boldsymbol{l}'_2(p)_y & \boldsymbol{l}'_2(p)_z & 1 \\ \boldsymbol{l}'_3(p)_x & \boldsymbol{l}'_3(p)_y & \boldsymbol{l}'_3(p)_z & 1 \\ \boldsymbol{l}'_4(p)_x & \boldsymbol{l}'_4(p)_y & \boldsymbol{l}'_4(p)_z & 1 \end{bmatrix} \begin{bmatrix} (\rho(p)\boldsymbol{n}(p))_x \\ (\rho(p)\boldsymbol{n}(p))_y \\ (\rho(p)\boldsymbol{n}(p))_z \\ \mathrm{a(p)} \end{bmatrix}. \tag{6}$$

The unknown values are the scalar $a(p)$ and the 3D vector $\rho(p)\boldsymbol{n}(p)$. It corresponds to 4 unknowns. For a given pixel p, if we get at least $N(p) = 4$ points of view, it is possible to create a solvable system. If there is no ambient light ($a(p) = 0$), we can reduce the system to 3 unknowns. We will stick to the general case for now.

In theory, $N(p) = 4$ equations are enough but we consider solving the photometric system only if at least $N(p) = 10$ correspondences are found. Also, we want for each i $s_i(p) > s_{min}$ where s_{min} is a fixed threshold defining the minimum pixel intensity to be considered relevant in terms of intensity information. We then solve this equation using the least-square method. In our method, we use this equation in two different steps of our process. For the process of estimating the normal vectors, there is no particular difficulty since we use the correspondences between the key points found by our SLAM part. These estimated normal vectors are used to get an initial direction for each key point. The second time that we use this system is in the densification process. In this part, we astutely reverse the problem, and use it to find the missing correspondences.

3.4. Densification

So far, we got a set of key points, and after a first usage of the photometric equation, we got an initial estimation of the normals of those points. Now, our objective is to densify our 3D point cloud, which is done using a recursive approach.

First, we apply a 3D Delaunay Triangulation to generate a list of triangles [7]. Inside those triangles, we have no information. The inside is flat, which is unlikely to be the true shape of the object to reconstruct. We want to fill in the inside of those triangles with as many points as possible to reconstruct details. For each initial triangle, we compute the gravity point G_0.

Thanks to the normal vectors previously computed with photometric stereo, we estimate a searching direction (only one degree of liberty). This direction is simply computed as

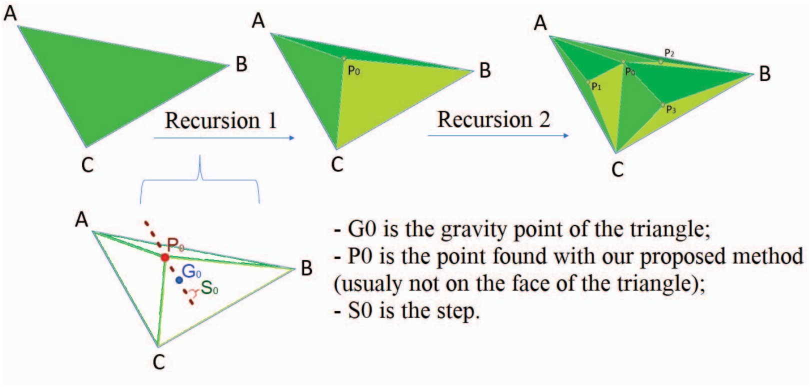

A
B
C
Recursion 1
A
P0
B
C
Recursion 2
A
P2
P0
P1
P3
B
C
A
P0
G0
S0
B
C
- G0 is the gravity point of the triangle;
- P0 is the point found with our proposed method
(usualy not on the face of the triangle);
- S0 is the step.

Figure 3. Process of densification. We start from a triangle whose vertices are key points found by ORB-SLAM. We slide across a line that cross the gravity point G_0, and whose direction is the mean value of the vertexes' normal vectors. The step is S_0 at first step. P_0 is the most reliable point. We then repeat the process considering P_0 as a vertex of 3 new sub-triangles

the mean value of the normal vectors of each vertex. Then, we slide along the axis created by this direction.

The initial sliding step S_0 is defined as a tenth of the smallest side of the current triangle. When the point is in its true position on the surface, it means 2D reprojections on each image should lead to corresponding points. If the points are corresponding, the photometric constraint (4) should be satisfied. Consequently, by picking the candidate which minimizes the photometric error $e(p)$, we can identify the true position of the point on the surface without knowing the correspondences. If the point is reliable enough, it is added to the mesh as P_0, and can be used as a new vertex, and thus creates 3 sub-triangles. Inside each sub-triangle, we compute a new gravity point G_1 and a new sliding step S_1 to add a new point P_1. The process can then be repeated again and again.

$$e(p) = \frac{1}{N_p}.\sqrt{\sum_{i}^{N_p}(s_i(p) - \widetilde{s}_i(p))^2}. \tag{7}$$

$$d(p) = \frac{1}{\binom{N_p}{2}}.\sqrt{\sum_{i,j,i\neq j}^{N_p}(s_i(p) - s_j(p))^2}. \tag{8}$$

A point is considered reliable enough when the two following conditions are met. First, the error $e(p)$ has to be inferior to a minimum error called τ_e. $\widetilde{s}_i(p)$ corresponds to the recomputed pixel intensity using the solution of our system. Secondly, the variation $d(p)$ for pixel p between all the pixel intensities of the system should not be too wide (inferior to τ_d) to avoid discontinuities. Outliers are also rejected by applying RANSAC. Each key point of the initial point cloud is also evaluated based on the same error conditions, which leads to the removal of some points in the initial point cloud.

During the 2D reprojection, some triangles might overlap with the background on some frames, which leads to useless or even wrong new points. The threshold s_{min} mentioned in Section 3.3.3 is a way to address this issue. A concrete example will be provided with the cone dataset in Section 4.5.

As we mentioned before, this approach is recursive. For each initial triangle, we subdivide into 3 triangles which are also subdivided 3 times etc... We perform this process 5 times. Consequently, we can create a maximum of 3^5 = 243 subtriangles for each initial triangle. In practice, we never reach this maximum since there are always some points that are not reliable. Once all the consistent triangles are computed, we get a densified point cloud as our final output.

4. Experimental results

We evaluate our proposed method with real datasets. We used the quadratic model ($f = 2$ in Equation (5)). We experimentally fixed $\tau_e = 0.04$, $\tau_d = 50$ pixels and $s_{min} = 50$.

4.1. Camera and Torchlight specifications

We captured the data using a Huawei P20 smartphone. This smartphone uses Leica optics and a LED flash. The torchlight and the camera are positioned at around 1 cm to one another. It is equipped with a 50 mm f/1.8 lens. To acquire the images, we used the Android application FooteJ which allows us to manually control all the parameters of the camera. We conducted our experiments with a 1/60 shutter, an ISO of 1200 and a focal of 2.4 mm. For the cone dataset, we used an ISO of 200 because of the whiteness of the surface. Beforehand, we calibrated the camera using a chessboard [27]. Here as well, we used 960x720p images saved in .png file format. As it is a real case, we have no information about the real fall-off function.

Box	COLMAP	ORB-SLAM	Our method
KF / Total	NA	16/860	16/860
Points count	NA	9,763	105,276
Time	NA	∼1 min	19 min

Cushion	Colmap	ORB-SLAM	Our method
KF / Total	832/832	11/832	11/832
Points count	10,840	5,531	22,2941
Time	∼14h	∼1 min	12 min

Cone	COLMAP	ORB-SLAM	Our method
KF / Total	706/706	15/706	15/706
Points count	6547	4726	13,691
Time	∼22h	∼1 min	9 min

Table 1. Comparison between COLMAP, ORB-SLAM and our method to highlight the densification. KF corresponds to the number of key frames. COLMAP could not render a proper result for the box dataset after more than 30 hours of computation

4.2. Datasets

If a part of the scene contains many features (such as a textured wall), our ORB features will mainly focus on those points. Consequently, since we do not want to focus on simple walls, we avoid textures on the walls and the floor if there are some. We also avoid ambient light ($a(p) = 0$) .

We used 3 different objects: a box, a cushion and cone-like shape. Note that the box and the cushion have some textures on it. Concerning the object with the cone-like shape, it is completely white which makes it challenging to reconstruct with SLAM. During the acquisition, the camera is zigzagging around the object at 180 degrees.

Table 1 displays for each dataset a quantitative comparison between COLMAP, ORB-SLAM and our proposed method. All experiments where conducted on the same CPU.

4.3. Dataset with flat surfaces

Figure 4 shows some typical frames of the box dataset and the point cloud of our algorithm. The box was placed at a distance from the camera so that the object is blurry. With this experiment, we can prove that our algorithm still works when some pixels are ambiguous due to bluriness.

The white point cloud (Figure 4 (c)) is the point cloud obtained with the standard ORB SLAM algorithm [17]. As we can see, SLAM can generate the general shape, but it is sparse and it cannot properly find the border of the surfaces. The green point cloud is obtained with our technique. Our result has a higher density of points in comparison with ORB-SLAM. Note that some parts are more densified than others. While the green point clouds emphasizes the densification, the normal map highlights the shape of the box.

We could not generate a similar point cloud with COLMAP. Whenever we tried, COLMAP failed to provide a proper point cloud. We reason that the bluriness of the images might be a hindrance for the computation. Unlike COLMAP, our proposed method is not hindered by such problems.

This example shows that our proposed method gets properly the general shape of the object, which is not the case with SLAM. It also provides a higher density of points. This type of flat surface is a simple case, so we tried with other surfaces.

4.4. Dataset with curved surfaces

On the Figure 5, the point clouds obtained with the cushion dataset are displayed. We chose this cushion to test curved surfaces, as the cushion has two different kinds of curvatures (top part and lateral part). The shakiness of the user and the blurriness of some frames do not disturb our proposed method.

As we can see, our proposed method significantly improves the density of the point cloud in comparison with SLAM. Besides, while the ORB-SLAM's point cloud is ambiguous for some parts of the surface, the photometric stereo helps us to remove the inconsistent points as explained in Section 3.4, and thus, get a thin surface. The white points on the point clouds generated by our proposed method correspond to points found by SLAM and consistent enough to be kept.

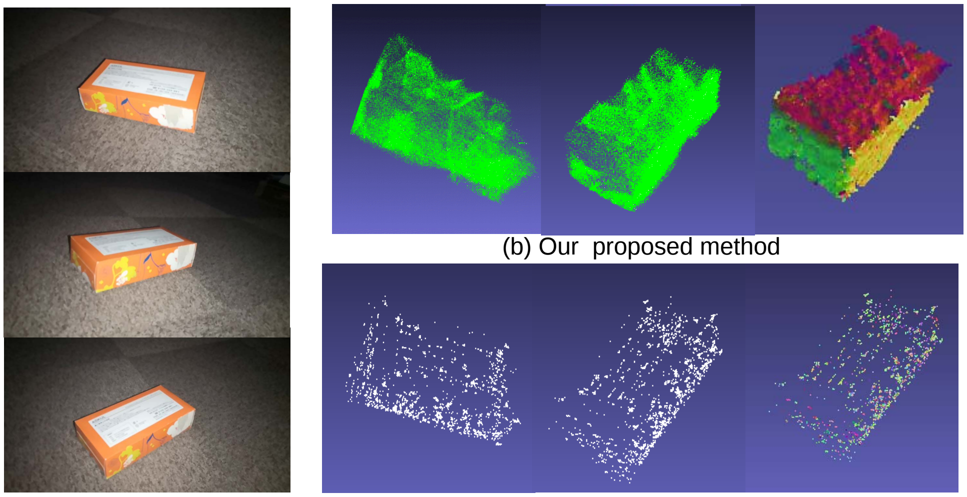
Besides, our proposed method is better than COLMAP to render the true shape of the object. The rendered point cloud from COLMAP seems smooth and homogeneously dense, nonetheless, it wrongly curved the shape. In Figure 5, we can notice that the real cushion is not as round as COLMAP output, but way closer the shape of our proposed method. COLMAP does not distinguish the two kinds of curvatures on the cushion while our proposed method does.

4.5. Dataset with both flat and curved surfaces

The third dataset is a more challenging surface. We display the result on Figure 6. It is a kind of cone with a cuboid on top of it. The object is completely white. Since ORB-SLAM had difficulties to find key points on the top of the cone, we added a red marker at this position and 3 small thin yellow markers to make the initial point cloud computation a bit more consistent.

One of the other challenging aspects of this surface is the occlusions due to the cuboid overlapping with the cone. For this dataset in particular, the minimum intensity threshold s_{min} explained in Section 3.4 was crucial to avoid wrong points between the cuboid and the base of the cone.

Due to the sparseness of the point clouds, the computation of the normal based on the closest neighbour does not render the perfect orientations of the surfaces (for all the techniques). Nonetheless, we can see that we obtained local densifications. For example, the densification is more important on the right part than on the left part of the cuboid.


(b) Our proposed method

(a) Typical input frames

(c) ORB-SLAM

Figure 4. Densification result obtained using the box dataset. With the same input frames (a), we generated a point cloud, using our proposed (d) and ORB-SLAM (b). The last column represents a smoothed colored map to visualize the shape of the box.

We explain this difference of density by the fact that the initial time required by ORB-SLAM to initialize the poses is higher than for the box and cushion dataset. ORB-SLAM particularly struggles with this partially textureless object. Our proposed idea to add photometric stereo is here to drastically diminish the impact of this struggle in the final output.

5. Discussion and further work

By combining ORB-SLAM and photometric stereo, it is possible to highly benefit from these two techniques. First, photometric stereo needs accuracy and robustness which is provided by ORB-SLAM; the user can do rotation around the object and be shaky without altering the results. It does not require any a priori knowledge regarding the poses of the camera and the light. Thanks to the selection of key frames and key points by ORB-SLAM, it is possible for our photometric computation to focus only on images of interest. Also, ORB-SLAM performs poorly with textureless surfaces and dark environments which are cases where photometric stereo performs well. However, ORB-SLAM and photometric stereo's opposition on the texture conditions leads to a difficulty: we need ORB-SLAM to be able to track key points, which means purely textureless surfaces cannot be used. Only partially textureless surfaces allows ORB-SLAM to track points.

In terms of computation time, we managed to get a density of point similar to COLMAP with never more than 2% of its computation time, as Table 1 shows. Besides, even though COLMAP renders good-looking surfaces, the real shape is usually not as curved as what COLMAP thinks it is. Our proposed method provides a more realistic rendering for curves, in a way smaller computation time.

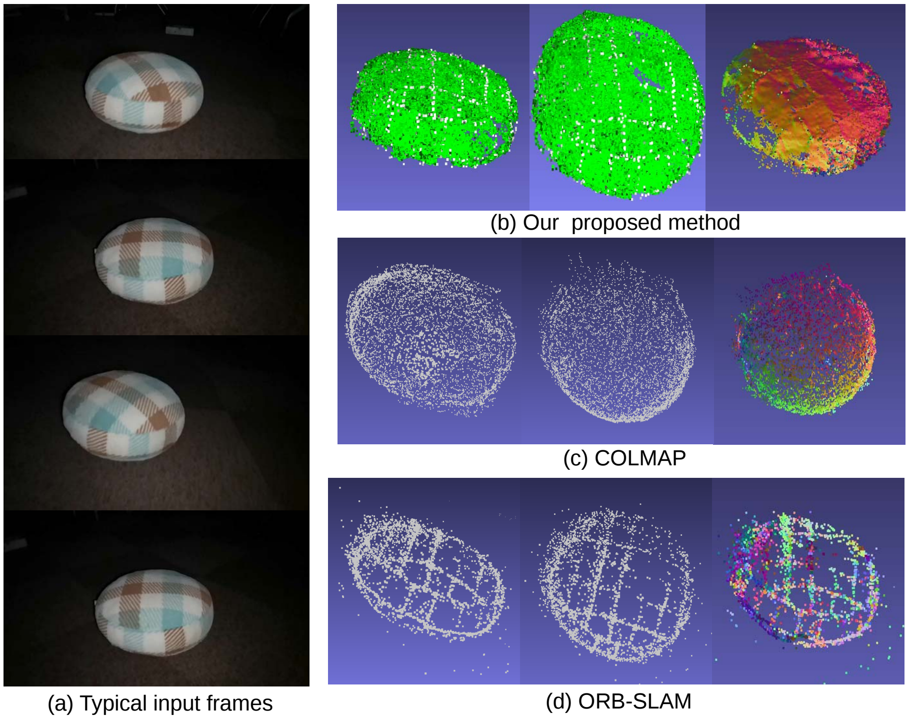
We improve significantly the density of a point cloud that can be obtained with a mere use of ORB-SLAM. Our experiments with real objects tend to prove the feasibility of real reconstruction with a simple smartphone in dark environments. Our field of interest was not dealing with non-lambertian surfaces and shadows. Different optimization techniques has been developed to target those issues [22] [2]. Using those techniques will probably increase the versatility of this smartphone use of photometric stereo.

6. Conclusion

Our method grants access to smartphone users with the acknowledged technique of photometric stereo. Thanks to ORB-SLAM, we created an automated process for a simplified usage; no beforehand knowledge concerning the camera pose or light pose is required, and we obtain an initial sparse 3D scene. Starting from this sparse scene, we use a photometric system to compute a sparse set of normal vectors. Then, we densify our 3D scene using multi-view photometric consistency. We obtained promising results with real scenes and put forward the potential of photometric stereo to be complementary with the ORB-SLAM difficulties to reconstruct dark scenes and partially textureless object.

Acknowledgment

A part of this work is supported by JSPS KAKENHI Grant Number JP17H01768.



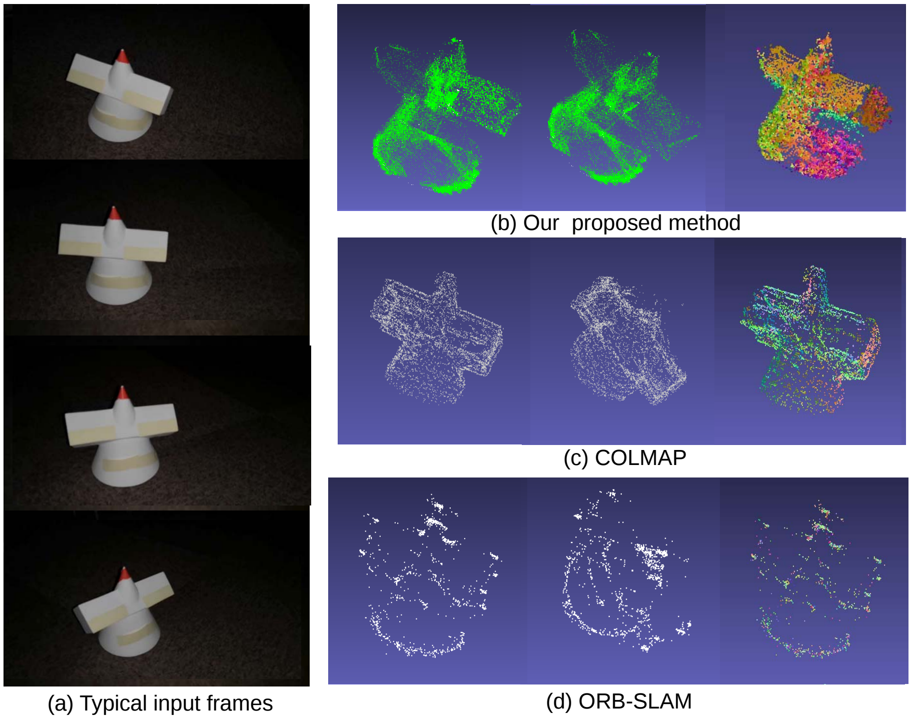
(b) Our proposed method

(c) COLMAP

(a) Typical input frames

(d) ORB-SLAM

Figure 5. Point clouds obtained using the cushion dataset. With the same input frames (a), we generated a point cloud, using our proposed (b) and compared with COLMAP (c) and ORB-SLAM (d). The last column represents a smoothed colored map to visualize the shape of the cushion.



(b) Our proposed method

(c) COLMAP

(a) Typical input frames

(d) ORB-SLAM

Figure 6. Results obtained using the cone dataset. With the same input frames (a), we generated a point cloud, using our proposed (d) and compared with COLMAP (b) and ORB-SLAM (c). The last column represents a smoothed colored map to visualize the shape of the cone.

References

[1] N. Birkbeck, D. Cobzas, P. Sturm, and M. Jagersand. Variational shape and reflectance estimation under changing light and viewpoints. *European Conference on Computer Vision (ECCV)*, pages 536–549, 2006.

[2] M. Chandraker, S. Agarwal, and D. Kriegman. Shadowcuts: Photometric stereo with shadows. *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[3] D. Cho, Y. Matsushita, Y.-W. Tai, and I. S. Kweon. Semi-calibrated photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 72–87, 2018.

[4] D. F. Denis Simakov and R. Basri. Dense shape reconstruction of a moving object under arbitrary, unknown lighting. *ICCV*, 3:1202–1209, 2003.

[5] J. Engel, T. Schps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. *Proceedings of ECCV*, pages 834–849, 2014.

[6] H. Fan, L. Qi, J. Dong, G. Li, and H. Yu. Dynamic 3d surface reconstruction using a hand-held camera. *Annual Conference of Industrial Electronics Society, (IECON)*, pages 3244–3249, 2018.

[7] L. D. Floriani and E. Puppo. A survey of constrained delaunay triangulation algorithms for surface representation. *Issues on Machine Vision. International Centre for Mechanical Sciences*, 307, 1989.

[8] M. Grochulla and T. Thormahlen. Combining photometric normals and multi-view stereo for 3d reconstruction. *Proceedings of the 12th European Conference on Visual Media Production*, 2015.

[9] R. Hartley and A. Zisserman. Multiple view geometry in computervision. *Cambridge University Press*, 2004.

[10] C. Hernandez, G. Vogiatzis, and R. Cipolla. Multi-view photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2000.

[11] T. Higo, Y. Matsushita, N. Joshi, and K. Ikeuchi. A handheld photometric stereo camera for 3-d modeling. *IEEE 12th International Conference on Computer Vision*, 12(1):1234–1241, 2009.

[12] A. Huletski, D. Kartashov, and K. Krinkin. Evaluation of the modern visual slam methods. *Proc. of the AINL-ISMW FRUCT Conference*, pages 19–25, 2015.

[13] I. Z. Ibragimov and I. M. Afanasyev. Comparison of ros-based visual slam methods in homogeneous indoor environment. *IEEE 14th Workshop on Positioning, Navigation and Communications (WPNC)*, 2017.

[14] M. Liao, L. Wang, R. Yang, and M. Gong. Light fall-off stereo. *Proceedings of CVPR*, 2007.

[15] A. Maki, M. Watanabe, and C. Wiles. Geotensity: Combining motion and lighting for 3d surface reconstruction. *International Journal of Computer Vision*, 48:75–90, 2002.

[16] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[17] R. Mur-Artal and J. D. Tardos. Orb-slam2: an open-source slam system formonocular, stereo and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[18] D. Nehab, S. Rusinkiewicz, J. Davis, , and R. Ramamoor-thi. Efficiently combining positions and normals for precise 3d geometry. *inACM transactions on graphics (TOG)*, 24:536–543, 2005.

[19] S. Park, T. Schops, and M. Pollefeys. Illumination change robustness in direct visual slam. *IEEE international conference on robotics and automation (ICRA)*, pages 4523–4530, 2017.

[20] Y. Queau, T. Wu, and D. Cremer. Semi-calibrated near-light photometric stereo. *Scale Space and Variational Methods in Computer Vision: 6th International Conference*, 6(1):656–668, 2017.

[21] R. Sabzevari, A. D. Bue, and V. Murino. Structure from motion and photometric stereo for dense 3d shape recovery. *International Conference on Image Analysis and Processing*, pages 660–669, 2011.

[22] K. Schluns and O. Wittig. Photometric stereo for non-lambertian surface using color information. *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[23] J. L. Shnberger and J.-M. Frahm. Structure-from-motion revisited. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[24] R. J. Woodham. Photometric stereo: A reflectance map technique for determining surface orientation from image intensity. *Proceedings of the SPIE 0155, Image Understanding Systems and Industrial Applications I*, 1979.

[25] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1), 1980.

[26] L. Zhang, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multi-view stereo. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003.

[27] Z. Zhang. A flexible new technique for camera calibrations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[28] Z. W. Zhenglong Zhou and P. Tan. Multi-view photometric stereo with spatially varying isotropic materials. *Computer Vision and Pattern Recognition (CVPR)*, 13, 2013.