Merging SLAM and photometric stereo for 3D reconstruction with a moving camera

Remy Maxence
Hyper Vision Research Laboratory
Keio University
Yokohama, Japan

Saito Hideo
Hyper Vision Research Laboratory
Keio University
Yokohama, Japan

Uchiyama Hideaki
Laboratory for Image and Media Understanding
Kyushu University
Fukuoka, Japan

Kawasaki Hiroshi
Computer Vision & Graphics Laboratory
Kyushu University
Fukuoka, Japan

Nozick Vincent
Laboratoire d'Informatique Gaspard Monge
Paris-Est Marne-la-Vallée University
Marne-la-Vallée, France

Thomas Diego
Laboratory for Image and Media Understanding
Kyushu University
Fukuoka, Japan

***Abstract*—By using photometric stereo it is possible to compute the normal of the surfaces of a scene. This paper applied photometric stereo using a moving camera. This implementation is meant for dark environment enlightened by the torchlight of the camera. Since the photometric stereo is applied while the camera is moving, it requires a camera pose estimation. This estimation is performed thanks to ORB features used in a similar way as ORB SLAM algorithm. A first estimation of the depth map is provided by stereo matching.**

Keywords—photometric stereo, stereo, SLAM, surface, matching, ORB features

I. INTRODUCTION

AR technologies such as ARCore usually focus on an estimation of the plane surfaces, avoiding to fully reconstruct the scene they are applied in. In addition, they perform poorly with close-up scenes and with poor light conditions. Consequently, some AR concepts are still far from being implementable because they require round surfaces or dark environment. For example, implementing an AR guide to exit a building after a power outage is a challenging implementation with the current technologies.

Our system uses photometric stereo to generate an accurate estimation of the normal vectors and the albedo of the surfaces. Usually, **photometric stereo** is fixed and the light source is moving. However, it is possible to apply photometric stereo even if the camera is moving. A simple way to do so is to move the light and the camera simultaneously as in [1]. This method suits perfectly with the usage of the camera and the torchlight of a smartphone which moves altogether. The challenge consists in estimating the camera pose (and consequently the torchlight pose) as the user is moving. This can be done thanks to **SLAM algorithm**. In this research, the camera pose is obtained thanks to a key points-oriented approach similar to ORB SLAM [2]. By using **stereo matching**, it is possible to get a first estimation of the depth of the scene [3]. Finally, with the camera/light poses and an initial depth estimation, it is possible to apply photometric stereo to finally get an accurate estimation of the normal and the albedo of the scene. The full pipeline is summed up in Fig. 1.

Our contribution consists in (1) the creation of an algorithm combining SLAM and photometric stereo, (2) the adaptation of photometric stereo for a moving user and (3) a reconstruction technique that can be applied in dark environment.

A. *Previous works*

SLAM technologies are usually used for odometry and can be used with a monocular camera and provide really interesting results as long as some constraint to get a right acquisition are respected. With images simply 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) as in [4] and others are sparse (key-point oriented) as in [2]. In our case, we use a sparse approach inspired by ORB SLAM. The reason is to get a fast process that is robust to various camera movement.

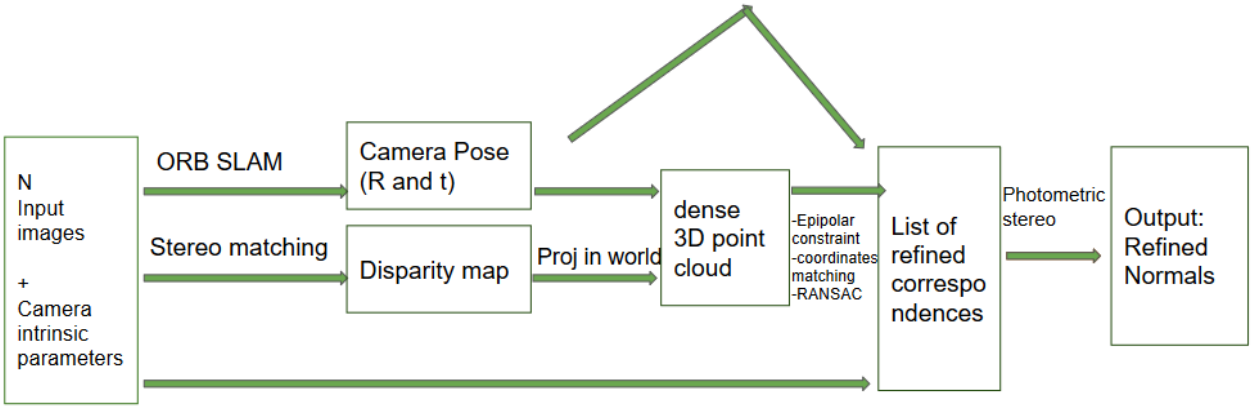

Fig. 1. *Key steps of our pipeline.*

Photometric stereo is an approved technology that uses a pixel-wise approach to reconstruct the normal and the albedo of the technology. Based on the variation of the light intensity, it performs well when the good conditions are respected (fixed camera, moving light, fixed small object, dark light conditions). Here, we would like to use the most constraint-less approach for photometric stereo. We want to apply photometric stereo with a moving camera. To do so, we need a large variation of camera movement which explains why we require the robustness of ORB features (see [5]).

Finally, since we require an initial depth estimation, we use another often-used technology which is stereo matching [3]. By using two images of the same scene slightly translated, it is possible to compute easily a depth map of the scene.

II. Proposed Method

A. Image acquisition

The idea is to acquire the data as if we are using a smartphone with the torchlight turned on. The camera is moving around the objects with a large variation of movement to have a large panel of light orientation. We need a large number of frames to get an accurate result with the photometric stereo. In our experiments, we have the control of the colorimetric and compressing settings. The idea is to get images that can directly feed our ORB SLAM based algorithm without requiring additional preprocessing of the images.

B. ORB SLAM based-algorithm

The algorithm extract ORB features in all images to find a set of correspondences between all the images as explain in [2]. Among all the images, a series of key frames based on the number of correspondence is found. To compute the camera pose, the algorithm uses the homographies between the images and the fundamental matrixes. 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. The process is able to work in real-time. A set of Key Frames is obtained and for each of them, we are able to get the camera pose between world and camera coordinate system (R_{cw}, T_{cw}) and vice-versa (R_{wc} , T_{wc}). We also have a sparse reconstruction of the scene in 3D and a mapping of those points with the pixels of each Key Frame. Note that our reconstruction is unscaled.

C. Stereo Matching

With this camera pose, we are able to get the rotation $R_{1\to2}$ and the translation $T_{1\to2}$ between an image 1 and an image 2 :

$$R_{1\to2} = R_{cw2}R_{wc1} \quad (1)$$
$$T_{1\to2} = R_{cw2}T_{wc1} + T_{cw2} \quad (2)$$

This is then used to apply stereo matching. After rectification of the images, we can get a disparity map and consequently a depth map estimation. We decide to apply stereo matching for each Key Frame as a left image and a non-Key Frame as a right image. After rectification of the images, we apply a semi-dense block matching technique to get a disparity map. From this disparity map, we are able to get a depth map and to project our image in 3D. The depth map is used as a first estimation for the photometric stereo. The 3D projection is used for correspondences. We get a series of 3D point clouds for each Key Frames. By projecting everything in the world coordinate system, we are able to get a dense point cloud with overlapping points. For each point we store the following data: the image index where it was extracted from and the intensity of the pixel used for this point. By using overlapping points, we get a series of pixel intensities for each Key Points. As we will explain in II. D, if we get at least 4 overlapping, we can apply photometric stereo. Overlapping is estimated based on the Euclidian norm. To refine the result, we also use the 2D projections to check if the overlapping is preserved. Ultimately, we only keep the key points that corresponds to fair correspondences. We use the epipolar constraint to check if the points are corresponding. The fundamental Matrix is computed using the intrinsic and extrinsic parameters. We ensured a rank two to this matrix by using a SVD decomposition to set the last Eigen Values to 0.

D. Photometric Stereo

To represent this 3D reconstruction, we create a list of 3D points such as each member can be represented as X, Y, Z, V where X, Y, Z are the coordinates of the 3D points and V a vector containing a series of tuples. Each tuple contains the information of different images that will be used for photometric stereo. The representation of a tuple is (i, p) where i is the index of the image used to compute this 3D point and p the corresponding pixel intensity. If the length of V is superior to 4, we can apply photometric stereo. The classic equation used for photometric stereo with lambertian surfaces is:

$$p_i = \rho_i \; l_i \; (R_{cw}n_i) + a_i \quad (3)$$

where p_i is the intensity of pixel i, ρ_i the albedo of i, l_i the direction of the light vector of the torchlight toward i, n_i the normal vector in the world coordinate system of i and a_i the ambient light as received by i. Since the camera is moving, we need to compute for each pixel and each camera position a different light vector as followed:

$$l_i = -(R_{cw}x_i + T_{cw}) \quad (4)$$

where x_i is a first estimation of the normal of the corresponding pixel i (world coordinate). x is projected from the world coordinate system to the camera coordinate system where it was extracted. Since the camera is in the origin of the camera coordinate system, it comes that l is the vector that is directed from the pixel on the surface to the camera. For near photometric stereo, the equation (1) needs to be corrected into (5) by a quadratic or cubic inverse as explained in [6] where f = 2 or 3 depending on the selected model. If the model is not near photometric stereo, we set

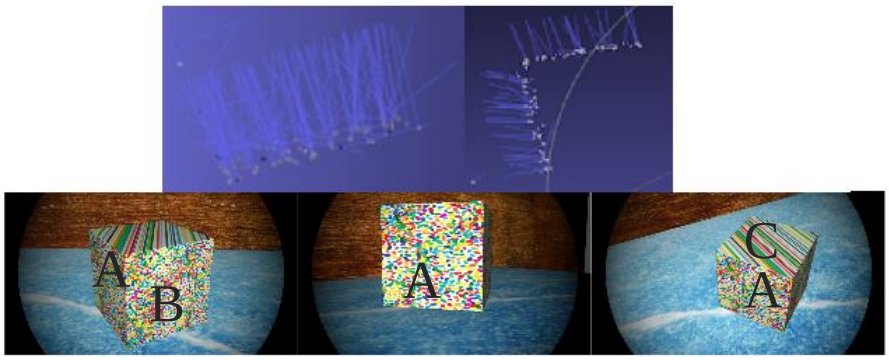

Fig. 2. *Bottom: example of 3 input images (bottom) with a textured cube. Up left: normal vectors for the face A with 1000 input images. Up right: normal vectors for the two faces A and B with 200 input images.*

$$p_i = \rho_i \frac{l_i}{|l_i|^f} (R_{cw} n_i) + a_i \qquad (5)$$

f = 0 which corresponds to the classic case of photometric stereo [7].

We consider the notation (6) where k means it is the index of the image the value was extracted from. (6) is a vector whose size is 3.

$$l'^k_i = R^k_{cw} * \frac{l^k_i}{|l^k_i|^f} \qquad (6)$$

With at least 4 different pixels corresponding to the same point in space, we get our photometric system that we can solve (7). The light vectors also need to be independent. Note that if we consider that there is no ambient light, we can solve the system with only 3 correspondences.

$$\begin{bmatrix} p^1_i \\ p^2_i \\ p^3_i \\ p^4_i \end{bmatrix} = \rho_i \begin{bmatrix} l'^1_{i,x} & l'^1_{i,y} & l'^1_{i,z} & 1 \\ l'^2_{i,x} & l'^2_{i,y} & l'^2_{i,z} & 1 \\ l'^3_{i,x} & l'^3_{i,y} & l'^3_{i,z} & 1 \\ l'^4_{i,x} & l'^4_{i,y} & l'^4_{i,z} & 1 \end{bmatrix} \begin{bmatrix} n^1_{i,x} \\ n^2_{i,x} \\ n^3_{i,x} \\ a_i \end{bmatrix} \qquad (7)$$

For better result, we solve this system using the least-square method with as many equations as possible instead of just 4. By solving this system, we get $\rho_i n_i$. The norm of the results, provide ρ_i and the normalized result provides n_i. We finally get normal vectors. We do this computation for all the pixels that have enough correspondences. To remove the outliers, we use a RANSAC approach.

III. RESULTS

We applied photometric stereo for sparse points which provides fair result. To get the full control of the camera movement, we generated a 3D scene using Blender where we can control the light and the camera. We tried several experiments using different conditions.

A. *Influence of the number of frame*

First, we investigate the influence of the number of frames and correspondences for flat surfaces. The object we used was a cube and we made the camera turned around the object. We chose a texture colorful but with light colors to facilitate the key points tracking and the photometric stereo. The light is virtually put far away of the scene but it still follows the camera movement. In this case, f=0. We tried to use different number of frames to see how the normal vectors evolve. For each experiment, we were oscillating around the cube to be sure that we have a wide variation of the light vectors for each axis. Without this large variation, the normal cannot be computed properly since the rank of the light matrix will not be 3. You can the result of one experiment on Fig. 2.

As you can see on figure 3, to recover the correct normal vectors, in practice, we need way more than 3 correspondences to get result. The results start to be fair when you have at least 8 correspondences. This method cannot work with just a few images.

Nb Key frame / nb frame	Nb of average correspondences after RANSAC	Nb good vectors (at most 1% shift from the expected direction) / Nb normal estimated
40/1000	11	247 /300
30/400	8	100/202
24/200	6	32/108
18/100	4	9/60
9/50	3	2/50

Fig. 3. *Evolution of the percentage of good normal depending on the number of frames used as input. The number of correspondences corresponds to the size of all the systems we solved. As a reminder, 3 is enough without ambient light.*

B. Comparison with stereo matching based normals,

For another experiment, we used a bunny as the object whose texture is again light and colorful. The camera trajectory is again moving around the object with a large variation on each axis. The Fig.4 shows the camera trajectory and some input images.

We noticed that the number of images has a huge influence on the

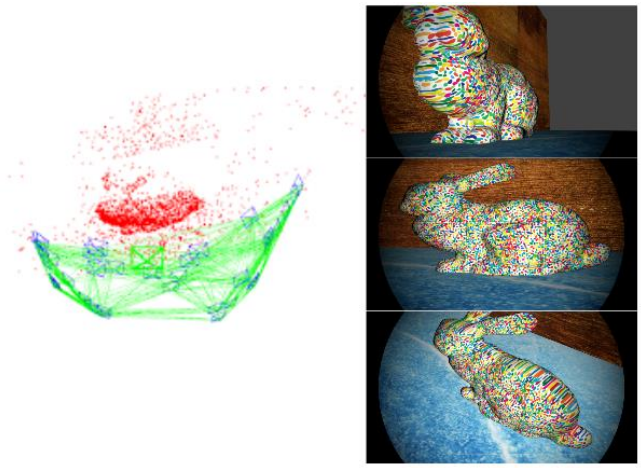

Fig. 4. *On the left: Trajectory of the camera (green lines), Key frames positions (blue rectangles) and key points (red dots). On the right: examples of 3 images used as input out of 1000.*

We compared the rendered normal vectors of our technique with the rendered normal vectors based on the computation of the point cloud estimated by the stereo matching (computed from a normal map). As you can see on Fig. 5, our technique is better to reshape the original shape. With stereo matching solely, the surfaces are estimated way more flat than they are.

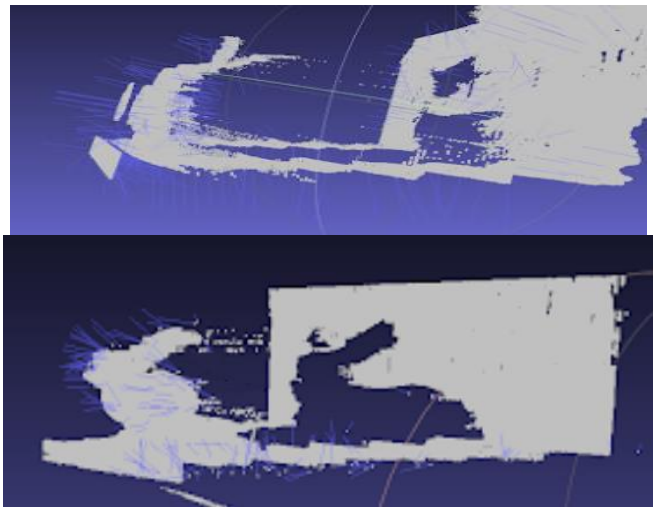

Fig. 5. *Up: the normal estimated thanks to stereo matching. Down: the normal vectors estimated with 1000 images. The grey part corresponds to the 3D reconstruction of the scene with stereo matching based on only two frames.*

IV. Conclusion and next steps

SLAM allows an accurate tracking that gives more freedom to apply photometric stereo by removing the classic fixed camera constraint. Thanks to photometric stereo, it is possible to refine the flatness that the stereo matching usually estimates. This technique is more suitable when there is no ambient light which makes it good in dark environment.

As in classic photometric stereo problem, we require a lot of camera movement and many images as input. The minimum number of correspondences (3 without, 4 with ambient light) is not sufficient this is one it is important to get a large number of correspondences.

Next step for us is to densify our vectors. Because of the poor efficiency of the correspondences found by stereo matching, only a few correspondences can be used for photometric stereo. Photometric stereo while the camera is moving is really sensitive to the correspondences and small shifts of few pixels can make the results wrong. We are now investigating to find a way to get more reliable correspondences. It will in the same time increase the quality of our normal vectors and densify or results.

We also target a pipeline fully applicable with a direct smartphone. Some calculation optimization needs to be addressed to make it possible.

References

[1] T. Higo, Y. Matsushita, N. Joshi and K. Ikeuchi, A hand-held photometric stereo camera for 3-D modeling, *2009 IEEE 12th International Conference on Computer Vision*, Kyoto, 2009, pp. 1234-1241.

[2] Raul Mur-Artal, J. M. M. Montiel and Juan D. Tardos, ORB-SLAM: a Versatile and Accurate Monocular SLAM System, *IEEE Transactions on Robotics*, 2015.

[3] Geiger A., Roser M., Urtasun R. (2011) Efficient Large-Scale Stereo Matching. *Computer Vision – ACCV 2010*, Berlin *Lecture Notes in Computer Science*, 2010, vol 6492.

[4] J. Engel, T. Schöps and D. Cremers, LSD-SLAM: Large-Scale Direct Monocular SLAM, *European Conference on Computer Vision*, 2014.

[5] J. Engel, V. Usenko and D. Cremers, A Photometrically Calibrated Benchmark For Monocular Visual Odometry, *CoRR abs Conference on Computer Vision*, 2016, 1607.02555..

[6] Y. Quéau, T. Wu and D. Cremer. Semi-calibrated Near-Light Photometric Stereo, *Scale Space and Variational Methods in Computer Vision: 6th International Conference, SSVM 2017*, 2017. pp. 656-668

[7] K. Mu Lee, C.-C. Jay Kuo. Shape Reconstruction from Photometric Stereo, *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*., 1992.

[8] H. Hayakawa. Photometric Stereo under a light source with arbitrary motion, *J. Opt. Soc. Am.*, 1994. A **11** 3079-3089