A Comparison of Methods to Reconstruct Static Background in Dynamic environments

陳威宇 (B07505020)

National Taiwan University Department of Engineering Science and Ocean Engineering
Taipei, Taiwan

ABSTRACT

Static background reconstruction is an important factor in various fields such as robotics and VR. Since many decisions depend on knowing which objects are moving and which aren't. For example, in an autonomous vehicle in order to dodge a static obstacle you just need to categorize the space the obstacle in as a obstacle space, whereas to dodge a dynamic object you need to include an estimate of future states to correctly map out where the obstacle would be. Thus, in this report I would be comparing four different methods to detect moving objects based on visual odometry and Simultaneous Localization and Mapping.

1. INTRODUCTION

In recent years using image-based object detection for SLAM has become the norm. As an extensively researched topic, numerous background reconstruction systems have been invented with the purpose of determining the best candidate from a sequence of intensity values to represent the background of each pixel position within a processing period. Most of which utilizes feature extraction partnered with optical flow to determine dynamic and static objects. Yet the methods in which this is implemented are varied. This project would focus on four different methods which are each have a unique approach to the problem. Cao *et al.* proposed a method using the threshold method and a pixel intensity classification algorithm (PIC) to cut out the dynamic components before updating the background [1]. Zhang *et al.* proposed a method using two frames for visual odometry along with PWCnet [4] optical flow neural network to create a 2D scene flow before cutting out the pixels with movement levels above a certain threshold [2]. Zhong *et al.* proposed a method using DNN based object detection to detect objects before using moving probability update to cut out dynamic objects [3].

2. METHOD EVALUATION

2.1 An Effective Background Reconstruction Method for Video Objects Detection [1]

The main idea of the PIC algorithm is that the pixel intensity of the background in a sequence of frames would appear with the maximum probability. Thus, the algorithm classifies each pixel according to the difference between frames after which the pixels with max intensity would be categorized as the background pixels. However, there are several difficulties when trying to implement this method. First of which is that due to the nature of the algorithm depending on a fair quantity of frames, this method would be hard to implement in real time. Second of which is that after a few intervals of the algorithm the threshold value must be recalculated for a overall estimation, making it time consuming to merge separate intervals into one single observation.

Therefore, this paper proposes a method in which to overcome the drawbacks stated above. Which includes choosing *N* frames from the image sequence to construct the background and calculate an array of the pixel intensities of the image data. Then normalize the array for the elements to be from 0 to 1. After that divide the entire space into *M* intervals before finding the interval M_{max} where the interval has the greatest number of pixels that have the highest intensity. Then, making M_{max} the center interval, extend the interval out by 1 interval in both directions and assume that the position of the pixels with maximum intensity at M_{max} are assumed to be background pixels in the extended intervals in the same position and the average of the pixel intensity between the three frames can be regarded as the pixel intensity of the point in the background image. Thus, if there is a sudden change in the pixel intensity that means that a different object appeared or disappeared in that specific scene.

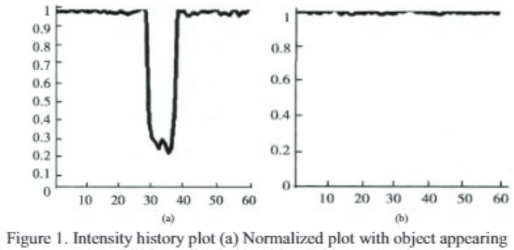


Figure 1. Intensity history plot (a) Normalized plot with object appearing (b) Normalized plot without object appearing

Fig1. Taken from Figure1 [1]

The result of this algorithm is compared to two other methods which are the time average algorithm and the gaussian model algorithm.

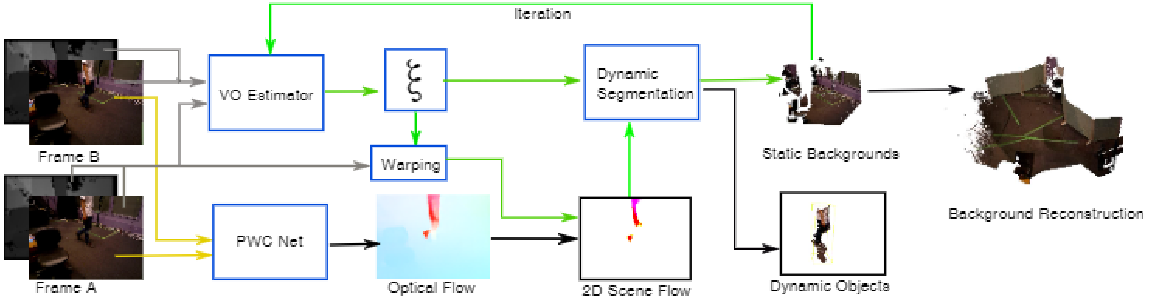


Fig. 1: The proposed FlowFusion framework: Input two continuous RGB-D frames A and B, the RGB images are first fed into PWC-net for optical flow (yellow arrows) estimation. Meanwhile, the intensity and depth pairs of A and B are fed to robust camera ego-motion estimator to initialize the camera motion ξ (introduced in Section III-A). We then warp the frame A to A' with ξ and obtain the projected 2D scene flow (Section III-B), then apply it to dynamic segmentation. After several iterations(Section III-C, the green arrows), the static backgrounds are achieved for reconstruction.

Fig2. Taken from Fig. 1 [2]

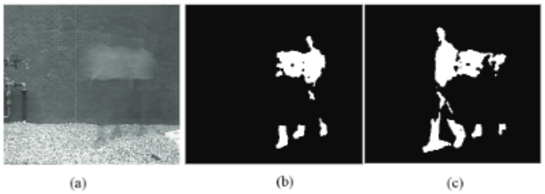


Figure 3 Results of background reconstruction using Time-Averaging algorithm (a) the reconstructed background (b)the 20th object detection (c)the 60th object detection

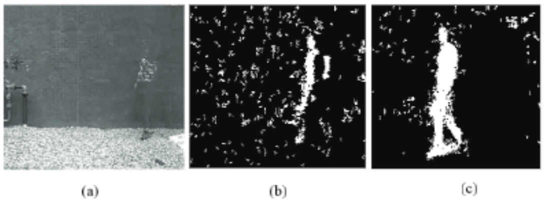


Figure 4 Results of background reconstruction using Gaussian model algorithm (a) the reconstructed background (b)the 20th object detection (c)the 60th object detection

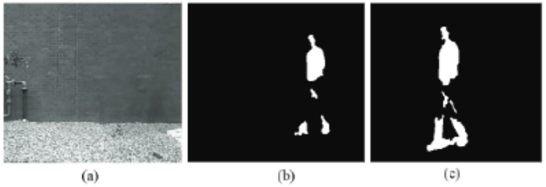


Figure 5 Results of background reconstruction using improved PIC algorithm (a) the reconstructed background (b)the 20th object detection (c)the 60th object detection

Fig3. Taken from Figure3,4,5 [1]

2.2 FlowFusion: Dynamic Dense RGB-D SLAM Based on Optical Flow [2]

The main proposition of this paper is to deal with dynamic objects using an optical flow residual based dynamic segmentation with two RGBD images, which is implemented by two parts. The first is visual odometry whereby we first take the two RGBD input and split it into *I* as intensity image and *D* as depth image. The 3D point cloud can be generated from (I_A, D_A) with pinhole camera model. Then segment the point cloud of image A into *N* clusters according to supervoxel clustering. Treating each cluster V_i as a rigid body. After that we define the rigid motion guess of the frame as ξ.

The second part is estimating optical flow. Although theoretically we can know whether cluster V_i is dynamic or static with ξ, intensity and depth are not ideal parameters for estimation because 1. Since intensity and depth are taken from different cameras, there cannot be perfectly calibrated and would have delay. 2. Depth measurement would result in large amounts of error at the boundary region because of its discrete nature. 3. The depth measurement would be more and more inaccurate when the measurement range get longer. Thus, this paper uses optical flow cross validating with visual odometry [2].

To estimate optical flow between time t and t+1 PWCnet is utilized. PWCnet is a CNN that utilizes feature pyramid, warping layer and cost volume layer which are as follows. First two images are given to a feature pyramid extractor, which generates L-level pyramids of feature representations with the bottom level being the input image. Then the image is put through the first *l* layers which warps features of the second image to the first image using the double upsampled flow from level *l*+1 which is

$$c_w^l(x) = c_2^l(x + up_2(w^{l+1})(x))$$

Where x is the pixel index and the upsampled flow $up_2(w^{l+1})$ is 0 at the top level. For non-translation motion, warping can compensate for some geometric distortions.

Then the layer after warping layers is the cost volume layer which uses the features of the two images to store the matching costs for associating a pixel with its corresponding pixels in the next frame.

$$cv^l(x_1, x_2) = \frac{1}{N}\left(c_1^l(x_1)\right)^T c_w^l(x_2)$$

Where N is the length of vector $c_1^l(x_1)$. After which the next few layers are the optical flow estimator whose input are the

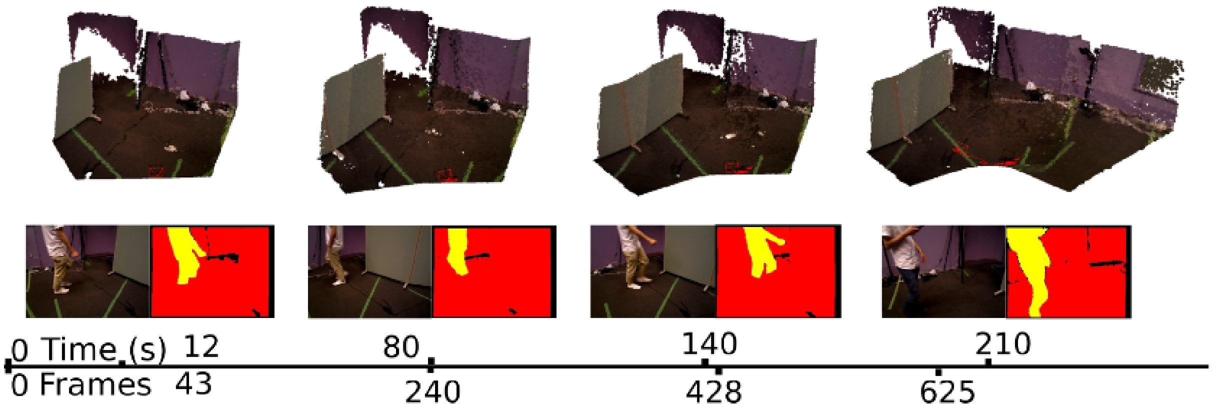


Fig. 6: FlowFusion Experimental Result on HRPSlam 2.1 sequence. The yellow parts are the estimated dynamic objects. In this sequence, an HRP-4 humanoid robot mount one RGB-D sensor firstly moved to his left and then turned rightwards. These datasets contaion abundant fast rotation motions and shaking, which make difficulties to obtain optical flow residual. The feet parts are segmented to the static background, since during the walking phase, the supporting feet on the ground are easily treated as static objects. Although the sweeping feet are moving fast and hold significant optical flow residuals, they are too close to the rigid grounds. Thus they are easily segmented to the the static background due to the graph connectivity.

Fig4. Taken from Fig. 6 [2]

cost volume, features of the first image and the upsampled optical flow from the second image.

The output is the flow of the images. The number of channels at each layer are kept fixed at all pyramid levels. Lastly a context network is used to post-process the flow whose purpose is to take the estimated flow and the features from the optical flow estimator to output a refined flow. The training process is only utilized at the optical flow estimator and context network since warping and cost volume has no learnable parameters [4].

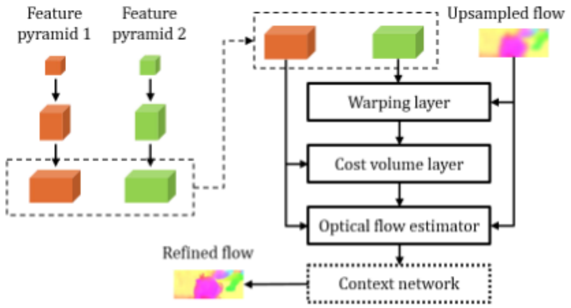


Fig5. Taken from Figure 3 [4]

After we have the ξ from the visual odometry estimator and the 2D scene flow from PWCnet, we can do dynamic cluster segmentation by categorizing the point cloud clusters to dynamic and static clusters. Which be used to reconstruct the background even when the camera is moving.

2.3 Detect-SLAM: Making Object Detection and SLAM Mutually Beneficial [3]

This paper proposes a method to remove moving objects or features that are associated with moving objects, creating an object map of point cloud to reconstruct the static objects that are detected, and to utilize the object map as prior knowledge to improve detection performance. For the implementation Detect-SLAM creates a static object map from a RGB-D camera.

For moving object removal, the system first categorize objects to be either movable or immovable, whereas movable objects are always removed from the frames. For this approach, the key issue regarding efficient segmentation is the efficiency of the object detection algorithm. The algorithm should be fast enough to perform frame by frame detection in real-time so that segmentation could also be performed by a frame-by-frame basis. Thus, this paper devised two strategies to overcome this issue. The system first detects moving objects in keyframes only before updating to the local map, then the system uses the local map to propagate the moving probability of an object by feature mapping.

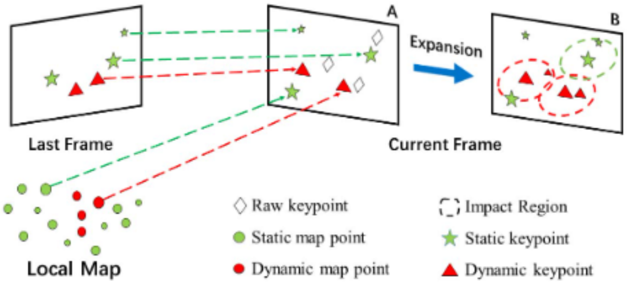


Figure 4. The process of propagating the moving probability frame-by-frame. The size of the points reflects the confidence.

Fig6. Taken from Figure 4. [3]

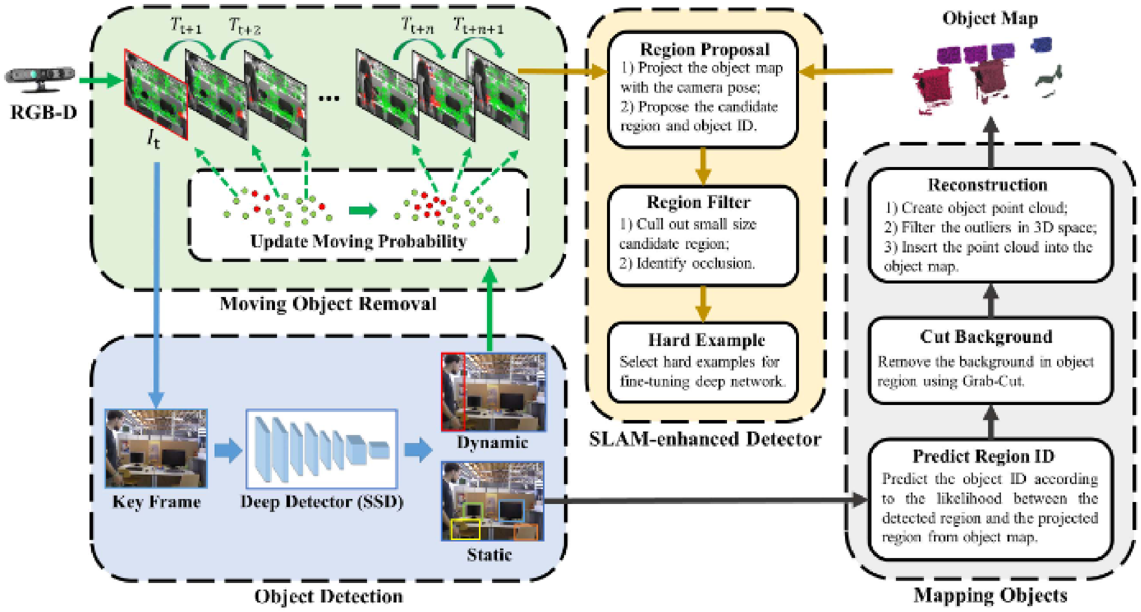

Object Map
RGB-D
T_{t+1}
T_{t+2}
T_{t+n}
T_{t+n+1}
I_t
...
Update Moving Probability
Moving Object Removal
Region Proposal
1) Project the object map with the camera pose;
2) Propose the candidate region and object ID.
Region Filter
1) Cull out small size candidate region;
2) Identify occlusion.
Hard Example
Select hard examples for fine-tuning deep network.
SLAM-enhanced Detector
Reconstruction
1) Create object point cloud;
2) Filter the outliers in 3D space;
3) Insert the point cloud into the object map.
Cut Background
Remove the background in object region using Grab-Cut.
Predict Region ID
Predict the object ID according to the likelihood between the detected region and the projected region from object map.
Mapping Objects
Key Frame
Deep Detector (SSD)
Dynamic
Static
Object Detection

Figure 1. **The framework of Detect-SLAM.** It is composed of *moving object removal*, *object detection*, *SLAM-enhanced detector* and *mapping objects*. The input data are RGB-D images. Moving object removal contains the tracking and local mapping thread of ORB-SLAM. The deep detector (SSD) is running on the GPU, the others are running on the CPU.

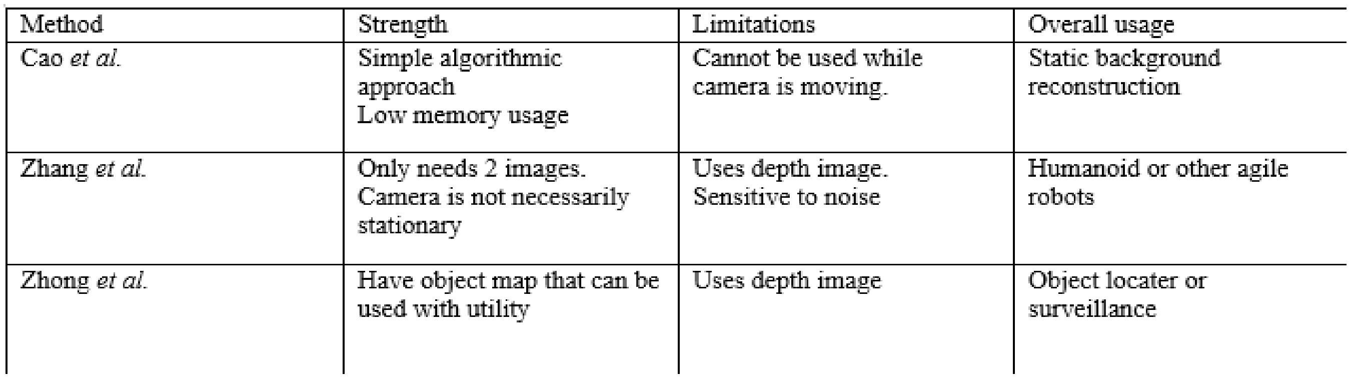
Fig6. Taken from Figure 1. [4]

The probability of a feature point belonging to a moving object is defined as moving probability. Then the feature points get categorized into four states which are high-confidence static (0-25% chance of moving), low-confidence static (25-50% chance of moving), low confidence dynamic (50-75% chance of moving) and high-confidence dynamic (75-100% chance of moving).

Once a new detection result is obtained, the keyframe is inserted into the local map and update the probability. The updated probability of 3D points that has found matching keypoints in the keyframe is

$$P_t(X^i) = (1 - \alpha)Pt - 1(X^i) + \alpha S_t(x^i)$$

Where $Pt - 1(X^i)$ is the moving probability of point X^i after the last keyframe. If X^i is a new point then $Pt - 1(X^i)$=0.5



Method	Strength	Limitations	Overall usage
Cao *et al.*	Simple algorithmic approach Low memory usage	Cannot be used while camera is moving.	Static background reconstruction
Zhang *et al.*	Only needs 2 images. Camera is not necessarily stationary	Uses depth image. Sensitive to noise	Humanoid or other agile robots
Zhong *et al.*	Have object map that can be used with utility	Uses depth image	Object locater or surveillance

3. CONCLUSION

All three methods proposed have corresponding strength and usages regarding dynamic object segmentation.

For [1] the advantage is that it is simple and cost efficient to implement with decent accuracy, but since it relies heavily on the same pixel being in the same position to be able to determine which pixel is background, this method could only be used when both the camera and the environment is near stationary. Also, the relatively low memory and computation usage means that the cost is vastly reduced, which means that you could potentially use multiple setups precisely positioned to create a 3D background like the other 2 algorithms.

For [2] the advantage is that it needs minimum input to create a background reconstruction, with it being possible to reconstruct a background accurately with only 2 frames. The downside to this is that it is costly to implement, requiring an RGB-D camera as well as a decent GPU and the immense calculation required in order to run both visual odometry and PWCnet simultaneously and in real time. This approach is also the most sensitive to noise out of the three approaches. However, this approach is the only approach that can be utilized on a moving platform since it also takes camera motion flow into consideration.

For [3] the method it proposed can be used for background reconstruction but in the paper, it is presented as a static object reconstruction system. While the other two methods can be repurposed to solely mark out dynamic objects, they cannot really determine a static background from a static object since they do not utilize object detection algorithms. Also, this method is the only method out of the three that stores data on a map with the position and image of all visible objects which can be used for multiple different usages such as external motion planning for robots or utilizing CCTV and static surveillance for other more complex tasks than just recording.

7. REFERENCES

[1] Cao, L., & Jiang, Y. (2012). An Effective Background Reconstruction Method for Video Objects Detection. 2012 Third International Conference on Networking and Distributed Computing.

[2] Zhang, T., Zhang, H., Li, Y., Nakamura, Y. and Zhang, L., (2020). FlowFusion: Dynamic Dense RGB-D SLAM Based on Optical Flow. In: IEEE International Conference on Robotics and Automation (ICRA). pp.1-7.

[3] Zhong, F., Wang, S., Zhang, Z., Chen, C., & Wang, Y. (2018). Detect-SLAM: Making Object Detection and SLAM Mutually Beneficial. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV).

[4] Sun, D., Yang, X., Liu, M.-Y., & Kautz, J. (2018). PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.