OBJECT LABELING IN A MULTICAMERA ENVIRONMENT

Jordi Cenzano

UPC (Universitat Politécnica de Catalunya)

ABSTRACT

In this paper I describe an algorithm that it can relate objects from different images of the same scene. This algorithm needs like starting point the mask images from N full calibrated cameras[[1]. The algorithm is able to identify different objects presents in the scene and it can correlate these objects from one picture to another. Using that information I can optimize the 3D reconstruction algorithms.

1. INTRODUCTION

The main goal of this algorithm is to optimize CPU time of Shape from silhouette (SFS) [[2] algorithm.

The starting point of SFS is the mask images from N full calibrated cameras. Probably from all objects in the mask images we are only interested in one, and it could be very useful to delete the spare information that there are in those mask images. To delete that spare information first we have to identify disjoint objects from mask images. After that we have to select a camera like reference camera, we can use different criterias to do to that, for instance a zenithal camera would be a good choice if there aren't overlapped objects. The next step of this algorithm is to select a representative point (pixel) from all identified objects in reference camera picture. Finally we have to calc the epipolar[[3] lines of representative points over all pictures (except reference camera picture). And now we have identified the objects from mask images that are related through different pictures. And with this information I can select the object that I'm interested, and I can drop from all mask images the information related with another objects, doing that I'm optimizing the CPU time of SFS algorithm.

2. ALGORITHM

The **starting** point of this algorithm are the N captured images from scene, its background images and the projection matrices and center camera points of all cameras. In the *Figure 1* we can see in the first row the synthetic background images, and in the second row there are the same images of the scene with 4 floating spheres, in this case the spheres are the foreground (the objects).

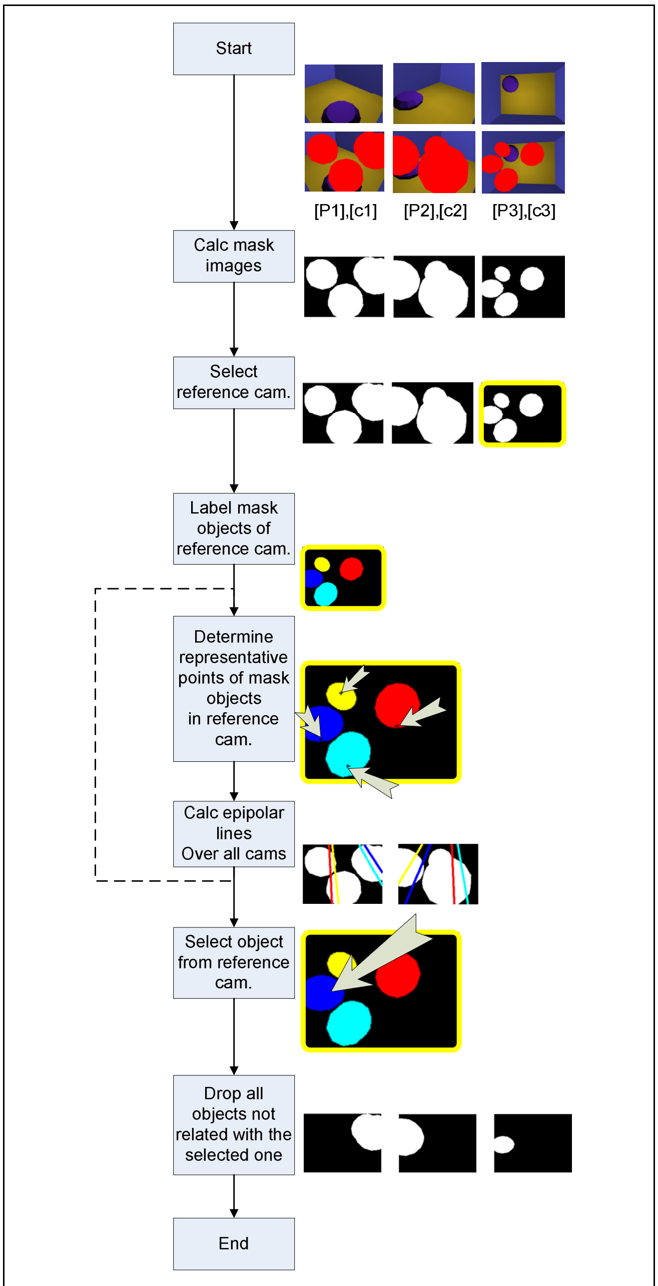


Figure 1: Object labeling algorithm

The second step of the algorithm is: to **calc mask images**. In this step we have to extract foreground from background, there are a lot of methods to do that, the simplest one is to sub the luminance background image from luminance foreground image, and the non-cero areas will be the foreground. This method is useful when images are synthetic; when the images are real we have to use complex methods.

The third step is to **select a reference camera**. This step depends on the concrete problem. In this case, a room where overlapped objects are not allowed, a good choice for a reference camera will be a zenithal camera.

The step **label mask objects of reference camera** is so simple, it only labels with the same ID (or color) the pixels of the reference mask image that belongs to the same continuous foreground zone (object).

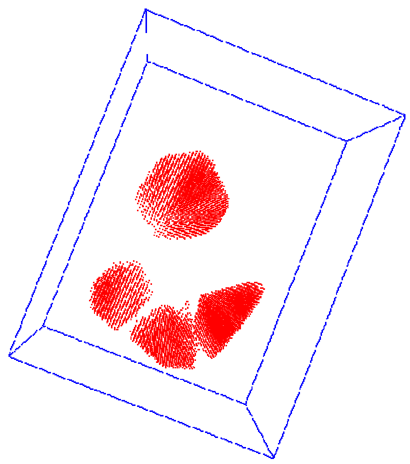
To **determine the representative points of mask objects in reference** camera image, I can use a lot of methods: Centroid, random, edge points, etc...
In this article we have used only 1 point determined by a pseudo-random method.

The next step is to **calc epipolar lines of all representative points over all camera images**. The result of this step can be used to change the representative points selected in previous step.
If the epipolar line of any representative point of RED object crosses any foreground zone (object) of any camera, this means that this object is necessary to reconstruct the 3D image of RED object.

The final step is to **select an object to reconstruct** using SFS algorithm. In our example (*Figure 1*) we have chosen the blue sphere. Finally we have to drop from all cameras all objects that there aren't intersected with epipolar lines of any representative point of blue object.
And now, for instance, we could send the resulting mask images to SFS algorithm.

3. RESULTS

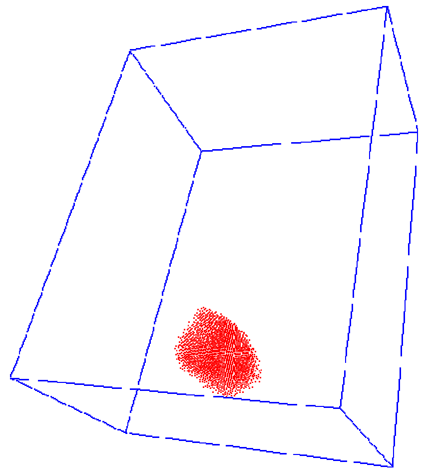
The test environment is the matlab 7.0 suite executed in a windows XP PC (P4, 2.4GHz, and 512MB).
In the *Figure 2* we can see the 3D result of SFS, the source mask images of SFS was the ones generated from a synthetic room with 4 floating spheres recorded with 5 virtual cameras.



SFS Calc time: **32.5 ±3** secs.

Figure 2: Normal result of SFS

In the *Figure 3* we see the results of the same SFS used in the previous results but in this case the mask images used as source parameters of SFS are the mask images obtained from the application of object labeling algorithm (we have suppose that we are only interested in 1 object).



SFS Calc time: **23.6±2** secs.

Object labeling calc time: **1,4±0.1** secs.

Figure 3: Result of SFS after object labeling (selected 1 object)

To calc all algorithm times, we have executed the algorithms 10 times and we have calc the mean and the standard deviation. The given results are: $\overline{X} \pm 1\sigma$

Finally we can calc the saved time in SFS thanks to use object labeling algorithm:

(*)SAVED TIME ≈ 32.5 – (23.6 + 1.4) = 7.5 secs.

(*) The saved time depends directly that how is implemented the SFS algorithm.

4. CONCLUSIONS

This algorithm is a simple, fast and it uses low CPU resources, it can be executed before SFS and it can save a lot of CPU time, in our test the saved time was 7.5 secs, over 32.5 (**23%**).
This algorithm works very well if no occlusions happen, if an object covers another in a determinate point of view (image) both masks will be integrated in only 1 mask in that image, and the algorithm can't drop all extra information.
The most critical point of this algorithm is to "determine representative points of mask objects" due to an error in that choice could do that any mask don't find its related mask in another image, if this happen we could redo these step chosen another representative point.

5. REFERENCES

[1] Josep Ramón Casas, Camera projection and calibration matrices. Power point presentation. *UPC, 2006.*

[2] A. Laurentini. The visual hull: A new tool for contour-based image understanding. Proc. Seventh Scandinavian Conference on Image Processing, *pages 993–1002, 1991.*

[3] Jordi Salvador Marcos, "Anàlisi de l'espai 3D des de múltiples càmeres adaptada al mostreig regular de les imatges", *PFC, UPC, 2006.*