Mesh Reconstruction from Range and RGB Data

Michel Amat, Miles Hansard & Radu Horaud
michel.amat@inrialpes.fr / miles.hansard@inrialpes.fr / radu.horaud@inrialpes.fr
INRIA Rhône-Alpes
655 Avenue de l'Europe, Montbonnot, France 38330.

Last modified December 3, 2010

1 Introduction

This paper gives a method to reconstruct a textured mesh of a scene seen from a set of two color cameras (RGB cameras) and one range camera (Time Of Flight camera).
The TOF camera has a very low resolution (40,000 pixels) and is used to get the general 3D shape of the scene, while the RGB cameras have a high resolution (2,000,000 pixels) and provide rich texture information.
The figure below shows the used configuration of this set of cameras to capture the scene.

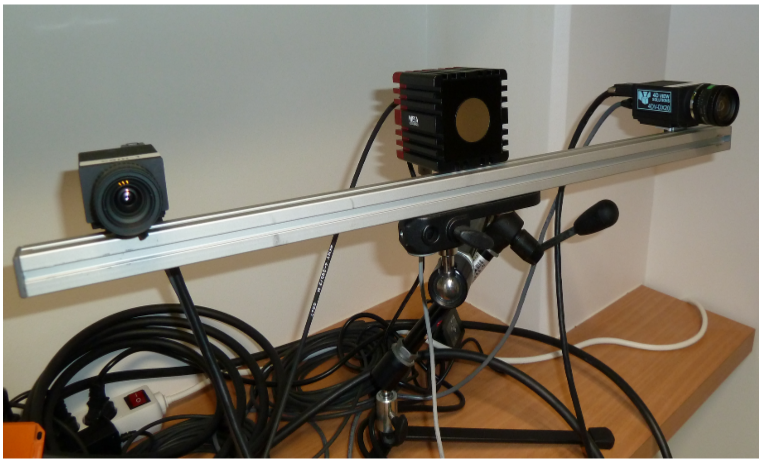

Figure 1: mesh reconstruction

We assume that the camera system has been calibrated and that we can directly use the undistorted images.
We also assume that all the calibration information (camera intrinsic and extrinsic matrices) are known, and that have the projective transformation from the 3D world points to the 2D images.

2 Mesh construction

The 3D points of the scene are directly taken from the undistorted range buffer from the TOF camera. Those points are the vertices of the initial mesh.
Then, the order of those vertices in the range buffer is used to obtain the edges of this mesh.
Thus, the figure below shows how the vertices of the mesh (which are the pixels of the TOF camera) are linked.

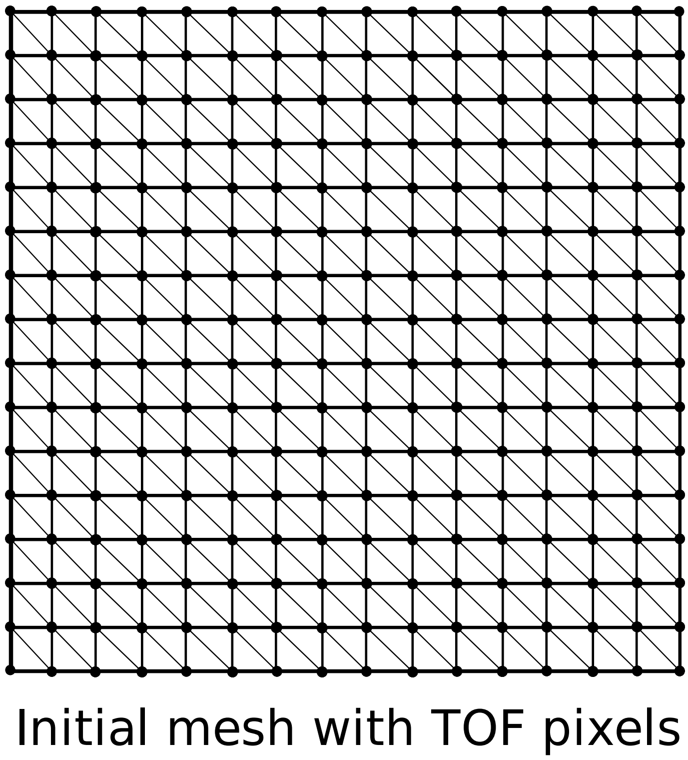

Figure 2: mesh reconstruction

The projective transformation from the 3D world points to the 2D images being known, we can easily find the RGB information of each vertex of the mesh, and also, apply the texture information of the 2D images at each triangle of the mesh.

But, for each point of the scene which can be reprojected in both of the tow RGB images from the color cameras, we have two different textures.

Figure 3: TOF reconstruction

3 Filtering

This initial mesh contains some spacial incoherences at the border of different grounds (between foreground and background object).
Indeed, objects at different levels are all the same linked at their border into the mesh by long and thin triangles.
To get rid of those wrong triangles, the method used was to filter the triangles for which the ratio between their area by their perimeter was too little. More precisely, we removed the triangles for which $\frac{\sqrt{area}}{perimeter} < thresh$ where 'thresh' was a threshold to fix with care.
With a too big threshold, some holes appear in the mesh (some good triangles have been removed), with a too low threshold, some of the triangles linking foreground and background objects remain.

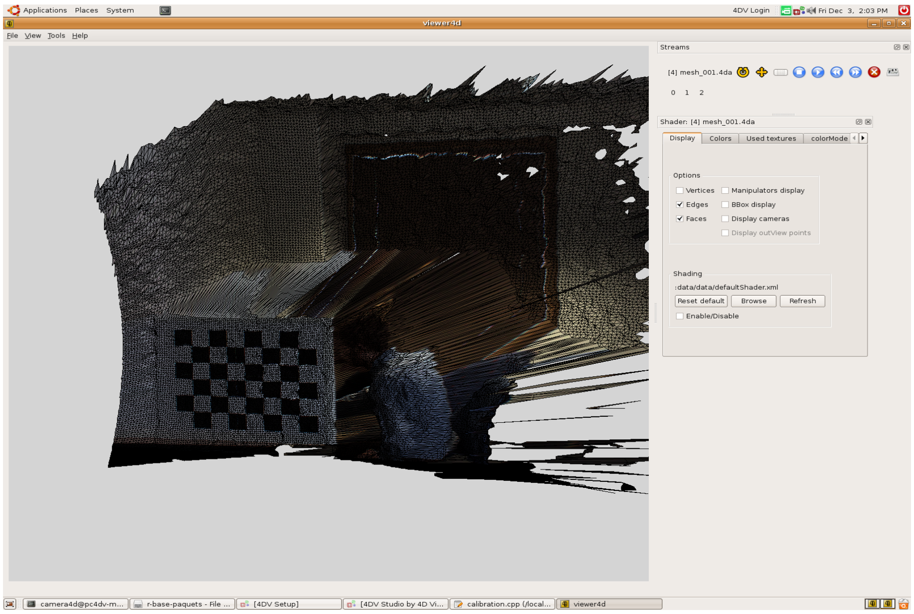

Figure 4: without filtering

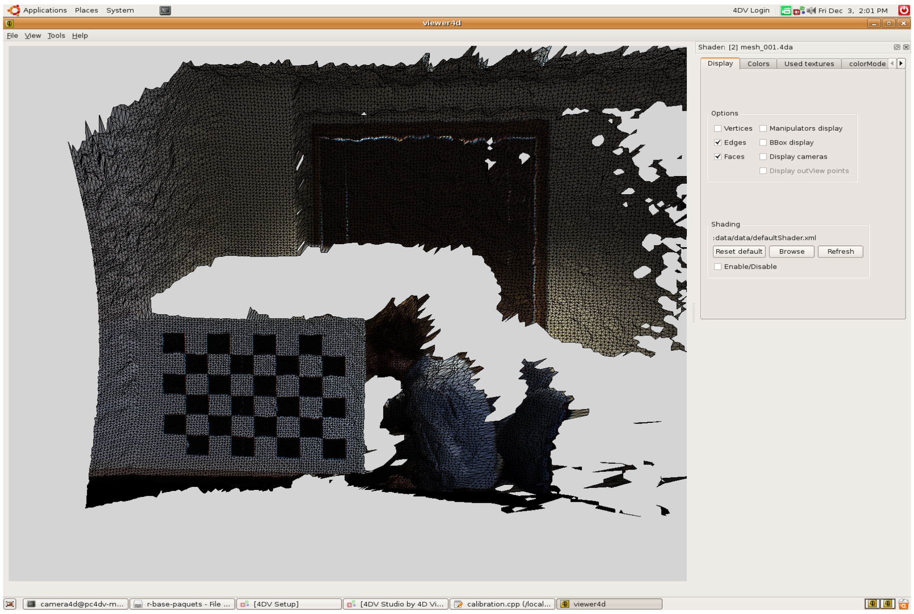

Figure 5: right threshold

4 Refinements

To improve the mesh, in particular at the borders of objects, at the mesh construction, in each pair of triangles, we flip the way of triangles if it let obtain thicker triangles.

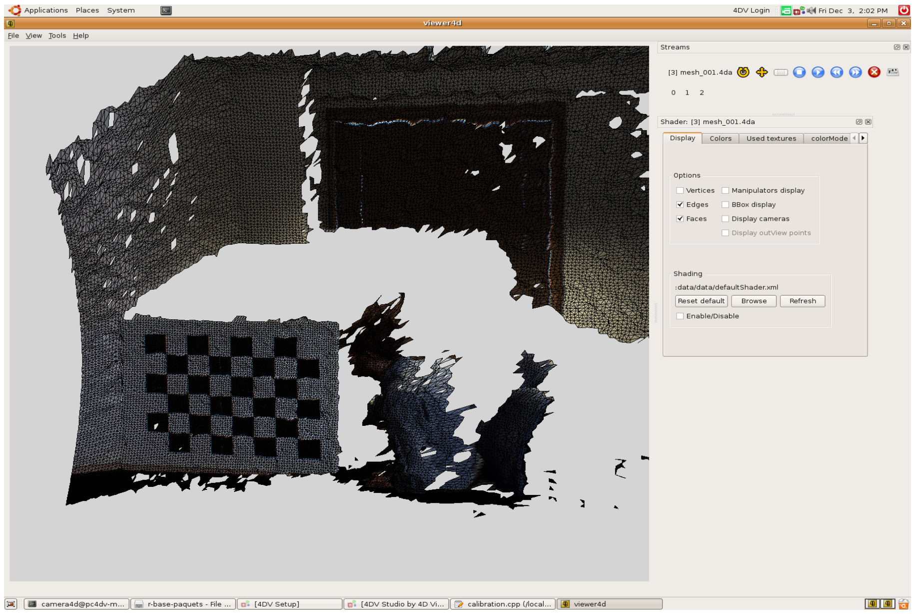

Figure 6: wrong threshold

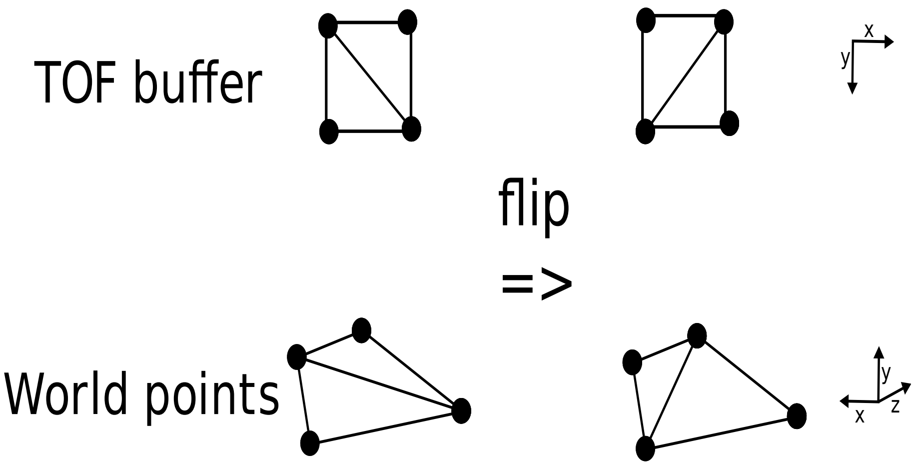

Figure 7: flip

5 Occlusions

Some part of the mesh will be 'seen' by both of the RGB cameras, or only one of them, or none of them.
If it is seen by both of them, the average of the two textures can be used to color the mesh. If it is seen by only one of them, we use only its texture to color this part of the mesh. And at last, if none of the camera can see a certain part of the mesh, this part will just remain black.

To compute which camera can see which part of the mesh, shadow mapping technic is used, by replacing the light in the shadow mapping method by each of the RGB cameras.

The remnant effect is due to the low resolution of the TOF camera (and also to its

Figure 8: TOF reconstruction with flip

error, at the hair more particularly).

Figure 9: flip improvement

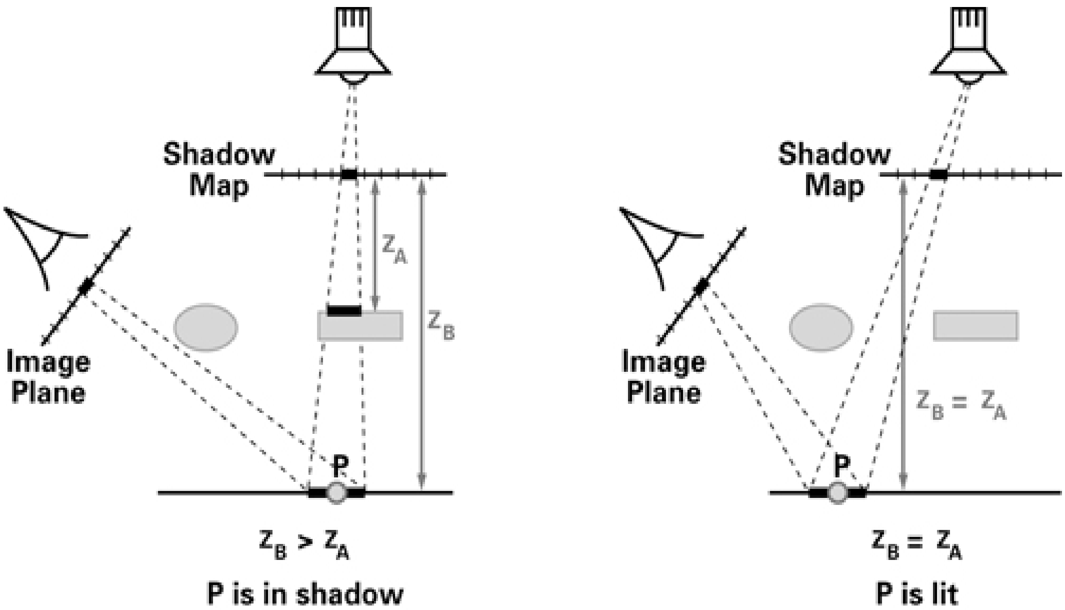

Figure 10: shadow mapping

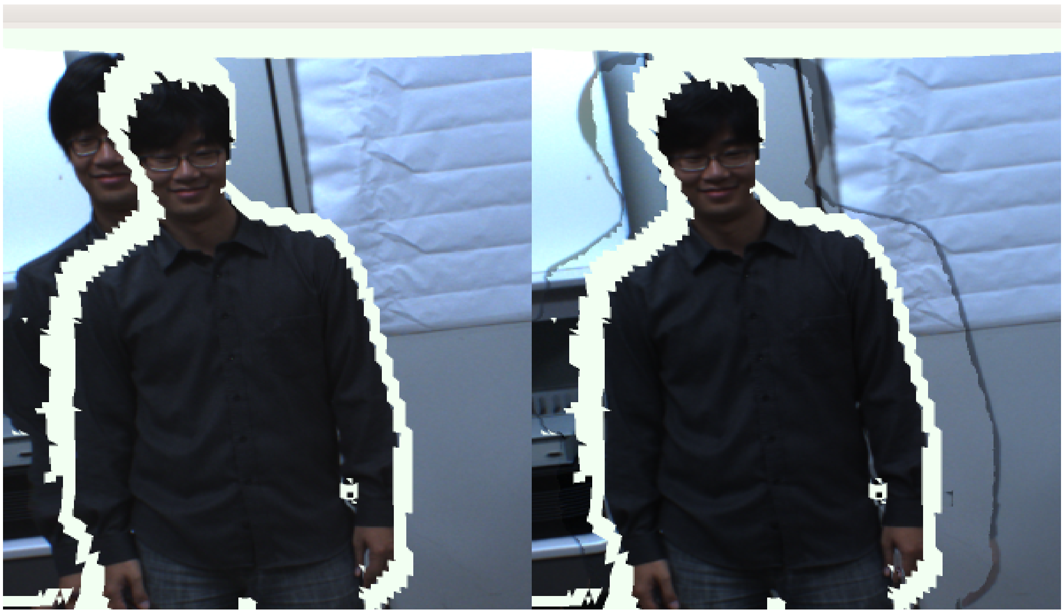

Figure 11: texture with shader