Laplacian smoothed depth proxy for rendering

Neil Birkbeck

March 19, 2010

In the past colleagues in my lab have used either a view-dependent triangulation or a view-independent tetrahedralization as a geometric proxy for rendering. Recently, we were adapting the view-independent tetrahedralization to work in a predictive display setting on a real robot.

There was the occasional problem with the geometry not occupying all of screen space, meaning that portions of the screen would not be rendered with a photo-realistic prediction. A quick thought on how to overcome this was to use a laplacian smoothing of the 3D points in a view-dependent manner. The proxy would work if the scene only contained a few sparse samples (as the smoothing would ensure that the surrounding depths would be filled in), and it should also work if there were many depth samples.

Given a set of 3D points $\{\mathbf{X}_i\}$ and a camera projection matrix $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$, the goal is to generate a scene-dependent triangulation that respects the projection of the 3D points, while covering the entire screen. First the image is partitioned into a number of rectangular bins. Let each of these bins have a depth estimate z_k. An averaging of the depths of the projected points is one way to get a depth estimate:

$$\hat{z}_k = \frac{\sum_{i \in bin_k} [\mathbf{P}]_3 \mathbf{X}_i}{|bin_k|}$$

Where i is in bin_k when $\Pi(\mathbf{P}\mathbf{X}_i)$ projects into the bin. Of course, there will be some bins that are empty (and hence have no depth estimate).

The empty bins get their value from interpolation. Solving the following system of equations gives a smooth estimate of the depth that roughly obeys the depth estimate:

$$\lambda \nabla^2 z = 0 \tag{1}$$

$$z_k = \hat{z}_k \qquad \{|bin_k| > 0 \tag{2}$$

λ is a parameter controlling the amount of smoothing. The idea is illustrated in Figure 1

Figure 2 shows the smooth proxy in use as well as a tetrahedral proxy.

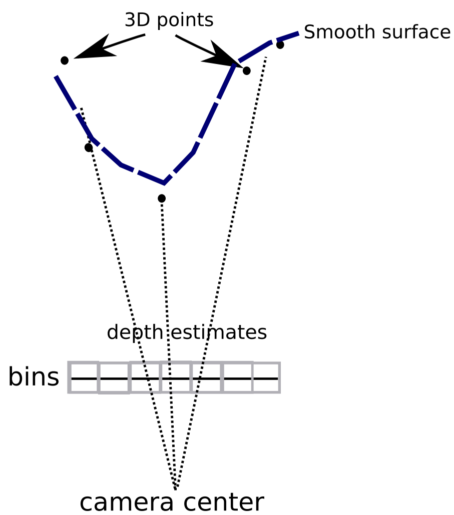

Figure 1: A few 3D points are projected into the image bins. The bins with an observation are used as constraints in the smooth depth. The result is the dashed blue curve.

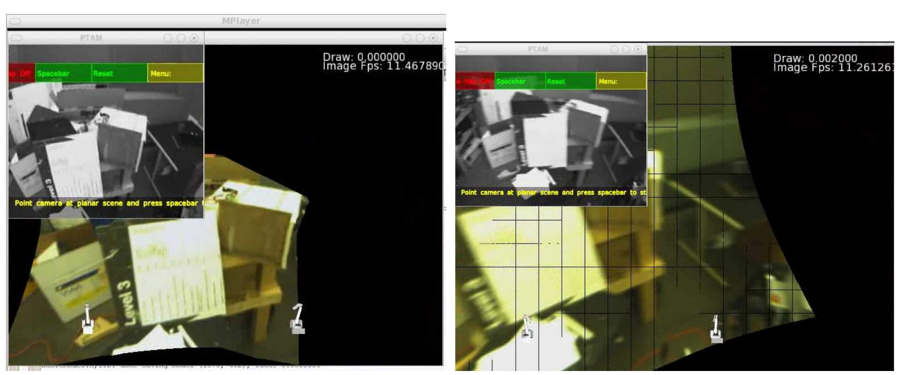

Figure 2: During simulation the user display (background) shows a predicted version of where the operator would like to be. The foreground window shows the current camera view (it is mounted on a robot). Left: tetrahedral surface. Right: smooth surface. The wireframe overlay indicates the resolution of the bins. The smooth surface will always show the latest image.