Computer Vision
COMP-290 Spring 2000
Voicu Popescu

3D from Michael Naimark's photographs

Project Report

1 Problem Presentation

Inferring 3D from photographs using depth from stereo has been the goal of much research, and justly so:

- images enhanced with per-pixel depth allow the creation of new views of the 3D scene (Image-Based Rendering);
- it bypasses the increasing difficulty of modeling conventionally, with polygons; as polygon-renderers became powerful enough to render millions of triangles every second, even every frame now, producing the triangles has become a serious bottleneck;
- the system for depth from stereo is fairly simple and inexpensive, which cannot be said about laser rangefinding for example, or other techniques that require precise control over the lighting;
- it is less, or not at all, invasive; the system can potentially function wherever you can snap photographs;

Michael Naimark, a researcher from Interval (research lab that is disappearing as we speak), visited numerous interesting, and sometimes exotic, places. He filmed them using pair of 35mm movie cameras arranged in the classic stereo setup (figure 1). Panoramas were acquired by rotating the cameras around the left camera's center of projection.

The lenses are of high quality (Zeiss), care was taken to synchronize the shutters and they filmed some calibration patters so we decided to try to use them for depth extraction. The first step was the digitization. The film was quite slow, of very fine granularity so a California lab sent us a tape with 4kx3k digitizations of a few of the images, including some images of the calibration grid (figure 2).

Figure 2. The left (top) and right images of a stereo pair. Dawn in Timbuktu.

2

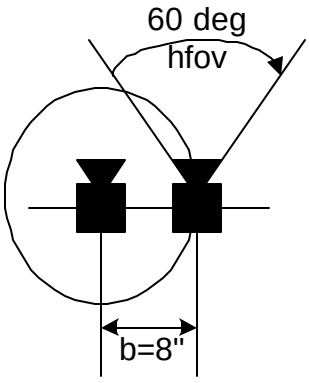

Figure 1. Naimark's stereo setup

Camera Calibration

We had a rough guess: the cameras were supposed to be 8 inches apart and 60 degrees horizontal field of view each. Since the film was 35mm and the image aspect ratio 4:3, we had an approximate calibration solution. In order to determine other intrinsic parameters like radial distortion, image coordinates of the projection of the center of the lens and pixel aspect ratio, we need to establish correspondences between 3D points and image plane coordinates. One needs as many as possible (to cover the entire viewing volume) and as precise as possible. Figure 3 shows an image of the grid pattern we tried to use for calibration. The grid-square corners were detected (figure 3 right) but when it came to associating the corresponding 3D coordinates we encountered the difficult problem of not knowing the geometry (dimensions) of the grid.

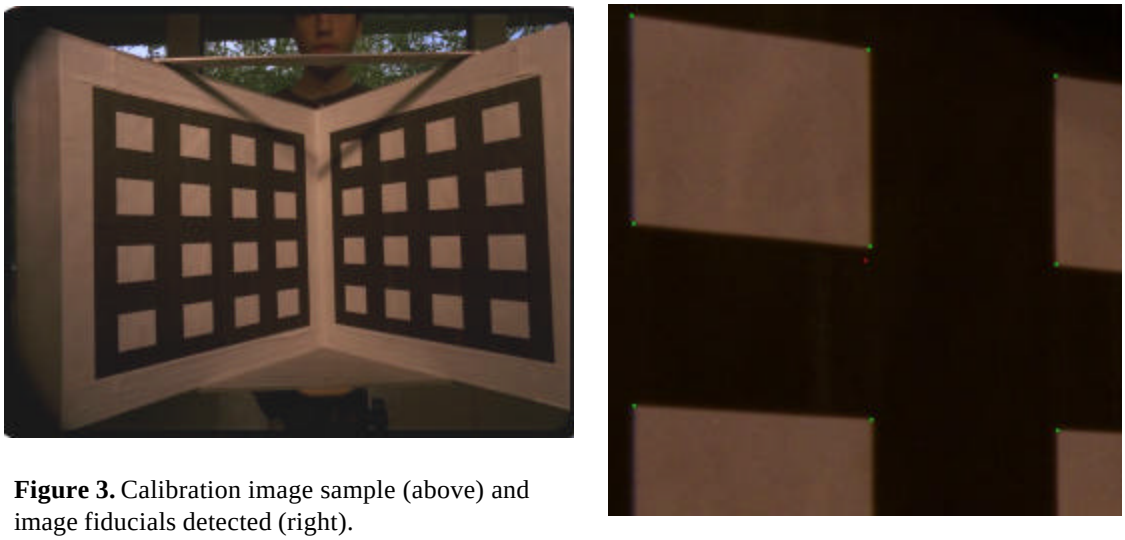

Figure 3. Calibration image sample (above) and image fiducials detected (right).

The images were acquired in 1996 (last century) and the calibration grid doesn't exist anymore. Paul (the person behind the grid found in an old email some numbers that looked like being the grid measurements, but unfortunately they did not match (see figure 4).

At this point it became clear that calibration will not be solved as precisely as one would like. The last attempt was to guess the 3D measurements of the grid. Appendix 1 lists the results of the calibration. The errors are large (20 pixels on average) so we did not trust the intrinsic parameters found. We decided to go ahead and use the rough guess presented above.

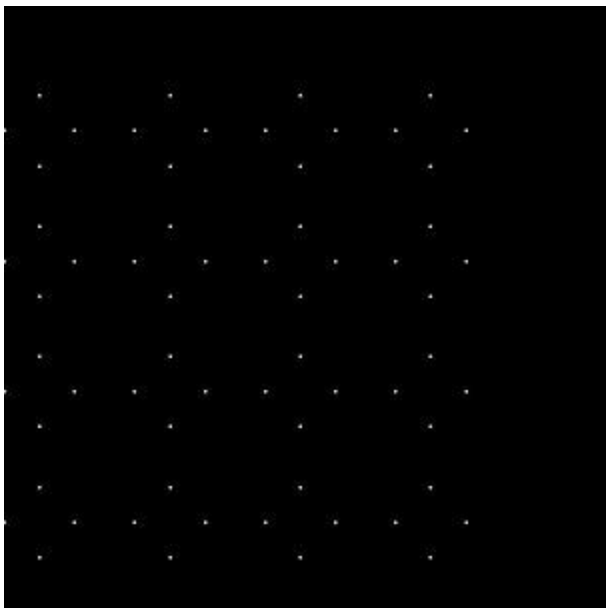

Figure 4. View of the points with the same 3rd coordinate.

One more difficulty made us decide to abandon Naimarks's images: the left and right images do not match, that is they were taken at different moments in time. Somehow, along the path from Interval-UNC-Lab-UNC, the images ended up not matching each other. This can be noticed in figure 2 by looking at the dromedaries' legs. Since the camera moved itself, not even static scenes were an option. Of course it is not impossible to extract depth even when the cameras are at arbitrary positions with respect to each other, but this implies finding the extrinsics of the cameras first.

3 Correspondences

We proceeded with images generated synthetically from a geometric model using a polygon renderer. Figure 5 shows the *Kamov* and *Eurotown* models.

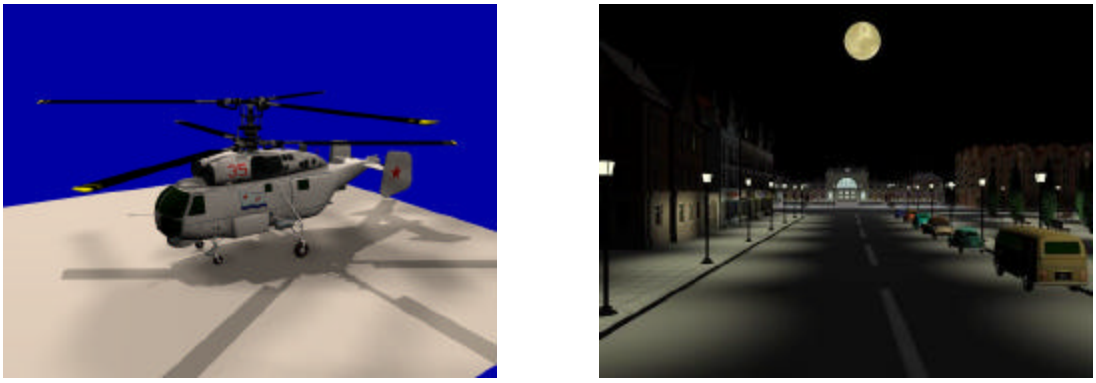

Figure 5. Kamov helicopter and Eurotown.

What does using synthetic image imply?

- perfect camera calibration;
- no camera errors (possibly not modeled and solved for in calibration) like higher order distortion parameters, CCD noise, aperture effects etc.
- no errors in the stereo setup, no errors in shutter synchronization;
- elimination of all other errors we didn't even know we had in the cameras / camera setup.
- decreased scene complexity: the models are quite complex and the state-of-the-art rendering techniques [1]are employed in order to produce images that are as close as possible to photographs, but the images might still be simpler than real world photos; we are not sure what this implied for our results: it seems like finding correspondences from synthetic images is harder at first since there are more featureless surfaces but then edges and other high-frequency features are very regular and more pattern-matching friendly. We will not find out the difference between synthetic and real images until we try it.

Under normal circumstances, the real problems in depth from stereo begin only now. Finding correspondences is a difficult problem because:

- *repetitive patterns*: the pattern search can get confused by other instances of a similar object;
- *featureless surfaces:* pattern matches fairly at a continuous set of locations;
- *motion parallax*: the left camera sees surfaces the right camera does not and vice versa;
- *view dependencies*: the same surface seen through the left camera appears different when seen through the right camera;
- in general, it requires understanding of the scene, which is what we want to gather to begin with
- *computationally intensive*: the cost of pattern matching for one reference position is proportional to the area of the pattern, and the length of the epipolar segment.

We implemented a simple pattern matching routine that found the minimum difference between the left-image's 10x10 pattern and the right image. Unique, per-image, minimum and maximum depths were used that defined 700 pixel long epipolar segments. The maximum depth used was infinity, which translates in having to search from the right-image pixel with the same coordinates. Since we knew that the stereo setup was perfect, we searched only on a one-pixel wide horizontal epipolar segment.

[1] We did not have available an unlimited rendering time; thus techniques like Monte-Carlo ray-tracing that come closest to photorealism were not employed.

4 Results

Figure 6 shows the results we obtained by rendering the resulting depth image from another viewpoint.

Figure 6. Resulting depth-images were warped to different views to illustrate depth. Only samples with good correspondences were rendered.

Good correspondences were established for numerous samples. In the *Eurotown* case the view change is illustrated by the "shadows" of the left light-poles on the sidewalk.

4.1 *Future work*

Obviously there are many problems that need to be solved before the warped images are of quality similar to the input images. Before we analyze the problems of the current implementation let us mention that scene acquisition for re-rendering is maybe one of the most challenging tasks for computer vision. Scene recognition for example can be readily done on the results above.

Currently the biggest challenge for finding correspondences are the featureless surfaces. We believe that solving the correspondence problem at lower resolutions and use that as initial guesses for the higher resolution images might be the way to go. If a strong correspondence is established at a higher resolution than it should replace the previous-level value. This would ensure nice, aliasing free results for high-frequency areas while the low frequency areas might have satisfactory solutions from the lower resolution stages. Conversely one could experiment with changing the size of the pattern in the high-resolution images but that method, probably of similar results, is much more expensive.

Another problem is the "doubling of silhouettes", easy to see at the blades of the helicopter propellers. It is somewhat related to the first problem since the sky is featureless but it seems much harder to eliminate.

Similar objects are frequent in man-made environments. For example the distant building (train station) in the *Eurotown* case has numerous identical windows. A third image, displaced vertically would help greatly at disambiguating the correspondence search. Also the vertical and horizontal directions are much more frequent than other directions and one should not use baselines that are aligned with them.

4.2 *Implementation details and performance*

The 300MHz PC C++ implementation needs about 70s to find the correspondences of an entire row (4k pixels) in the left image, thus about 60hrs for an entire 4kx3k image. Optimized code and latest PC would probably reduce the figure 4-fold. The parallel implementation for an SGI 32 R12000 processor-parallel computer needs only 4 hours to fully process an image.

An obvious way of improving performance is to reduce the length of the epipolar segments. One could find larger minimum depths for sub-regions of the image (manually) or could use the lower resolution guesses.

Also the computation is so simple that one could investigate building custom hardware. No data dependencies and simple operations suggest a SIMD architecture.

5 Conclusions

We conclude that depth from stereo is possible. High-resolution images introduce complexity (typical file size is 50MB, implies working with unreliable, unfriendly tape-readers, slow running times). A lesson I learned is not to agree to calibrate a camera if the camera is not available to you.

Also we realized late in the game that even if we computed perfect depth from Naimark's images, it would not be usable since as soon as the viewer will change the view-point significantly, unsampled surfaces will appear, destroying the immersion illusion. One really needs several sampling locations to ensure all the samples needed for a certain viewing volume.

6 Appendix

Results of calibration using guessed grid meaurements. The translations and the object space errors are in grid square widths (although listed in mm).

```
Non-coplanar calibration (full optimization)

camera type: Naimark 4096x3112 digitized images

data file: Cal_L.0001.dat  (128 points)

       f = 37.407382  [mm]  fovs: h = 62.738943, v = 44.420582  [deg]

       COP = (15.711861, 3.943370, 15.228418)

       kappa1 = -1.332402e-04  [1/mm^2]
```

```
        Tx = -0.228343,  Ty = -3.096169,  Tz = 22.015428  [mm]

        Rx = -176.826060,  Ry = 46.192055,  Rz = 180.476443  [deg]

        R
        -0.692219   0.031653   0.720993
        -0.005756   0.998764  -0.049374
        -0.721664  -0.038328  -0.691181

        sx = 1.089319
        Cx = 2280.621453,  Cy = 1527.350805  [pixels]

        Tz / f = 0.588532

        distorted image plane error:
          mean = 20.475701,  stddev = 10.402180,  max = 41.088342
[pix],  sse = 67406.632344  [pix^2]

        undistorted image plane error:
          mean = 20.143504,  stddev = 10.383350,  max = 40.876049
[pix],  sse = 65629.750992  [pix^2]

        object space error:
          mean = 0.097010,  stddev = 0.055484,  max = 0.220716  [mm],
sse = 1.595567  [mm^2]

        normalized calibration error:  48.668439
```