Chapter 4

Direct Estimation of Motion and Structure - Experiments

4.1 Overview

In this section we perform experiments which test various aspects of direct estimation of motion and structure. Experiments 1,2 and 3 deal with the pure translation case. In experiment 1, we test the accuracy of heading estimation for non-collinear motion. Experiment 2 tests the accuracy heading estimates obtained from collinear motion. Experiment 3 tests a standard optical flow program on some of the same input images and compares the results with the 'direct methods'.

We then move on to case where the motion involves both translation and rotation. Experiment 4 revisits the image sequence used in Stein & Shashua [74]. We show that the improvements described in Section 3.7.5 greatly improve the rotation estimates. In Experiment 5 we repeat the heading estimation experiments but this time the motion includes some rotation. We also show how not taking into account even small rotations ($\sim 0.5^o$) leads to large errors in heading estimates. In Experiment 6 we test the accuracy of Euclidean reconstruction using the simple image of a cube. We measure whether the right-angles between the cube faces are correctly recovered.

In all the preceding experiments the camera was mounted on a motion stage so that ground truth motion could be known accurately. In Experiment 7 we repeat the Euclidean reconstruction experiments but this time with a hand-held camera. We show that if we neglect the rotation the shape recovery is noisy. We also show that if we neglect radial lens distortion then the reconstruction is qualitatively good but the Euclidean measurements, such as angles, are less accurate. This agrees with the observations regarding lens distortion and the trilinear tensor in Chapter 7.

4.2 Pure Translation

4.2.1 Experiment 1: Accuracy of Heading Estimates for Pure Translation

In this experiment we measure the accuracy of the estimation of the camera motion direction (heading direction) in the case of pure translation. We also gauge the effect of the non-linear lens distortion.

Experimental Procedure:

The camera was mounted on a motion stage with three degrees of computer controlled motion (Figure 4-1): horizontal translation, vertical translation and rotation around the camera's Y axis. The axes of motion are highlighted in Figure 4-1a.

The camera Y axis was aligned with vertical axis of the stage. Initially the camera was positioned so that the optical axis was aligned with the horizontal axis of the translation stage. The accuracy of the alignment was $\pm 1^o$. The camera was rotated α^o and an image captured. Then the camera was translated vertically $12.5mm$ and an image captured. Finally the camera was translated horizontally $10mm$ and a third image captured. Thus we captured an image triplet where the camera motion was pure translation with one of the motions vertical and the second at a heading of α^o. This was repeated for angles $\alpha = 0^o, 10^o,\ \ldots\ , 90^o$.

Figures (4-2a,b,c) show the image triplet for a heading angle of 40^o. Motion and depth estimates were performed with and without lens distortion correction. The depth map estimate is shown in figure (4-2d). The camera lens was (4.9mm lens / 82^o FOV). The experiment was then repeated with a medium-wide angle lens ($8.5mm$ / 52^o FOV) for the scene shown in figure (4-15).

Results:

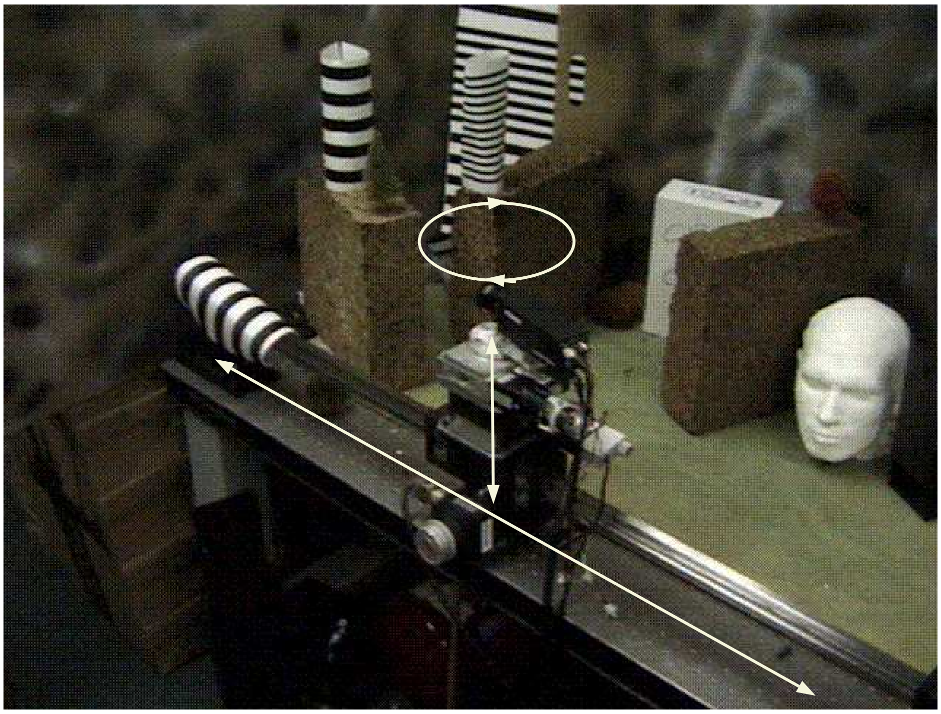
Figure 4-3 and Figure 4-4 show the heading estimates for the wide and medium-wide angle lens respectively. The heading estimates are within a few degrees. Notice that when the true heading was inside the FOV the errors were less than 1^o. When the true heading was outside the FOV the errors increased to $1 - 2^o$. In the case of the wide angle lens the errors were considerably larger when lens distortion was not taken into account (dashed line in Figure 4-3b).

4.2.2 Experiment 2: Collinear Motion with Pure Translation

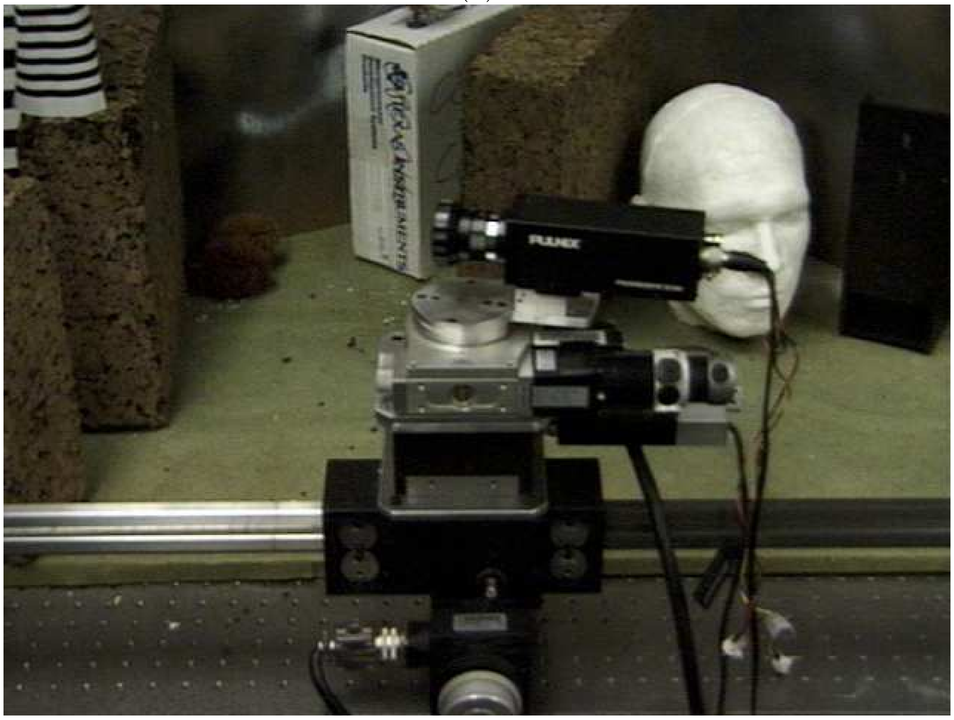
This experiment aims to test the accuracy of motion heading estimation using collinear motion as opposed to non-collinear motion. We also qualitatively compare how the two types of motion affect the structure estimates and convergence of the algorithm. The algorithm is tested against a scene which is challenging for optical flow based techniques because it lacks corner-like feature points.

Experimental Procedure:

The camera was mounted on the motion stage as described in Experiment 1 (Section 4.2.1). The vertical axis of the camera coordinate system was aligned with the vertical axis of the motion stage. The camera was rotated to a given camera heading and was then moved with three translational motions as shown in Figure 4-5. The first motion was vertical, the other two were equal horizontal motions in opposite directions. By choosing the first and second motions we have non-collinear motion.



(a)



(b)

Figure 4-1: *Two views of the optical bench and the camera mounted on the motion stage. The three axes of motion are highlighted in (a). The objects on the table are those that appear in Figure 4-6.*

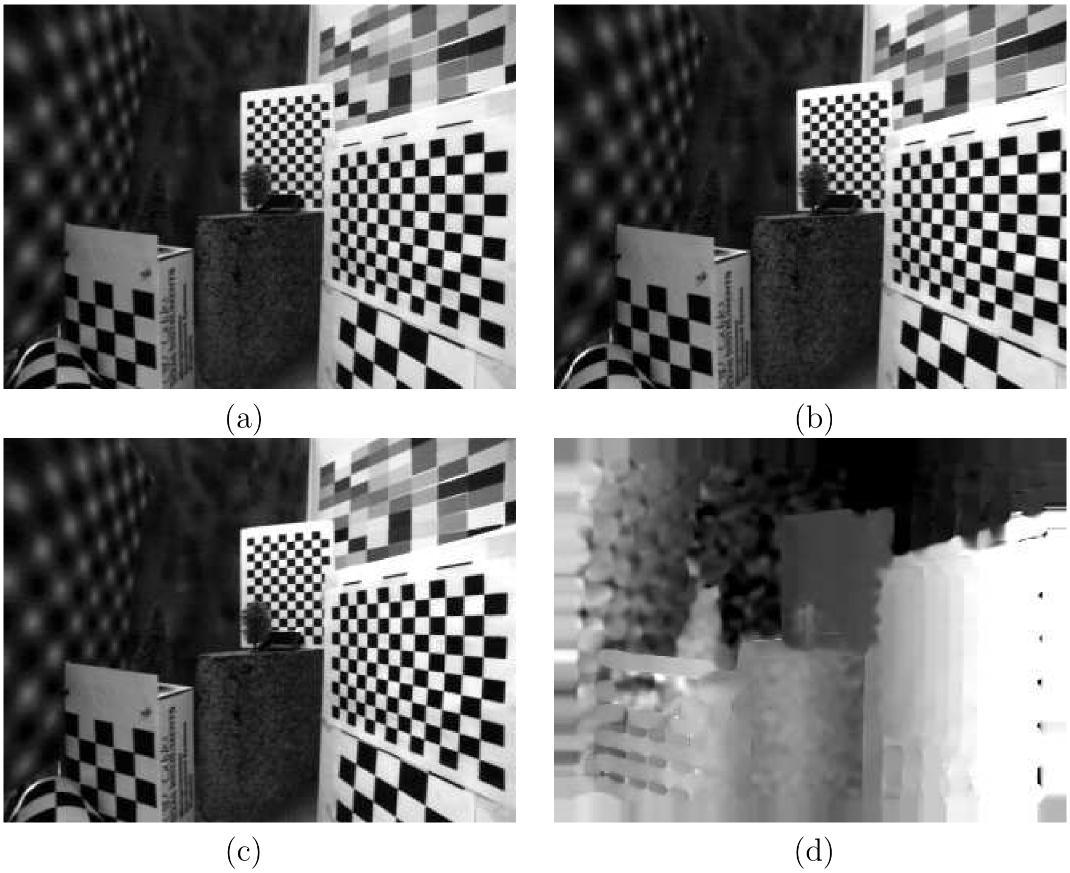
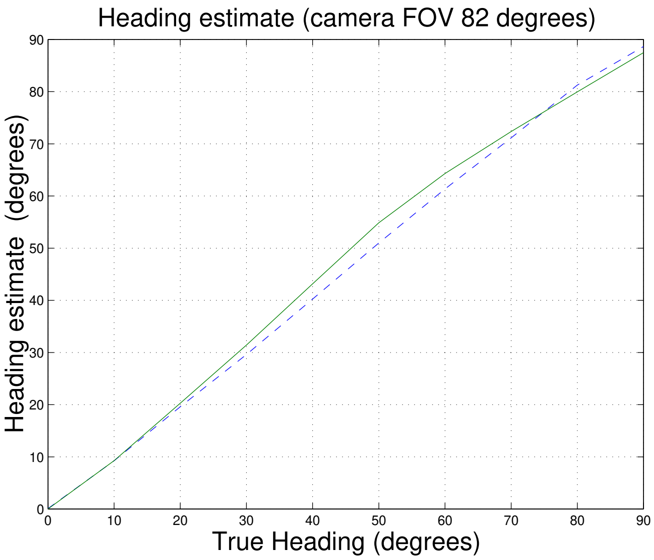
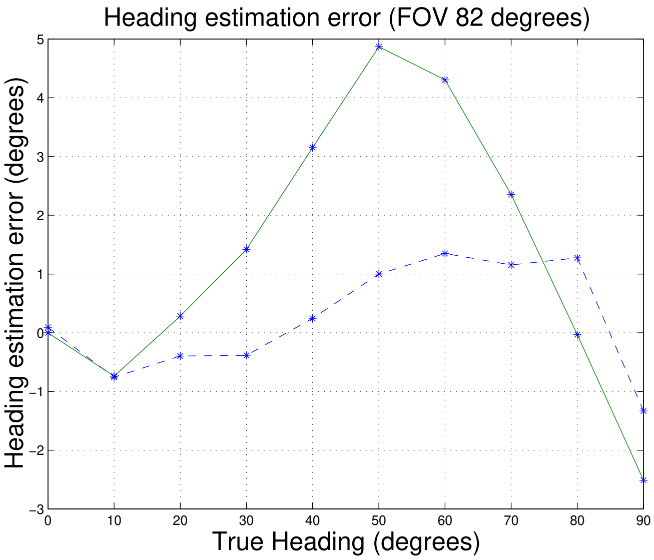


Figure 4-2: *The three input images (a,b,c) and the estimated depth map (d). The motion between images (a) and (b) is horizontal, forward and* 40^o *to the left. The motion between images (a) and (c) is vertical.* (4.9*mm lens /* 82^o *FOV)*



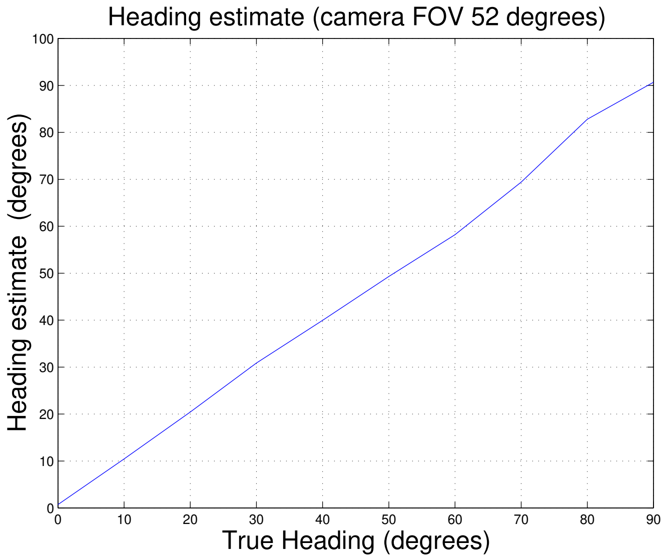
(a)



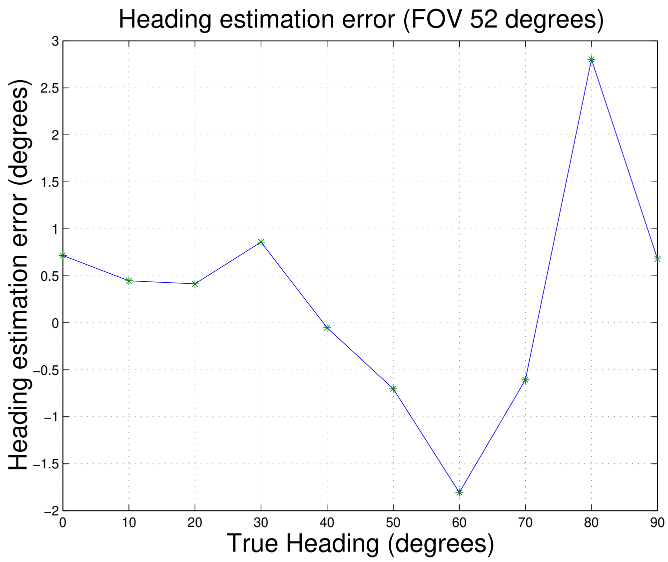
(b)

Legend:	
——	No radial distortion correction
- - -	Images corrected for radial distortion

Figure 4-3: *(a) Heading estimate from a translating camera. Pure translation was assumed. The* 4.9_{mm} *lens gives a* 82^o *FOV along the X axis. (b) Difference between heading estimate and motion stage reading ("True Heading").*



(a)



(b)

Figure 4-4: *(a) Heading estimate from a translating camera. Pure translation was assumed. The* 8.5_{mm} *lens gives a* 52^o *FOV along the X axis. (b) Difference between heading estimate and motion stage reading ("True Heading").*

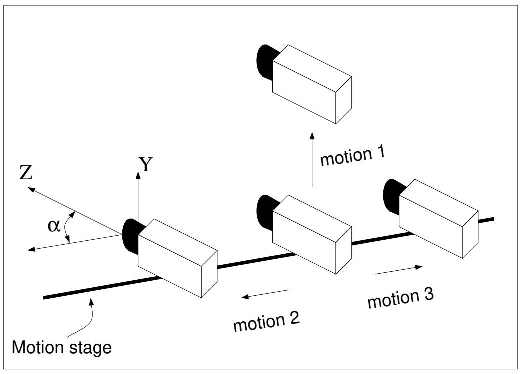


Figure 4-5: *Schematic diagram of the pure translation experiment. Camera heading is set by the angle α. Motion 2 and Motion 3 are collinear.*

By choosing the second and third motion we have collinear motion. A typical set of four images is shown in Figure 4-6 for a heading direction of 30^o.

Results:

Figure 4-7 shows the motion direction estimates using the non-collinear image triplets or the collinear triplets. In both cases the heading estimate is within a few degrees of the true stage motion.

Figure 4-6e and Figure 4-6f show the depth maps estimates for the non-collinear and collinear cases. The results are shown for the horizontal motion forward and to the left at an angle of 30^o to the camera optical axis. The FOE in this case was inside the image. As expected, at in the case of collinear motion the depth map around the epipole is not accurate since the image motion is very small. There are also errors in the depth estimates on the cylinders where the tangents to the iso-brightness contours pass through the epipole. These errors do not appear when the two motions are non-collinear because each motion can give depth estimates where the other motion fails.

The algorithm is very robust for non-collinear motion and will converge for large and small motions (for the whole tested range of $5mm$ to $40mm$) and a variety of parameter settings. For the collinear motion the algorithm would only converge for a smaller motion range ($5mm$ to $10mm$).

Figure 4-8 shows another example of 3D reconstruction using collinear motion. In this example the motion was vertical. There are no areas in the image where the edges are predominantly vertical so the depth estimates are quite good throughout the image.

4.2.3 Experiment 3: Comparison with Optical Flow Techniques.

After computing the motion and depth from an image triplet we can compute the motion flow using the motion equations (equation 3.14). In this experiment we compare

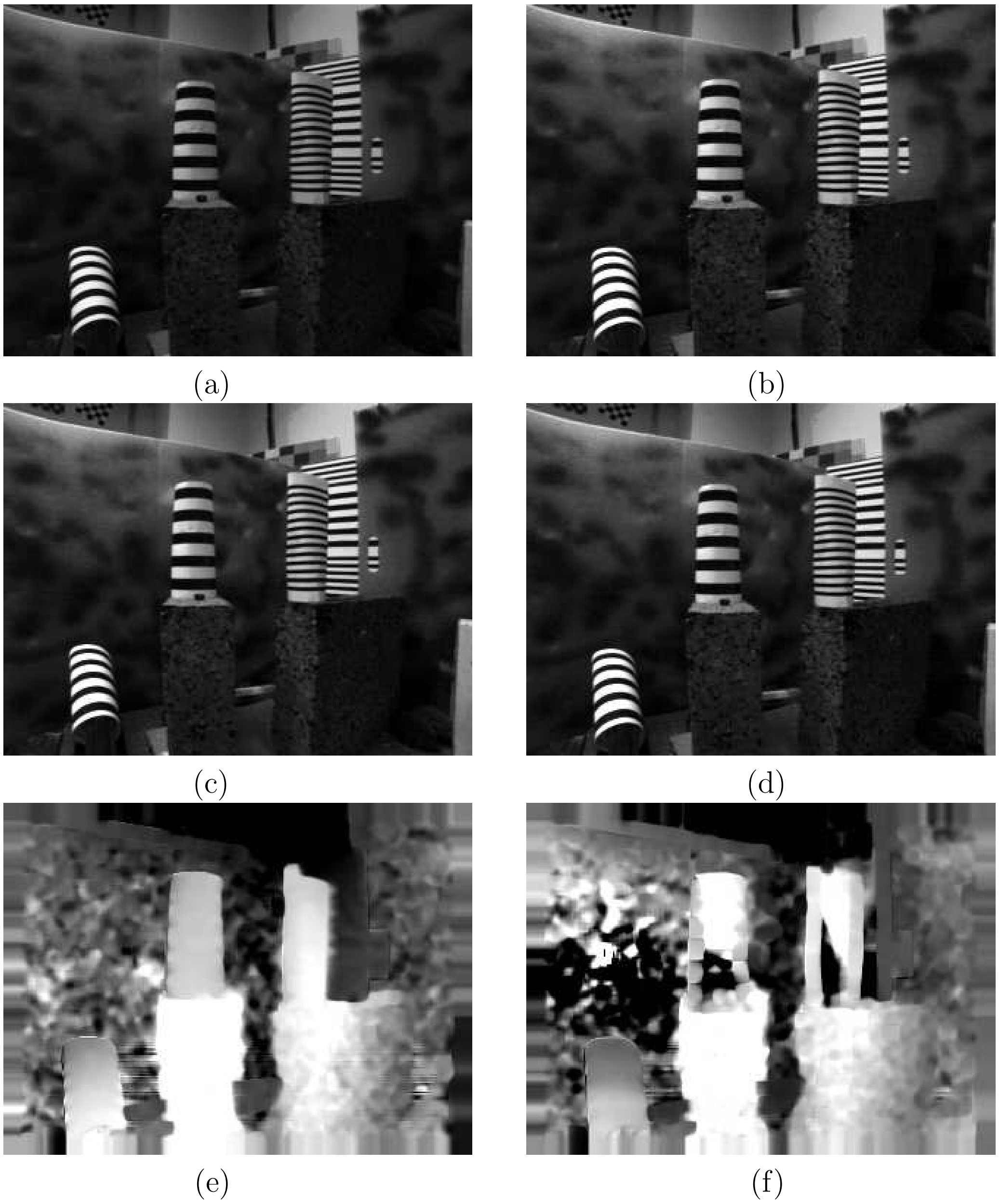
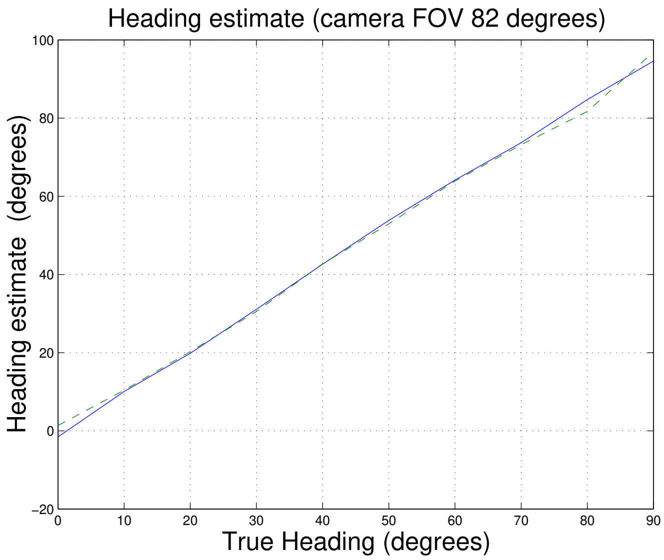
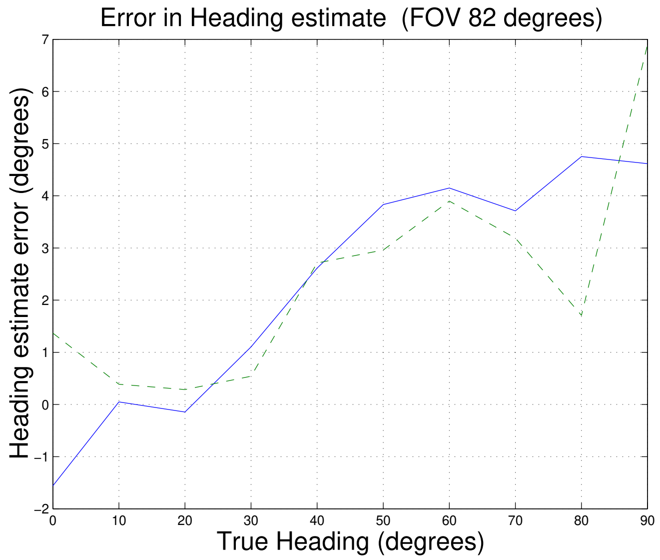


Figure 4-6: *(a) (b) (d) (c) Four input images (4.9mm lens, 82^o FOV). Motion between (a) and (d) is vertical, Motion between (a) and (b) is horizontal, forward and 30^o to the left. Motion between (a) and (c) is collinear with the motion $a \rightarrow b$ in the opposite direction. (e) Depth map recovered from images (a) (b) (d). (f) Depth map recovered from collinear motion, images (a) (b) (c). Note the depth map has errors (dark areas) around the epipole and also on the lines on the cylinders which pass through the epipole.*



(a)



(b)

Legend:	
——	Non-collinear motion
- - -	Collinear motion

Figure 4-7: *(a) Heading estimate from a translating camera - collinear and non-collinear image triplets were used. Pure translation was assumed. (*4.9_{mm} *lens / 82º FOV). (b) Difference between heading estimate and motion stage reading ("True Heading").*

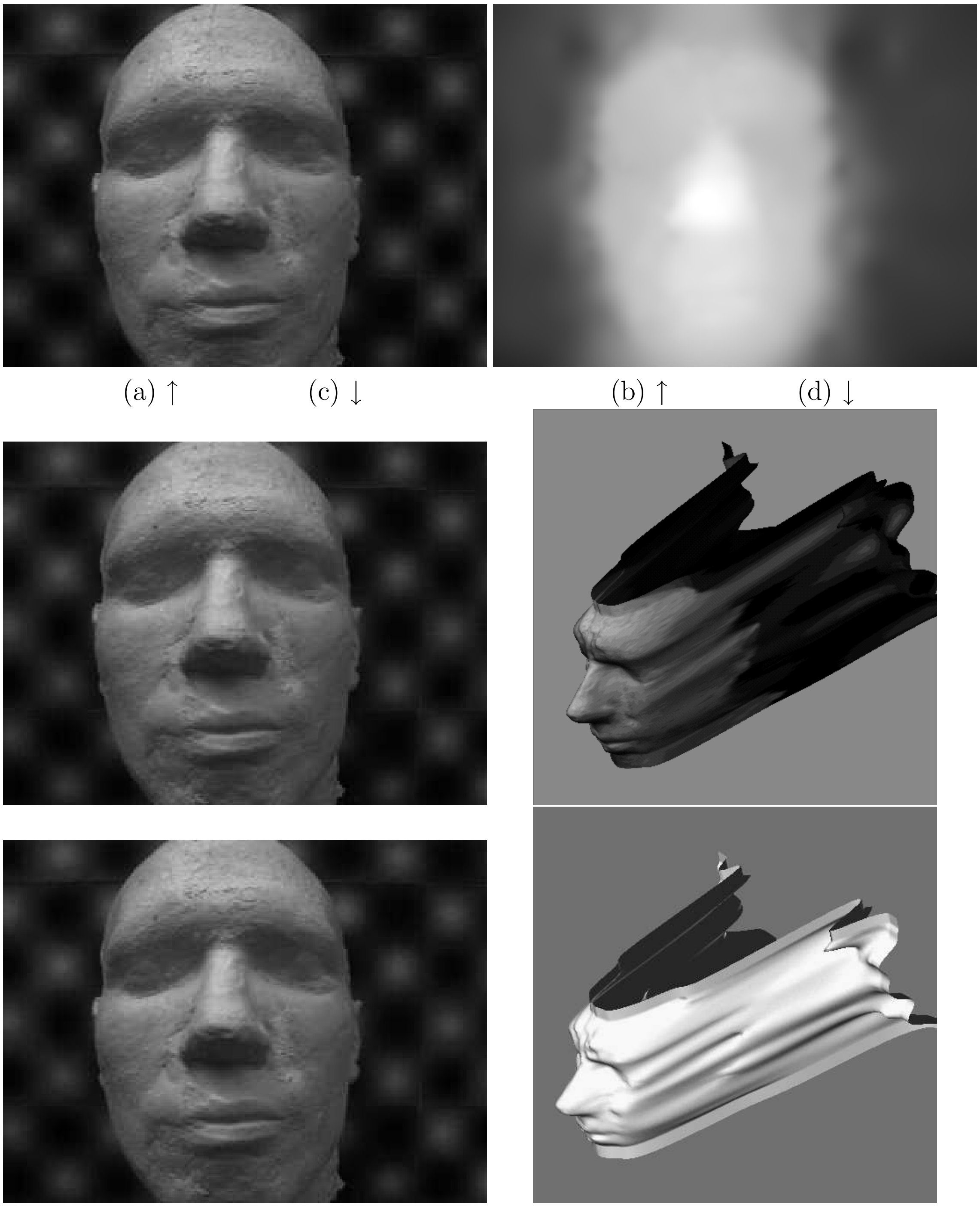


Figure 4-8: *The three input images (a,c,e) with collinear vertical translation. The estimated depth map (b) and 3D rendering of the surface (d,f).*

the optical flow estimates obtained in this manner with the flow estimates computed directly from two of the images using an ‘industry standard’ optical flow program. The program is based on code by Bergen & Hingorani of the Sarnoff Corp [7, 6]. We use two sets of images which have been chosen to be particularly difficult for optical flow programs.

Experimental Procedure

Figure 3-3 shows three images of a real scene: two paper cylinders, a plaster bust and a black metal bar in front of a background of vertical stripes. The two motions are pure translation, vertical and horizontal, parallel to the image plane. Figure 3-3a shows the recovered depth map for the sequence. The results are qualitatively correct. In Figure 3-3b the texture mapping was removed for clarity. The difficulties in the scene are explained in the introduction (Section 3.1) and expanded upon in the results section below.

From the depth map and motion the motion flow was computed using equations (equation 3.14). The flow field was also computed using the optical flow program and the results compared.

This experiment was repeated for a second set of images shown in Figure 4-12a,b,c. The main difficulty here is that most of the gradients are smooth a low contrast. The strongest edges appear at the occluding contour of the bust and these edges do not represent real features.

Results

Figure 4-9a shows the estimated flow computed from the recovered depth and motion. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero Y component. Figure 4-9b shows the magnitude of the flow vectors. Figure 4-9c shows the flow estimated from two images using an optical flow program. Near the vertical bars on the left the flow vectors have a strong Y component. This is due to the aperture problem. Even a large window (aperture) will see image gradients in only one direction giving no constraint on the Y component of the flow.

On the cylinder to the lower right of the bust the magnitude of the flow vectors is too small. This is due to the fact that the edges were horizontal and parallel to the horizontal epipolar lines. This means that the depth estimates for theses points are unreliable (Figure 4-9d). This problem does not occur in Figure 4-9b because two motions one horizontal and one vertical were use to compute the original depth map.

An enlarged detail of one of the more difficult regions is shown Figure 4-10. Near the intersections (in the image) of the diagonal line and the vertical bars the flow has a small Y component. This error occurs because the program 'tracks' the intersection point as if it were a real 'feature' point, but the lines do not intersect in space. Although smaller in magnitude this is the more significant error because the program will also give a high confidence to this value.

The results for the second set of images are shown in Figure 4-11. Errors in flow direction can be seen near the edges of the bust. The most significant errors are errors

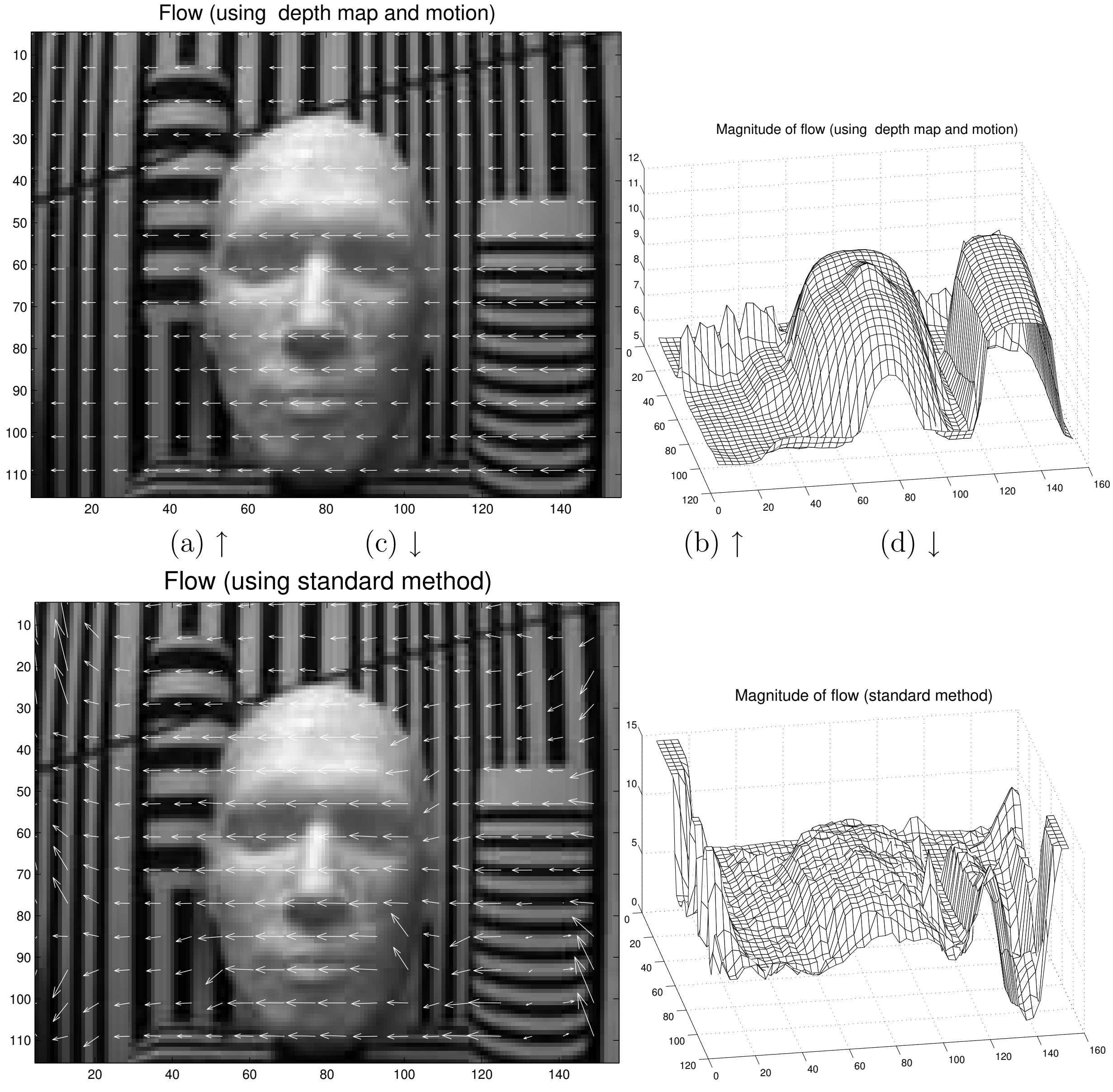
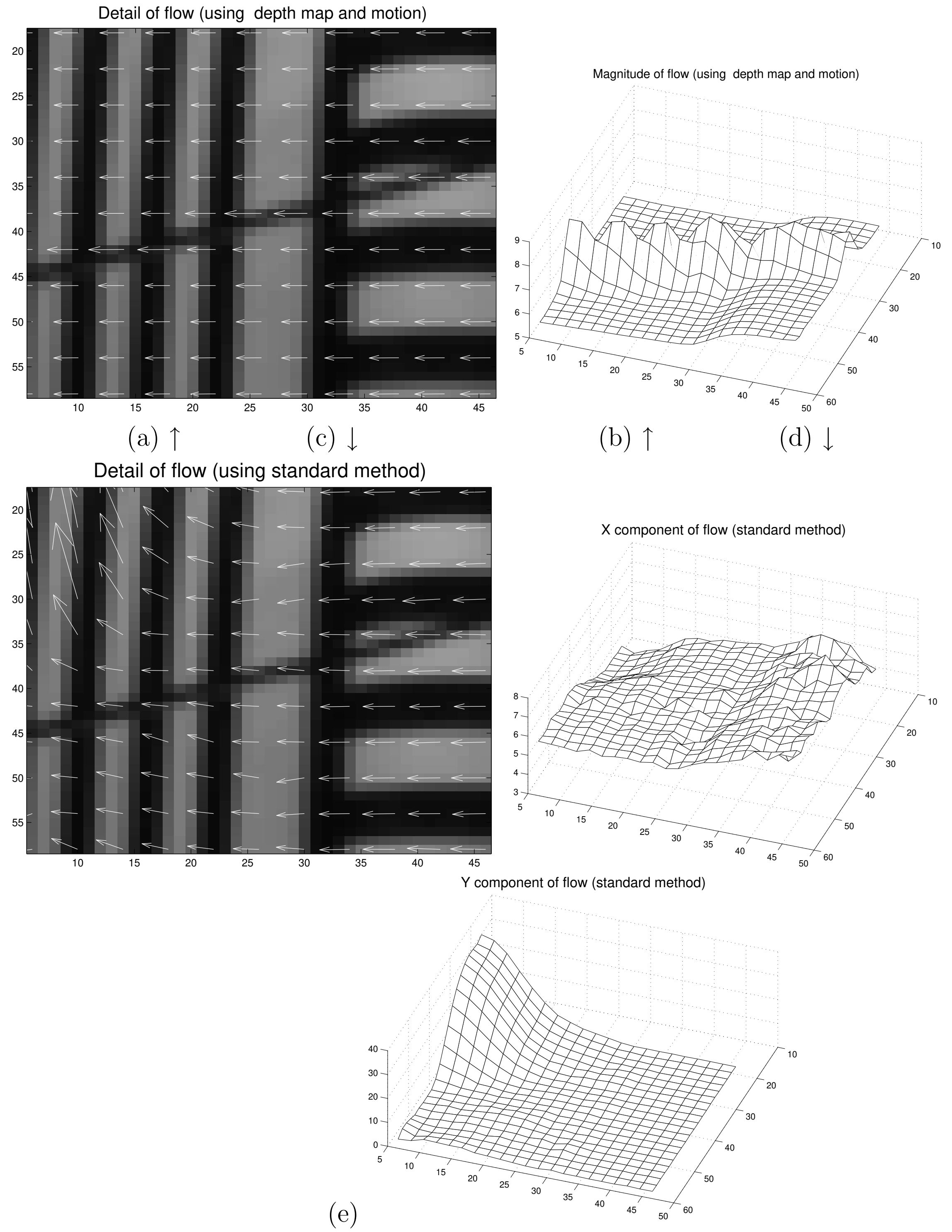


Figure 4-9: *Optical flow computed for Figure 3-1. (a) Correct optical flow computed using depth map and recovered motion. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero* Y *component. (c) Optical flow computed using code from Bergen & Hingorani. Near the vertical bars on the left the flow vectors have a strong* Y *component. This is due to the aperture problem. Even a large window (aperture) will see image gradients in only one direction giving no constraint on the* Y *component of the flow.*



(a) ↑ (c) ↓ (b) ↑ (d) ↓

(e)

Figure 4-10: *Detail of optical flow computed for Figure 3-1. Near the intersections (in the image) of the diagonal line and the vertical bars the flow has a small* Y *component. This error occurs because the program 'tracks' the intersection point as if it were a real 'feature' point, but the lines do not intersect in space. Although smaller in magnitude this is the more significant error because the program will also give a high confidence to this value.*

in magnitude of the flow in the background areas. These can been seen clearly by comparing the mesh plots.

This experiment was intended to highlight the problems with optical flow techniques and how they are overcome by using direct methods. While this was not a simple 'straw man' and the optical flow program was one of the best available a full system would based on optical flow would have more components intended to overcome the problems described. Together with flow vectors the optical program would compute a confidence ellipse for each vector. These estimates would then be combined using robust methods to reject outliers during the motion computation. After the camera motion was estimated the flow would be recomputed using the epipolar constraint and the new flow would be used to compute the depth. Probably three views would be used with different camera motions. Building such a worthy competitor and fine tuning it to these image sets is of course beyond the scope of this thesis but this description does give an idea of the complexity of such a system.

4.3 Translation and Rotation

4.3.1 Experiment 4: Translation and Rotation (Reanalysis of Data from Stein & Shashua, CVPR97)

In this experiment we test the modified algorithm for recovering both translation and rotation (Section 3.7.5) on the same data set used by Stein & Shashua [74].

Experimental Procedure:

The images used by [74] were taken with an $8.5mm$ lens in the following way: The camera was translated first vertically ($10mm$) and then horizontally ($5mm$) to the right. At this third position the camera was rotated to various angles ranging from -4.0^o to 1.0^o. The depth in the scene ranged from $170mm$ to $400mm$.

Figures 4-12a,b,c show the three input images for pure translation. Figure 4-12d shows the image of the scene after the camera has been rotated -4.0^o. The flow due to a rotation of -4.0^o is much greater than the flow due to the translation even for points nearest the camera. The direct of the flow due to rotation by a negative angle was in the same direction as the image flow induced by the translation.

Results:

Figure 4-13a shows the recovered depth map using images in Figures 4-12a,b,c. Figures 4-13b,c,d show the 3D rendering of the depth map. In this case the true motion was pure translation. Another example of the reconstruction for the case when the true motion was 1.8^o was shown in Chapter 1, Figure 1-2.

Figure 4-14 shows the recovered rotation estimates for true rotations ranging from -4.0^o to 0.8^o. Outside this range the old version of the algorithm did not converge. Using the new algorithm the rotation estimates are within 5% of the correct value. There is no scaling error as was observed using the old method.

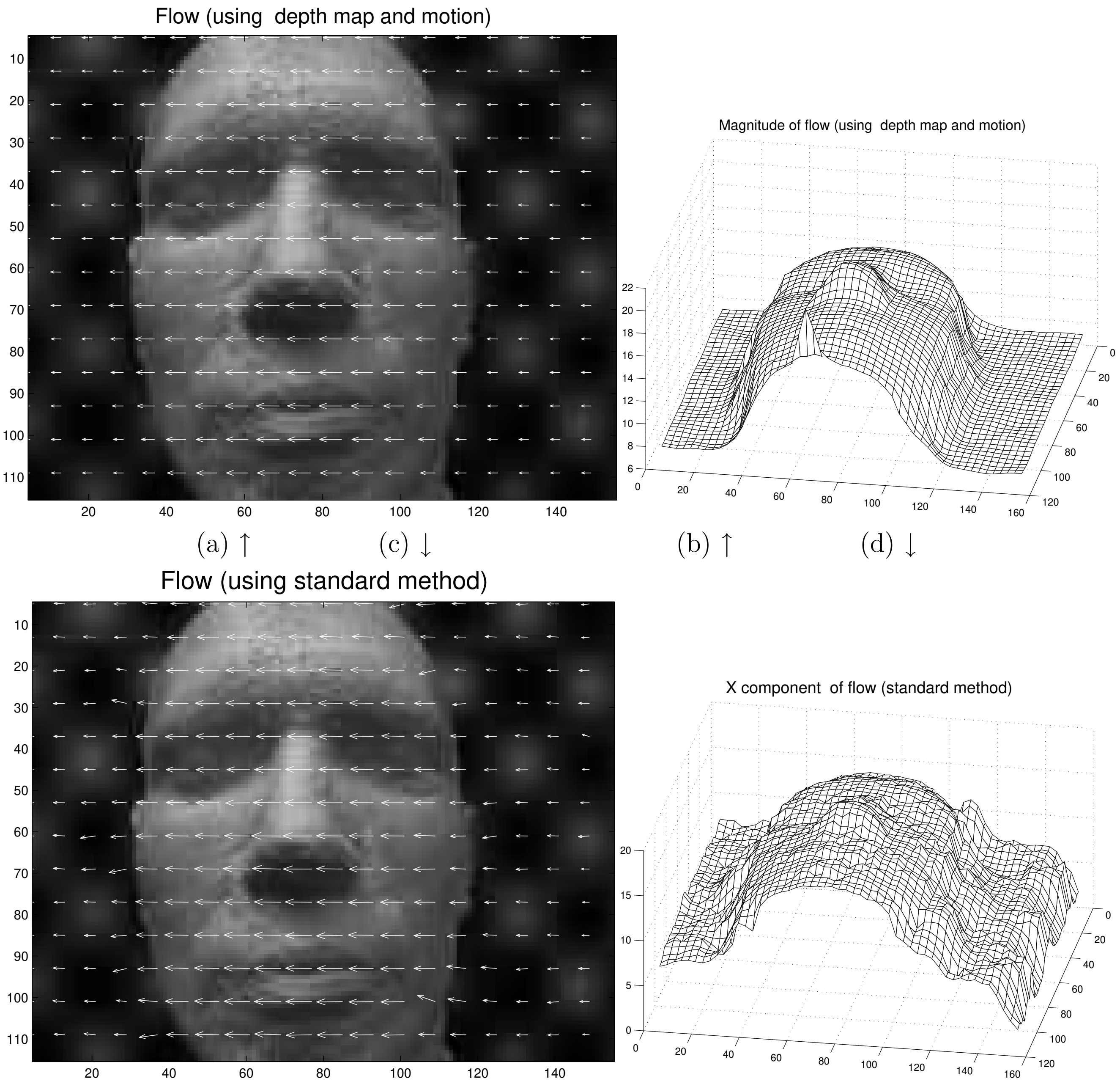
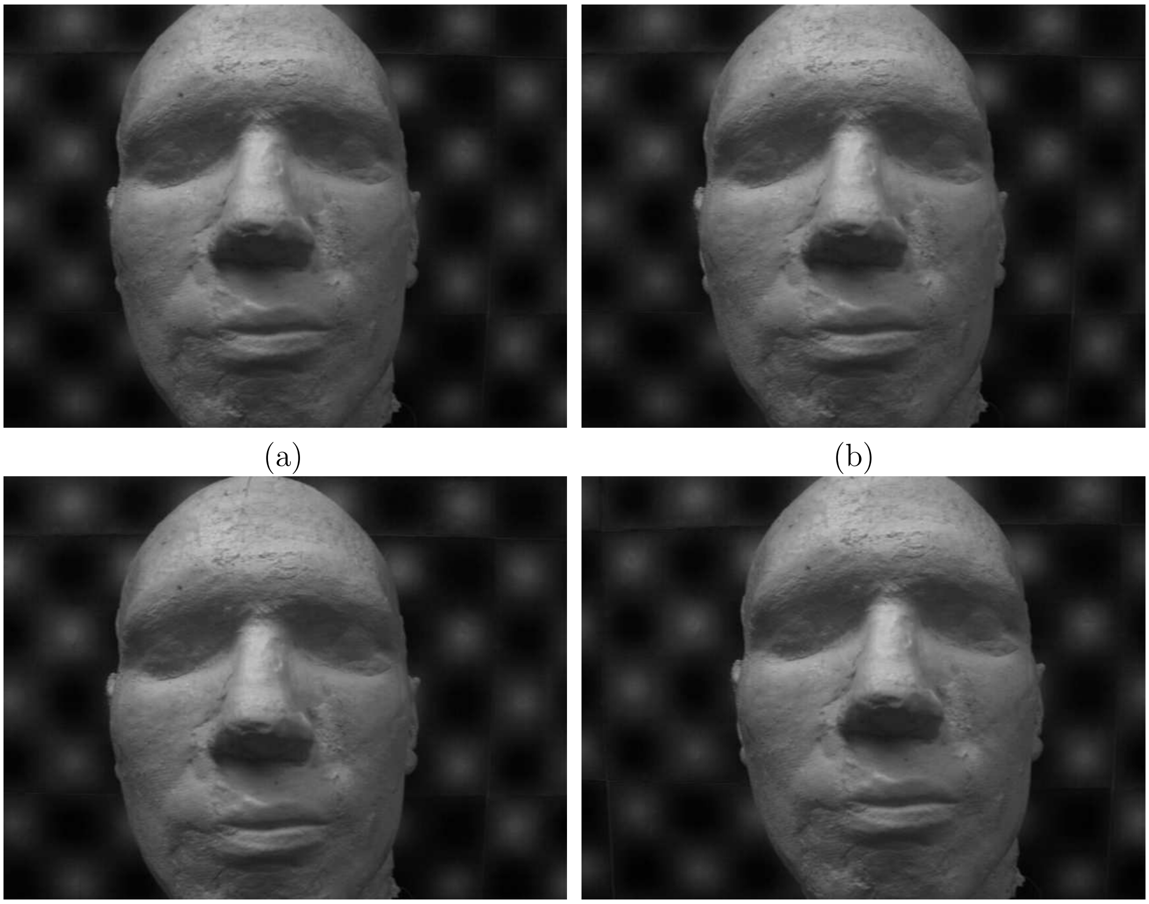


Figure 4-11: *Optical flow computed for Figure 4-12. (a) Correct optical flow computed using depth map and recovered motion. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero Y component. (c) Optical flow computed using code from Bergen & Hingorani. Errors in flow direction can be seen near the edges of the bust. The most significant errors are errors in magnitude of the flow in the background areas. These can been seen clearly by comparing the mesh plots.*



(a) (b) (c) (d)

Figure 4-12: *The three input images (a,b,c) with pure translation. (d) another image taken at the same location as (b) with -4.0^o rotation. Note that the image flow due to translation is much smaller than the flow between (b) and (d) which is due to the camera rotation.*

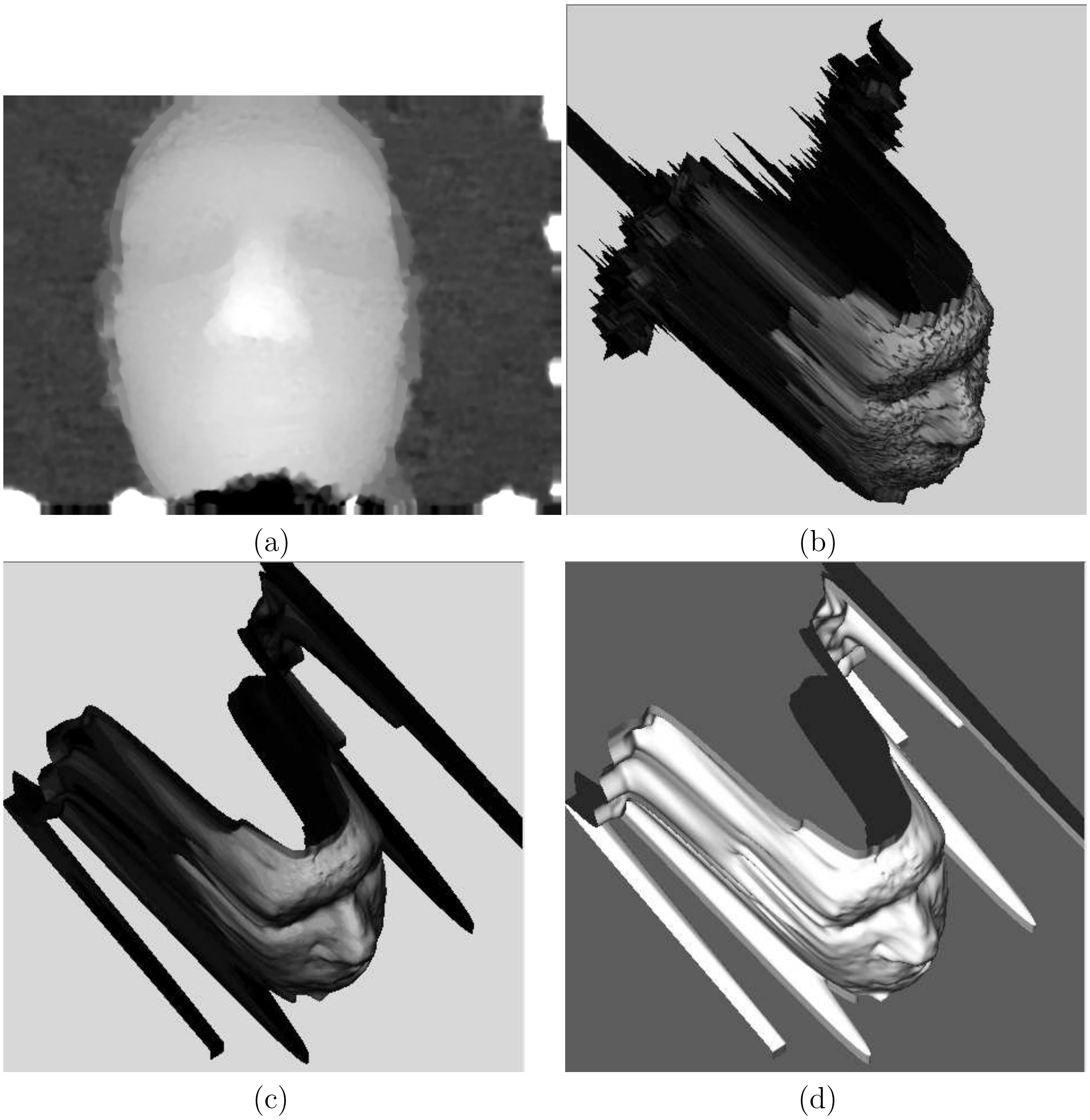


Figure 4-13: *The estimated depth map (a) and 3D rendering of the surface (b). There are errors in the depth estimates around the edges. Depth estimates were computed using* 9×9 *region and a local constant depth model. In (c) and (d) a* 3×3 *region was used to compute the depth together with a weak smoothness constraint.*

The new algorithm also has a wider range of convergence. The original images were not available for larger rotations but a similar setup yielded good results for rotation for the range of -5.0^o to 5.0^o. Since rotations can create large amounts of image flow, key to the successful convergence are strong signals with low spatial frequency such as the white head on a darker background. This leads to convergence at the coarse level which is then used as the starting point for motion estimates at finer levels. Strong repetitive high frequency patterns such as checker board patterns are more difficult. The convergence range for the scene in Figure 4-6 is only $\pm 0.5^o$.

4.3.2 Experiment 5: Motion Heading Estimation with Rotation

The aim of this experiment is to test the motion estimation algorithm over a wide variety of motion directions when the motion included some rotation. For example, this experiment tests whether the algorithm works when the direction of motion is inside and outside the FOV. We also find the error created by neglecting even small amounts of rotation. The two translations are non-collinear.

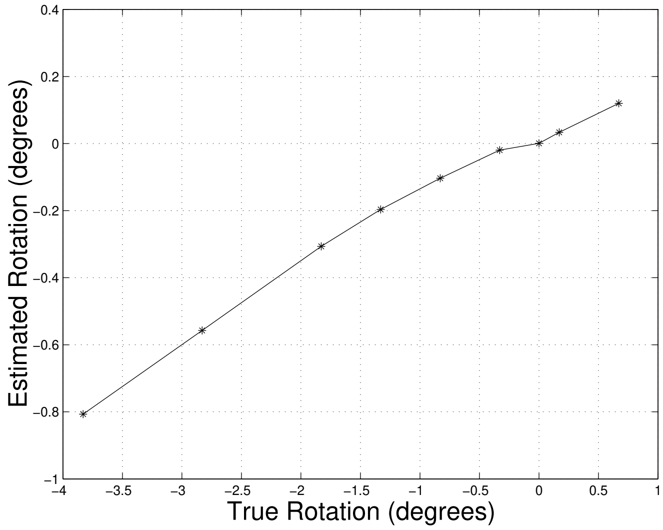
Experimental Procedure:

The camera setup is similar to that of experiment 1 (Section 4.2.1). The camera is rotated to a particular heading direction ($0^o, 10^o, ..., 90^o$). The camera was translated vertically and then horizontally. After the horizontal translation the camera was rotated $\pm 0.5^o$ and $\pm 1.0^o$. Therefore, for each image heading, we have an image triplet which is translation only and then triplets where the second motion also includes some small rotations. The rotation axis of the stage passed within $5mm$ of the camera center of projection thus rotations of 1.0^o produce translations of less that $1mm$. Figure 4-15 shows input images where the horizontal translation heading was 60^o to the camera optical axis.

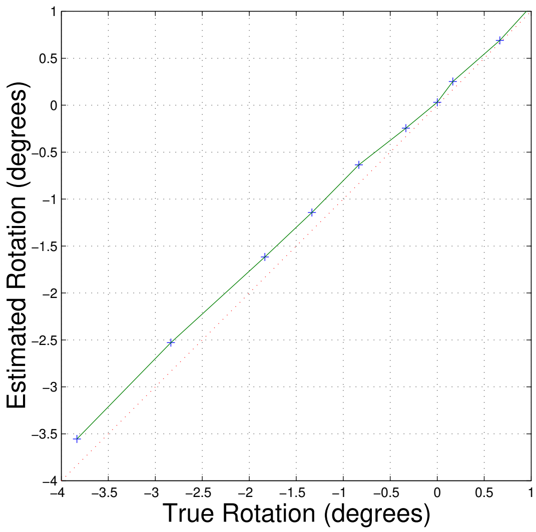
Results:

Depth and motion estimates were computed either assuming pure translation or allowing possible rotation. The results for various motion and motion-assumption combinations are shown in Figure 4-16, Figure 4-17 and Figure 4-18. We notice that for even a small rotation (0.5^o) the heading estimate is off by over 10^o if we do not take the rotation into account. When we estimate both rotation and translation the motion estimates are good (RMS error 2.1^o) although not as good as those obtained from a purely translating camera under the pure translation assumption (RMS error 1.44^o).

Figure 4-19a shows the depth map estimated from the images in Figure 4-15a), Figure 4-15c and Figure 4-15d allowing for both translation and rotation. The depth map is qualitatively correct even though the second camera motion included -1.0^o rotation. Figure 4-19b shows the depth map estimated assuming pure translation when in fact the second camera motion included 0.5^o rotation.



(a)



(b)

Figure 4-14: *Estimated rotation as a function of true rotation. (a) Old results: Although there is the correct linear relationship there also appears to be a significant scale error (note Y axis). (b) New results - the rotation estimates are correct.*

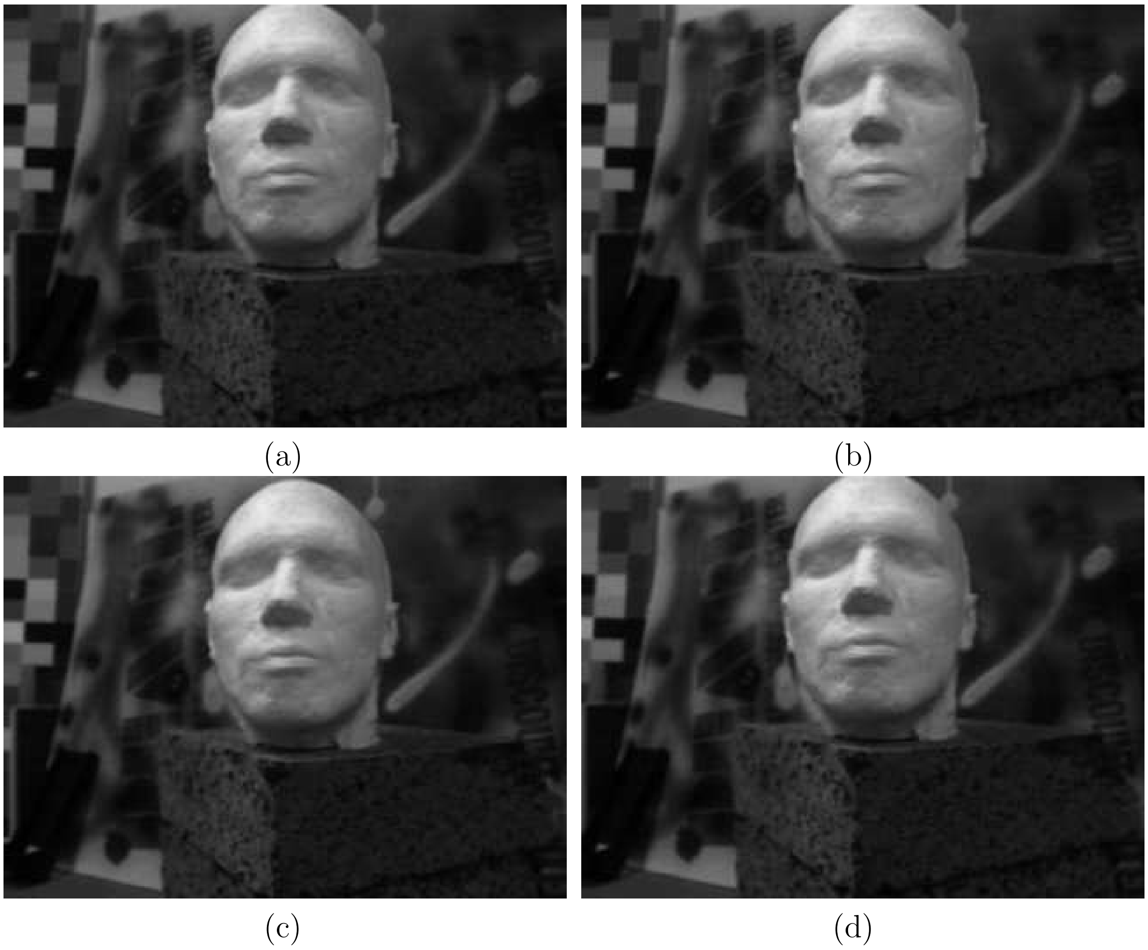


Figure 4-15: *Four of the input images used in the experiment to test motion heading estimation in the presence of rotation. Motion (a) to (c) is vertical motion. Motion (a) to (b) is horizontal motion forward and to the left (*60^o *from the camera optical axis). Motion (a) to (d) has the same translation as (a) to (b) but also with a rotation of* 1.0^o*. In this case the flow due to rotation of* 1.0 *cancels out the flow due to translation in the area of the face and more than cancels out the flow in regions of greater depth. (Lens* $8.5mm$ *lens /* 52^o *FOV)*

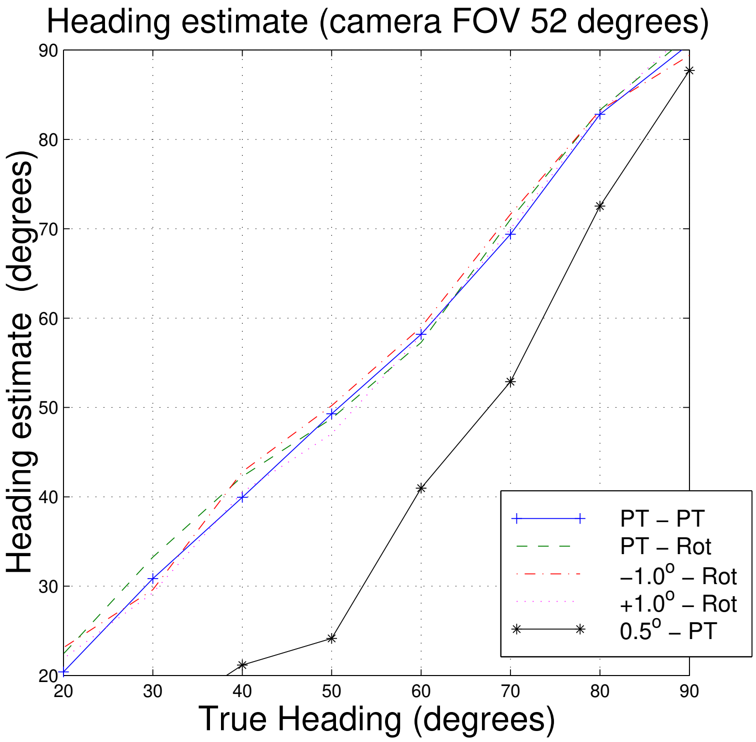


Figure 4-16: *Heading estimates from a moving camera. (PT - PT) Both true motion motion model was pure translation.* (0.5^o *- PT) True motion included* 0.5^o *and motion model was pure translation. Note that even a small rotation, such as* 0.5^o*, is enough to create large heading estimation errors if rotation is not taken into account. (PT-Rot,* $+1.0^o$ *- Rot,* -1.0^o*- Rot) Motion model included rotation Actual rotations were* $0^o, 1.0^o, -1.0^o$*.* (8.5_{mm} *lens /* 52^o *FOV)*

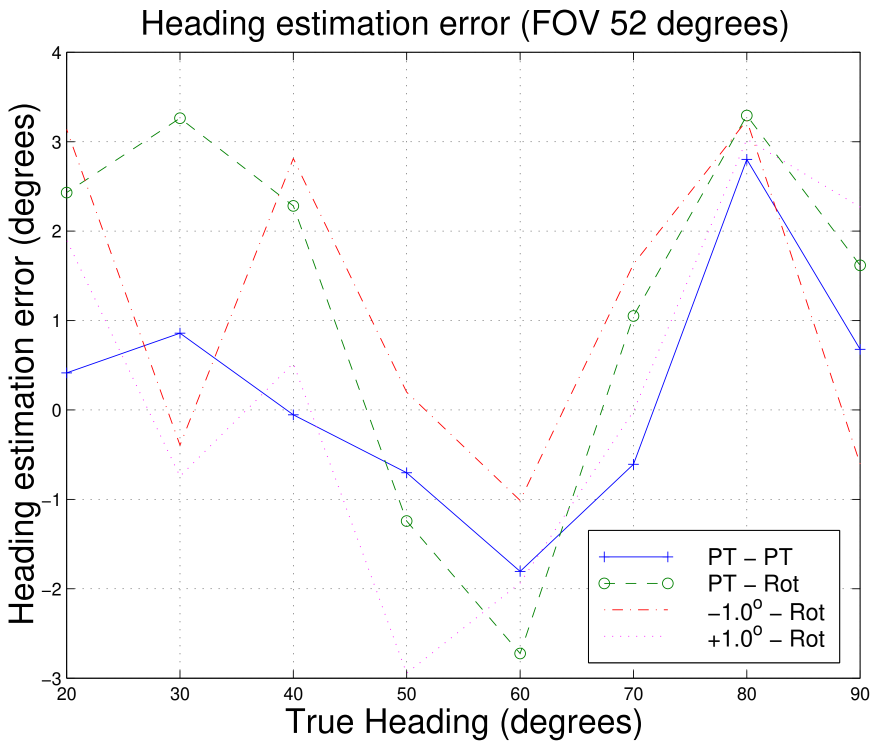


Figure 4-17: *Heading estimate error: the difference between heading estimate and motion stage reading. Even in the presence of camera rotation (*$+1.0^o$ *- Rot,* -1.0^o*- Rot) the heading estimation errors are small (RMS error* 2.1^o*) although not as good as obtained from purely translating camera under the pure translation assumption (PT-PT) where the RMS error was* 1.44^o*. (8.5mm lens /* 52^o *FOV)*

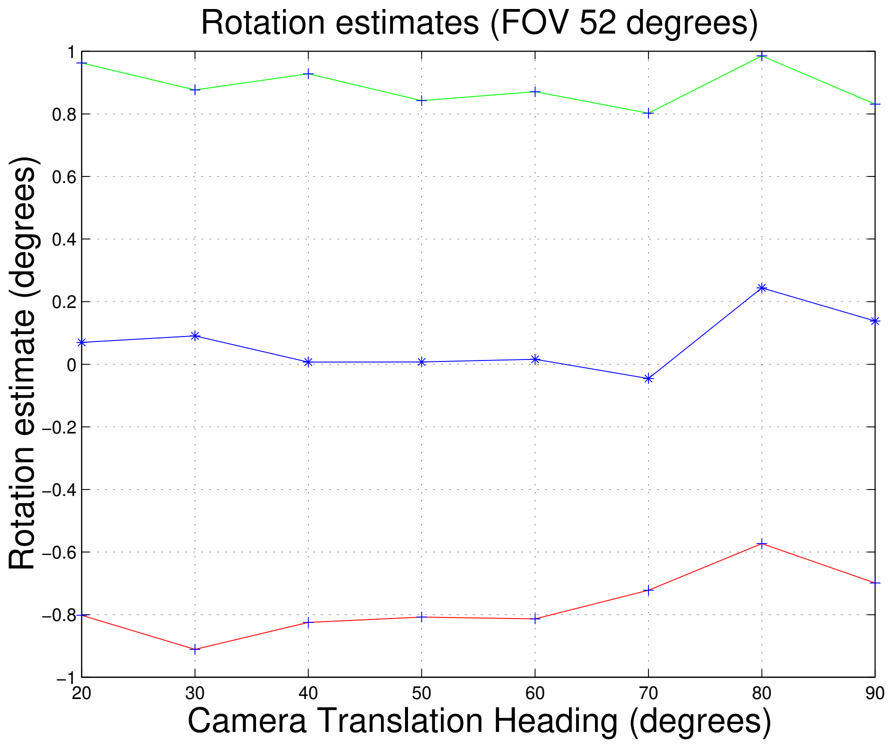
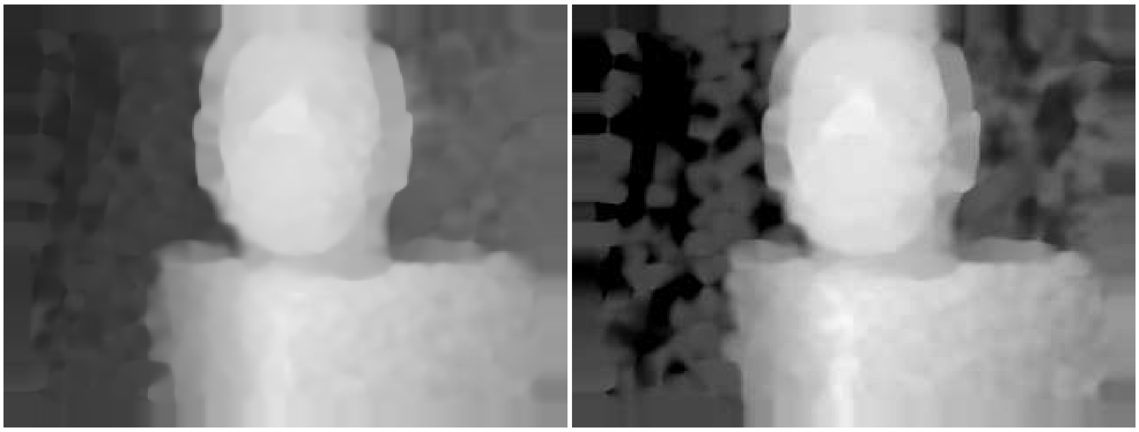


Figure 4-18: *Rotation estimates obtained from a camera translating in different translation directions with some added rotation. Actual rotations were* $0^o, 1.0^o, -1.0^o$*. (camera with* 8.5_{mm} *lens /* 52^o *FOV)*



(a) (b)

Figure 4-19: *(a) Depth map estimated after computing both translation and rotation. The second camera motion included* -1.0^o *rotation. (b) Depth map estimated assuming pure translation when the second camera motion included* 0.5^o *rotation. Notice the depth errors in the background.*

4.3.3 Experiment 6: Euclidean Structure Estimation in the Presence of Rotation.

In this section we evaluate the Euclidean structure estimation when the motion includes rotation. We use a simple scene, a cube, in front of a planar background. We perform the 3D reconstruction of the scene. We then measure whether the right-angles between the cube faces are correctly recovered.

Experimental Procedure

Figure 4-20 shows the simple scene of a textured cube in front of flat, low-contrast, background. The camera ($8.5mm$ lens / 52^o FOV) was translated vertically and then horizontally parallel to the image plane. The camera was then rotated to $+2.0^o$ and -1.0^o from its original heading.

Since we were interested in the 3D structure and not jut motion estimates, after estimating both structure and motion we went back down the pyramid and recomputed the structure while keeping the motion estimates constant (Section 3.7.6). The depth estimates before and after the second pass are shown in Figure 4-20d and Figure 4-20f respectively. In Figure 4-20d there is a "hole" on the right of the depth map. This is due to the appearance of a white patch near the border of the image. As with occlusions this sudden appearance violates the constant brightness assumption. The situation is fixed by the second pass. In general points near the edge of the depth map are unreliable.

Results:

The rotation estimates were 1.68^o and -0.67^o for true rotations of $+2.0^o$ and -1.0^o respectively. The 3D rendering of the cube is shown in Figures 4-21 and 4-22. The wire-frame rendering of the overhead views (Figures 4-21b and 4-22b) show the recovered angle between the cube faces. The estimated angles were 95^o and 86^o respectively. Figure 4-21d shows a side view of the cube. The estimated angle between the top and side face of the cube is 90^o.

4.4 Experiments with a Hand-held Camera

4.4.1 Experiment 7: Euclidean Structure Estimation Using a Hand-held Camera

In previous experiments the camera was mounted on a motion stage so that ground truth motion values were available. In this experiment the camera was hand-held to show that the system can deal with the more natural situation. Simple objects were used to create the scene so that it would be easier to evaluate the shape reconstruction.

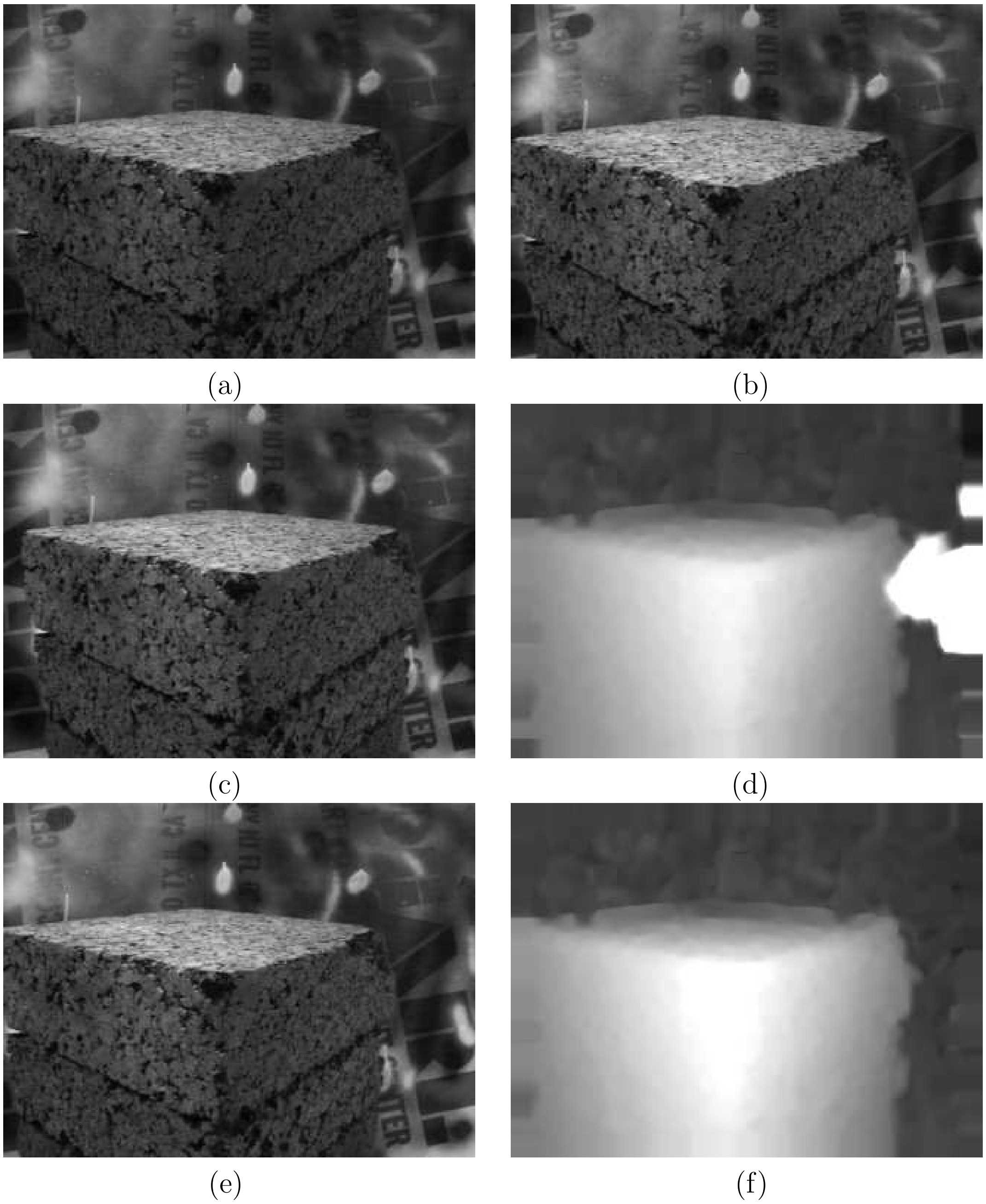
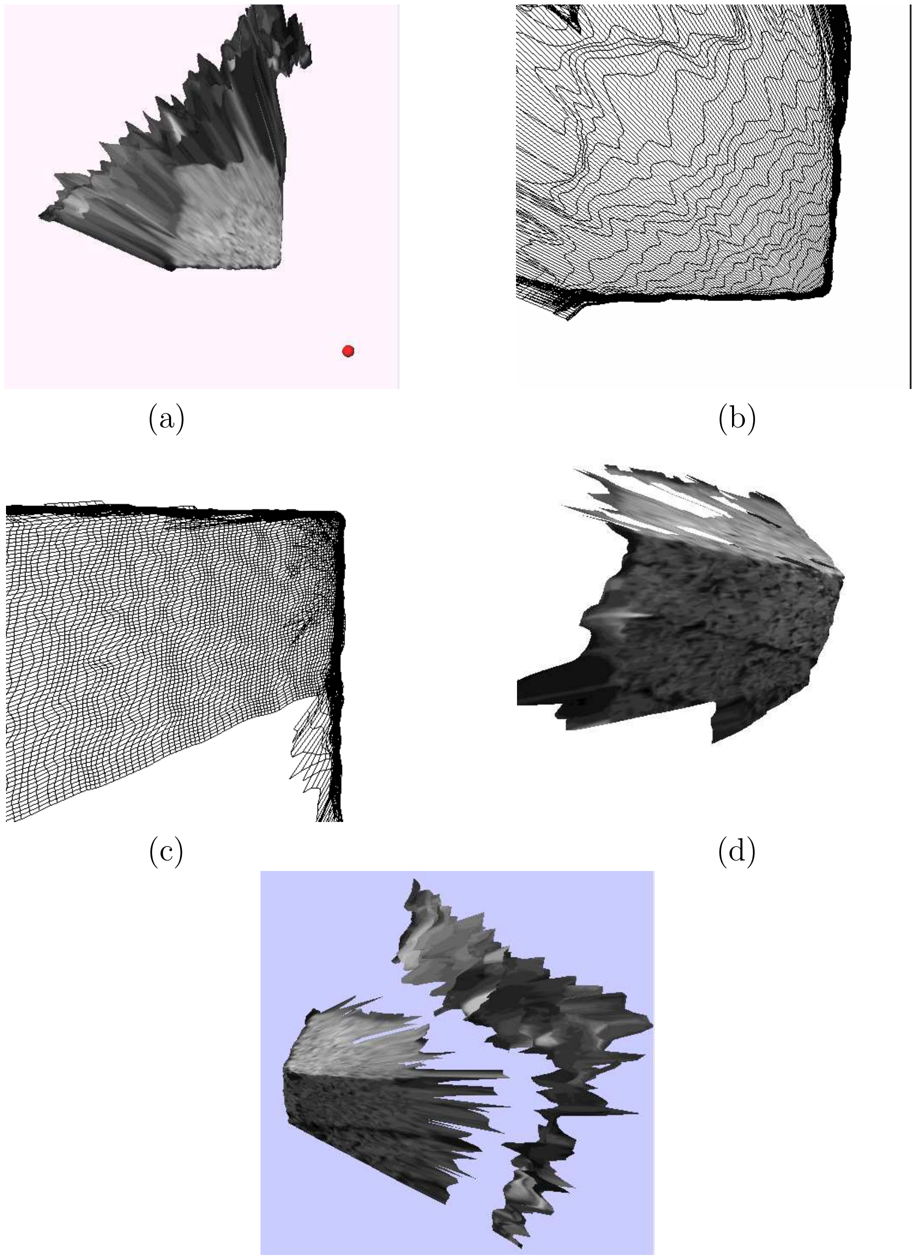


Figure 4-20: *A simple scene used to test 3D Euclidean reconstruction: a cube constructed from cork blocks in front of a flat background. Motion (a) to (b) was vertical. Motion (a) to (c) was sideways translation with* -1.0^o *rotation. Motion (a) to (e) was same sideways translation but with* $+2.0^o$ *rotation. (d) initial depth map from images (a), (b) and (c). Note how the disappearance of a white blob in the background near the right edge of the image creates an error in one region of the depth map. (f) After second pass of depth estimation the "hole" is fixed. (see text)*



(a) (b) (c) (d) (e)

Figure 4-21: *3D rendering of the recovered scene. The camera motion included* -1.0^o *rotation. (a) Overhead view. The estimated camera position is marked by the sphere in bottom right-hand corner. (b) An enlarged view of the corner of the cube. The angle between the faces of the cube is estimated* 95^o *instead of* 90^o*. (c) Side view of the segmented cube from up and to the side. (d) enlarged wire-frame side view of the cube. The estimated angle between the top face of the cube and the right face is* 90^o*. Another view of the scene showing the foreground/background segmentation. The depth estimates of the back panel are noisy because they are further away from the camera and because the texture does not have strong gradients.*

(a) (b)

Figure 4-22: *3D rendering of the recovered cube. The camera motion included* $+2.0^o$ *rotation. The cube has been segmented from the background using depth based foreground/background segmentation. (a) View from up and to the side. (b) Wire-frame rendering of the overhead view. From the overhead view the estimated angle between the cube faces is* 86^o *instead of* 90^o

Experimental Procedure

A progressive scan camera (Pulnix TMC9701) with a wide angle lens (4.9mm lens / 82^o FOV) was hooked up directly to the frame grabber of an SGI Indy workstation. The cameras help in the hand with no tripod. An image sequence of 30 frames was captured while the camera was moved up and down and side to side the combination of which produced a circular motion. An effort was made to keep the camera rotation down to a minimum.

From the image sequence three images were selected which seemed to give two distinct motion directions.

Results

Figure 4-23a,b,c show three input images of a cube. The motion and depth was estimated for three different cases. Figure 4-23d shows the estimated depth map allowing for both rotation and translation and taking into account the radial distortion. 3D renderings are shown in Figure 4-24. The rotation estimates for the two motions are shown in Table 4.1. Figure 4-23f shows the estimated depth map allowing for both rotation and translation but neglecting radial distortion. The resulting depth map is qualitatively the same but the measured angles between the cubes faces are larger and further from the true value of 90^o (Figure 4-25). If we assume pure translation the resulting depth map (Figure 4-23e) appears to be qualitatively the same but the 3D renderings (Figure 4-26) clearly show that the results are more noisy and further from true Euclidean reconstruction. Figure 4-27 and Figure 4-28 show results from a more complex scene.

Errors in depth reconstruction when we neglect the rotation are to be expected according to [14]. They show that for small total rotation angles ($|w|$) the error in

Table 4.1: *Rotation estimates (in degrees) from the images in Figure 4-23. The camera was hand-held so ground truth values of the motion are not known.*

	w_x	w_y	w_z
Motion 1	-0.7809	0.2262	-0.5520
Motion 2	-0.0538	0.2684	1.4246

depth due to an error in rotation ($|\delta w|$) is given by:

$$\frac{\delta Z}{Z} \approx \left(\frac{f}{|u|}\right)(|\delta w|)$$

where f is the focal length, $|u|$ is the magnitude of the image motion. In our case by assuming pure translation we get a rotation error of 1.45^o or $0.025rad$ (Table 4.1). The average image motions were on the order of $15pixels$ and the focal length $f \approx 180$. So we expect:

$$\frac{\delta Z}{Z} \approx 0.3.$$

[14] refers to depth from image motion. The errors we observe are not so large but are definitely noticeable. We also note that even if we neglect the rotation none of the depth estimates were negative so a *depth is positive* constraint, as suggested by [19], would not help in this example.

4.4.2 Experiment 8: Outdoor Scenes - Church Wall

In this experiment we tested the algorithm on outdoor scenes. These experiments are important because they show that the photometric constraints can be used in uncontrolled lighting conditions. Figures 4-29a,b,c show three images of a small part of a church wall. The camera was Sont Hi8 camcorder with the lens open as wide as possible (FOV 40^o). The camera was mounted on a tripod. Moving the camera horizontally ($a \rightarrow b$) was quite smooth but moving the camera up and down included about 0.5^o rotation. Figure 4-29d shows the recovered depth map which can be seen to be qualitatively correct. The Euclidean 3D renderings of the depth map are shown in Figure 4-30.

4.4.3 Experiment 9: Outdoor Scene 2 - Side Entrance

Figure 4-31 shows three views of the side entrance to a building. The camera was a progressive scan camera (Pulnix TMC9701) with a wide angles lens (4.9mm / FOV 82^o). The camera output was recoded on a Sony Hi8 camcorder. The camera was mounted on a tripod and an effort was made to reduce the amount of rotation, but rotation was 0.32^o. Figure 4-31d shows the recovered depth map. The 3d Euclidean rendering of the depth map is shown in Figure 4-32.

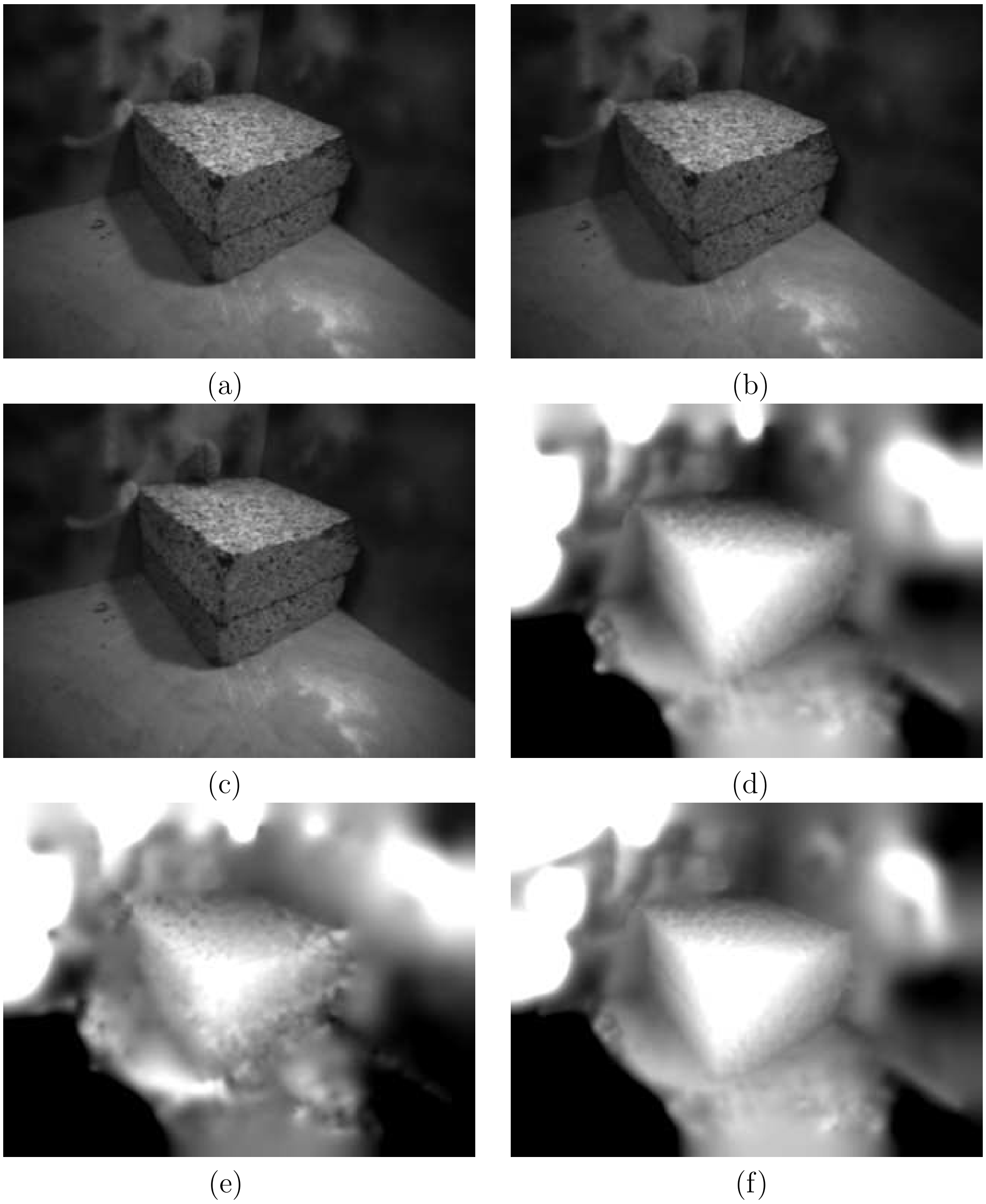
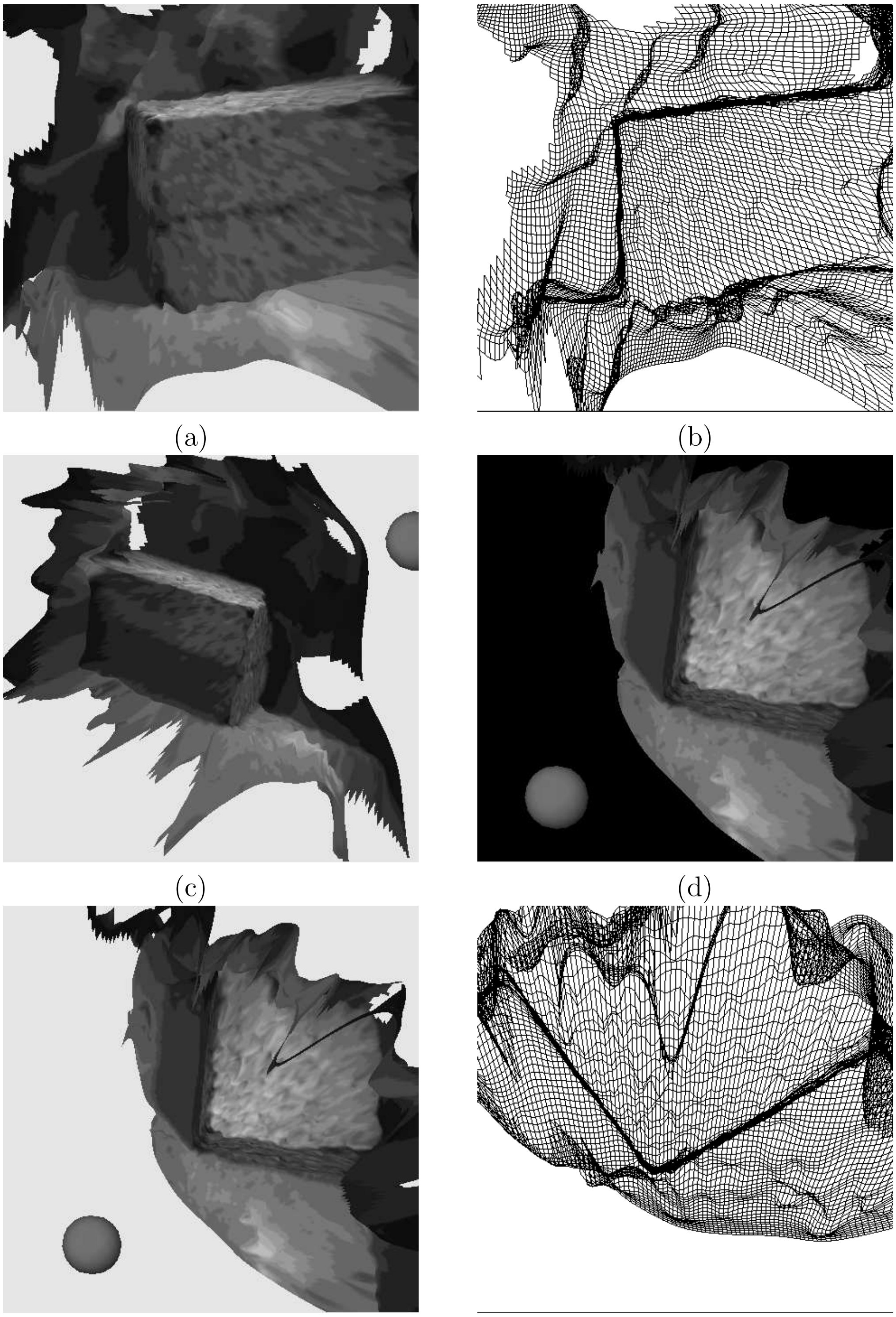


Figure 4-23: *A simple scene used to test 3D Euclidean reconstruction. Camera was hand-held so exact motion between images (a), (b) and (c) is not known. (d) Depth reconstruction allowing for both rotation and translation with radial distortion correction. (e) Assumes pure translation. (f) Allows for both rotation and translation but no radial distortion correction. The cube and it's surrounding are recovered well while textureless areas near the boundaries have large errors. The three depth maps look qualitatively the same but 3D renderings in following pages clearly show the differences.*



(a) (b) (c) (d) (e) (f)

Figure 4-24: *3D renderings of the depth map in Figure 4-23d. The motion estimation allowed for rotation and translation and the input images were preprocessed to correct for lens distortion. (a,b,c) side views. (d,e,f) top views. The estimated angles between cube faces are (b)* 96^o *and (f)* 97^o*.*

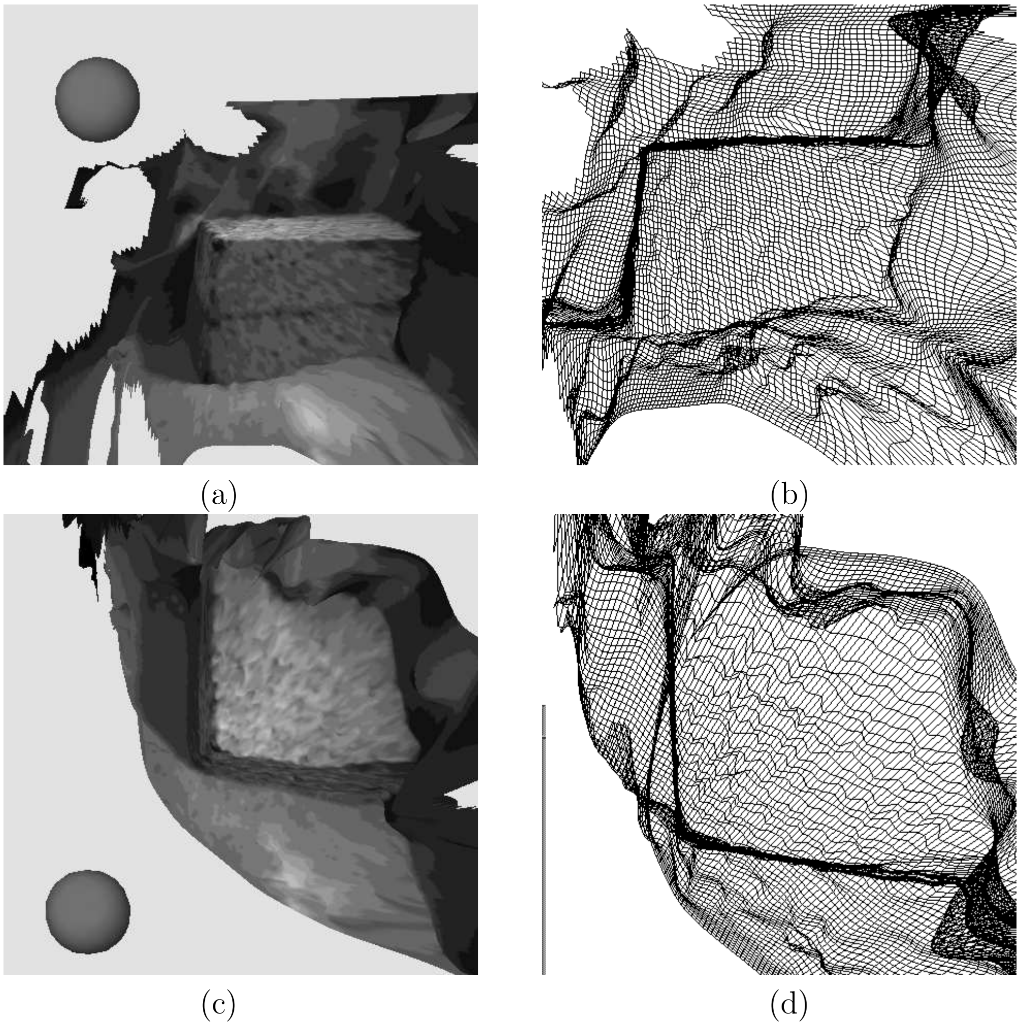
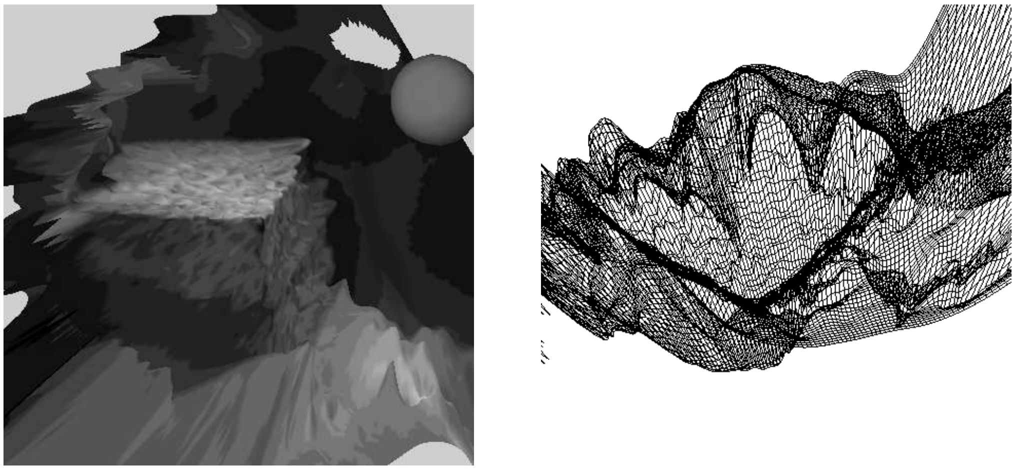


Figure 4-25: *3D renderings of the depth map in Figure 4-23f. The motion estimation allowed for rotation and translation. There was no correction for lens distortion. The estimated angles between cube faces are (b)* 97^o *and (d)* 101^o.



(a) (b)

Figure 4-26: *3D renderings of the depth map in Figure 4-23e. The motion estimation assumed pure translation only. The reconstruction has a lot more noise. The edges are less clean, especially the edge between the left and top faces. Also large errors are noticeable on the ground plane. The estimated angle between the side faces (b) is* 105^o.

4.5 Summary of Experimental Results

Using the proposed method for direct estimation of motion and structure we have found that heading direction can be estimated to within a few degrees for both purely translating motion and for motion that includes a few degrees of rotation. The estimates of camera rotation are within a fraction of a degree from the true rotations. The estimation of rotation is critical. If we neglect even small rotations (0.5^o) both heading estimates and structure estimates have large errors.

The flow due to even relatively small rotations can be quite large. This causes aliasing problems and therefore the program fails to converge on images with highly repetitive texture. While the problem is important in the case of pure translation as well, it is the typically larger image motions that make the problem more severe when rotations are involved. One solution to this problem might be to first find a rotation that best aligns the images (perhaps using a pyramid version of Horn & Weldon [34]) and only then try and find the translation and residual rotation. While this is similar to the idea of prealligning a plane it does not remove all rotation but on the other hand leaves us in the Euclidean realm.

The program converges well for non-collinear purely translating motion. In the presence of collinear motion the magnitude of the translation is more critical. In the presence of rotation the program converges well if either the rotational flow does not overwhelm the translational flow or if there is a strong signal with a low spatial frequency. That is why the program work well with the sequences in Experiment 4 and Experiment 5 but does not work well with the sequence in Experiment 1.

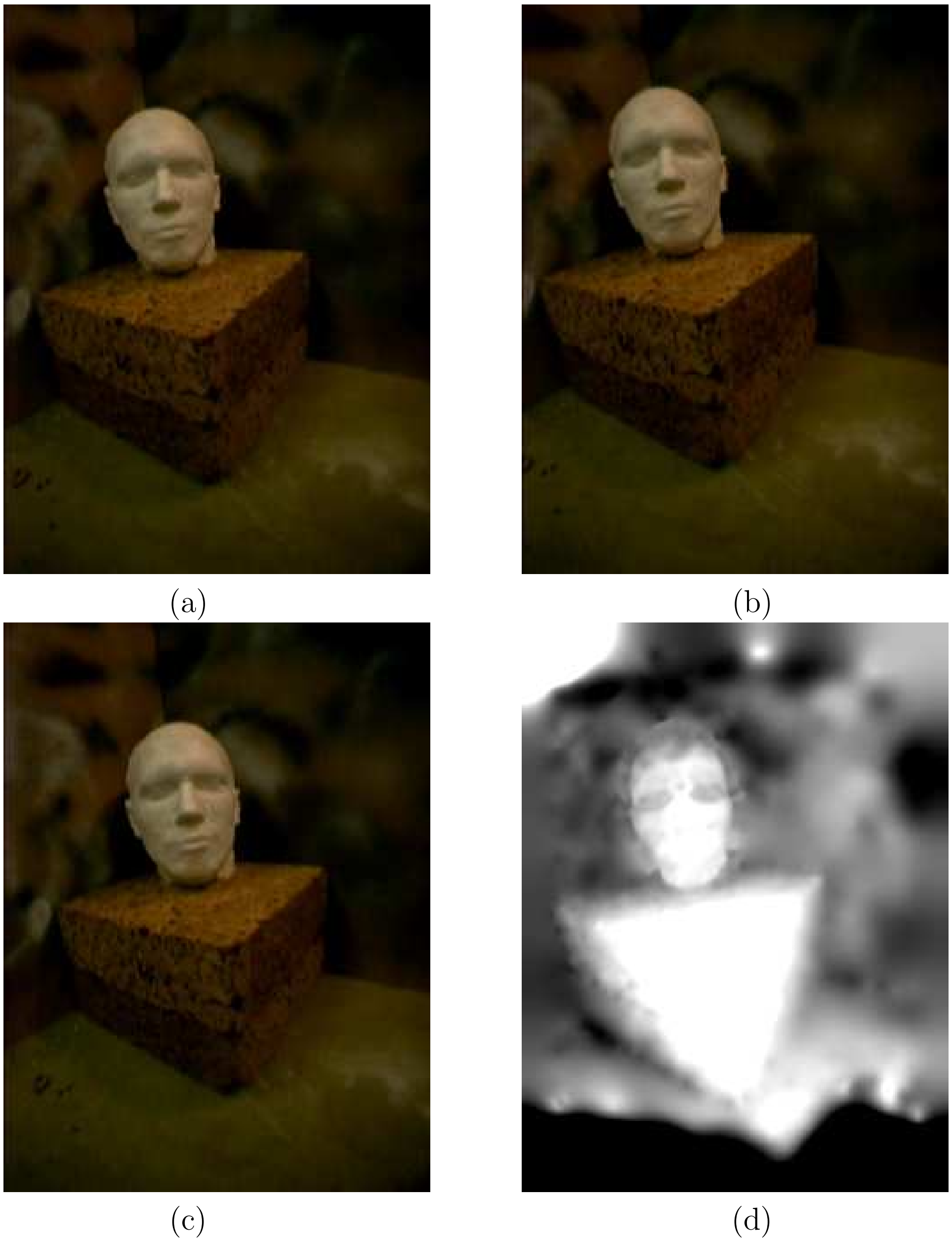
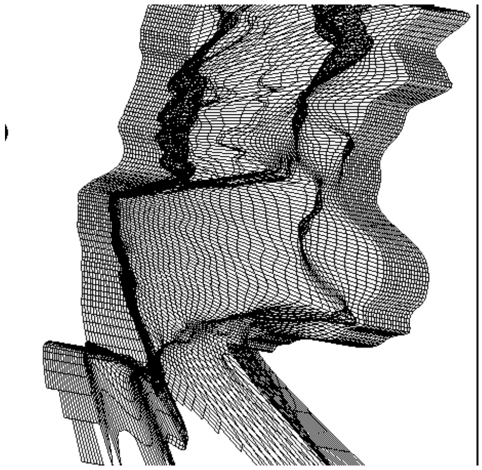


Figure 4-27: *Three input views from a hand-held camera and the resulting depth map. The motion model included both rotation and translation.*

(a)



(b)

(c)

Figure 4-28: *3D renderings of the depth map in Figure 4-27d.*

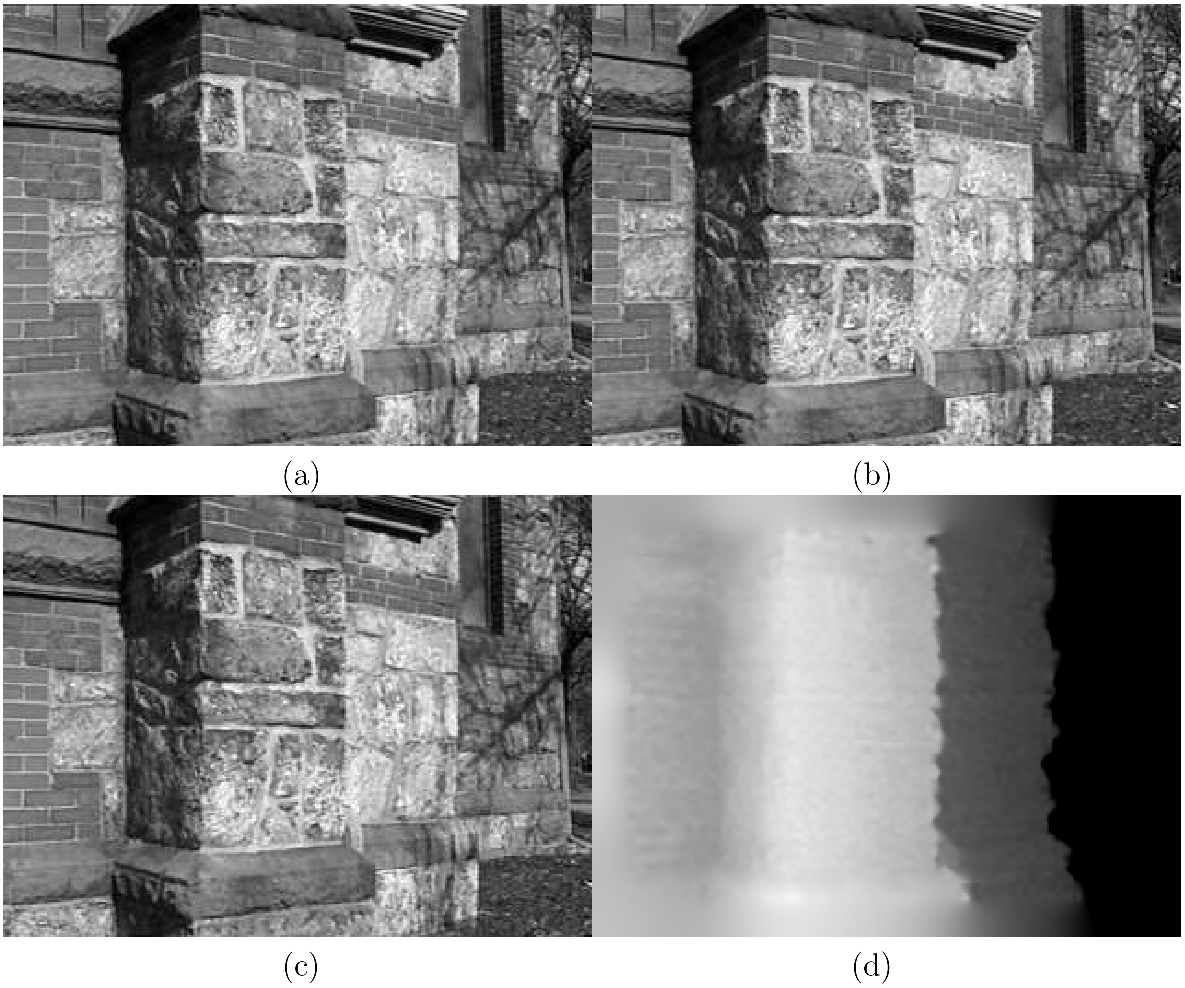
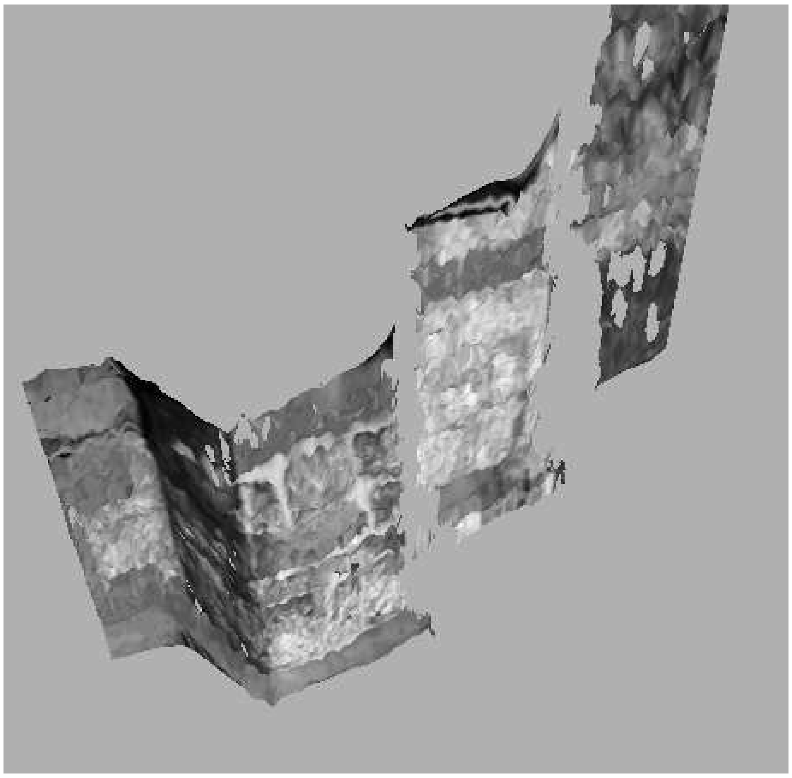
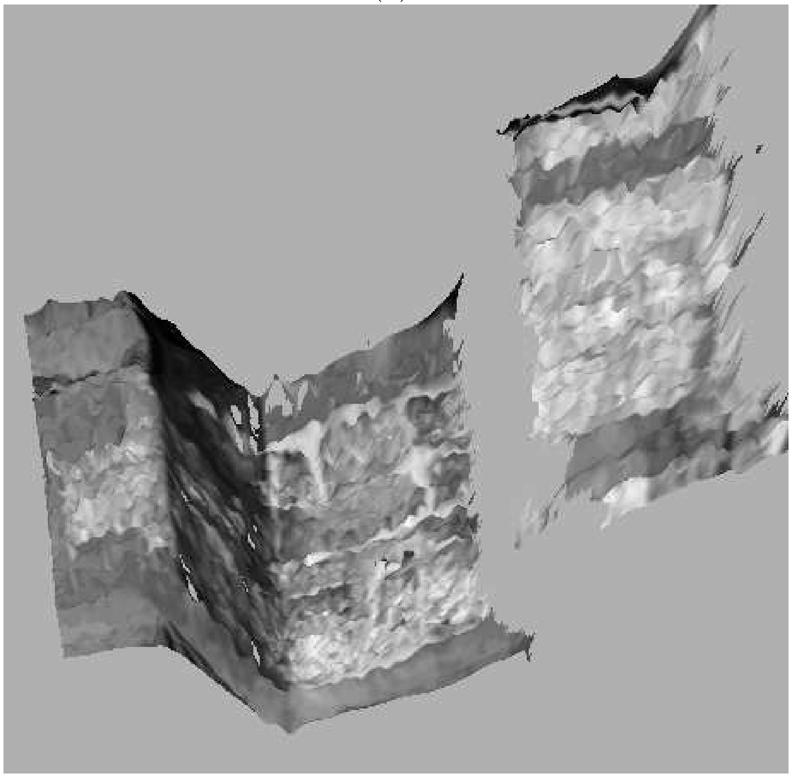


Figure 4-29: *Detail of church wall: Three input views and resulting depth map. (Sony Hi8 camcorder, FOV* 40^o*)*



(a)

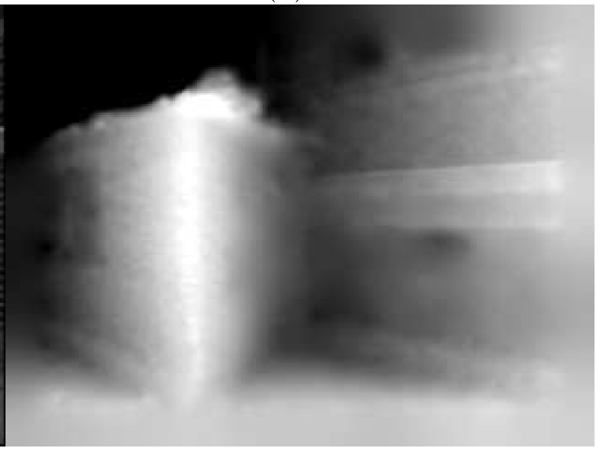


(b)

Figure 4-30: *3D Euclidean renderings of the depth map in Figure 4-29d.*

(a) (b)

(c) (d)

Figure 4-31: *Three input views of a side entrance to a building and resulting depth map. Camera was mounted on a tripod but motion included up to* 0.5^o *rotation.*

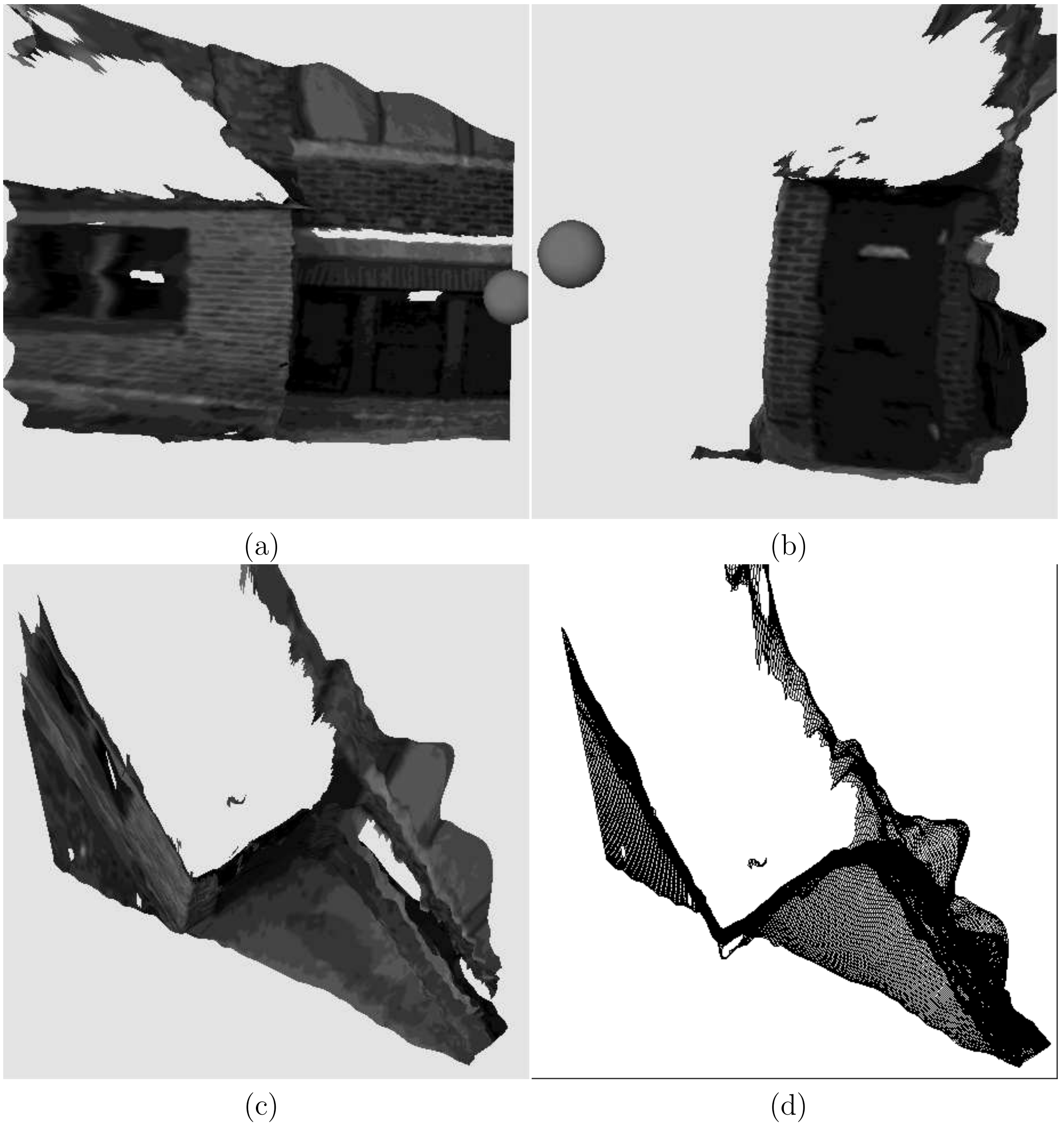


Figure 4-32: *3D renderings of the depth map in Figure 4-31d. The spheres in (a) and (b) indicate the estimated camera location. Note in overhead view (d) true Euclidean structure is recovered with correct* 90^o *angles.*