Chapter 5

Analysis and Results

This chapter provides an analysis of the performance of the DDM for semi-dense 3D scene reconstruction. The DDM has been evaluated using a number of synthetic and real test sequences, and the results of these are presented throughout this chapter. Results are given in terms of the model's tracking accuracy as the DDM deforms throughout a sequence, the accuracy of the resulting reconstructions and the performance gains when prior reconstructions are refined using the model-based energy.

5.1 Test collections

Both synthetic and real video sequences offer advantages for evaluating computer vision algorithms. Although synthetic scenes tend to simplify the additional problems associated with real image data (such as noise, uncontrollable lighting conditions and camera jitter) they can provide easily attainable ground truth data, and therefore are very useful for quantitative evaluation. Ground truths have been especially valuable for calibrating the various parameters of the DDM, which has lead to synthetic scenes being used extensively throughout its development. It is hard to realistically model factors such as noise and illumination variation in synthetic scenes, although these are necessary to evaluate the robustness of the algorithm. The common approach taken in this work is to model noise by corrupting image sequences with additive zero-mean Gaussian noise.

It is often useful to use standard benchmark sequences to evaluate the performance of a reconstruction algorithm. This is to allow for direct and unbiased performance comparisons to be made. Unfortunately, there is very little in the way of benchmark sequences for dense

multiple-frame SfM. There are three reasons for this. The primary reason is because very few algorithms have been developed which can track dense sets of points throughout video sequences. As outlined in Chapter 1, the division in SfM is very apparent; either a sparse set of points are tracked through a video sequence, or dense optic flow is computed only for image pairs or triples. The second reason for SfM benchmark sequences being limited in general is because of the vast number of approaches and assumptions made about the problem domain (e.g. static vs. dynamic scenes, known vs. unknown camera parameters, dense vs. sparse reconstruction, types of features tracked, variable vs. fixed lighting conditions, scene complexity and so on.) It is therefore difficult to construct test sequences applicable for all approaches. The third reason is because the computation of ground truth data for every pixel in every frame can result in large and unmanageable data files. One of the largest resources for computer vision test sequences can be found at CMU's 'Computer Vision Homepage' [13], although there were no sequences applicable for evaluating the DDM.

In light of this, many of the synthetic test sequences were constructed by hand using a ray tracker (POV-Ray). For each frame, the scene is rendered and the camera parameters outputted to a file. Most ray tracers including POV-Ray have no function for outputting ground truth data. A sub-ray tracer was therefore implemented which for every pixel in each frame would output the 3D scene coordinates associated with that pixel. Although this results in large data files of several megabytes for each frame, the alternative approach of computing the ground truths on-demand using the scene's geometric primitives is very impractical. From the ground truth data, the true image motion can be computed, which is used to evaluate the tracking performance of the DDM as it deforms throughout the sequence.

Ultimately, computer vision algorithms operate on real-world data, and so should also be evaluated using real-world test sequences. One of the current limitations of reconstruction using the DDM is that well-estimated camera parameters are required. This presents no problems for generating synthetic test sequences, although for real sequences recording accurately the six extrinsic camera parameters is a cumbersome process. The camera must be mounted on a rig or a robotic arm and this can severely restrict the range of possible test scenes. The real test sequences have been constructed using a standard quality 320×240 pixel webcam. An RTX Robot was used to grip the webcam. This then executed a pre-programmed trajectory whilst the webcam captured the image sequence. It would have been much more preferable to incorporate a camera self-calibration module into the reconstruction algorithm, which would have allowed

the camera to move freely by hand as it captured the sequence. This addition is discussed in Chapter 6.

5.2 Model Deformation

5.2.1 Internal energy weight calibration

The tracking accuracy of the DDM as it deforms to fit new image data is strongly influenced by the choice of internal energy weight w_{int}. This reflects a trade-off in the model's rigidity vs. elasticity. Too high and the allowable deformation is inhibited by its internal rigidity. Too low and the model deforms too freely, which may lead to the degeneration of its structure throughout the sequence. The task is therefore to determine a weighting for which the model will reliably deform, given any sequence. In one sense, emphasis on internal energy should reflect the degree of uncertainty in the image-based evidence. The intuition is that for a node whose image-based motion is poorly determined (i.e. at regions of low texture) more emphasis should be placed on maintaining its geometric relations with neighbouring nodes. Experiments were conducted to investigate this strategy by attributing a lower internal energy weight to nodes with high intensity variation. However, the results were in general no better than when using a static weight, and deformation stability in many cases. The reason for this is because when the internal energy is static, nodes at regions of low intensity variation are already more influenced by their internal energy. This is because their SSD scores within a local region are all similar, causing the image energies to be more uniform, which results in the internal energy dominating their combined energy.

The tracking performance when using various values of w_{int} is illustrated using the sequence shown in Figure 47. This sequence has been captured by a camera zooming towards a textured planar surface. The particular texture used has been chosen as it exhibits both areas of low and high texture. For the purposes of this discussion, the internal energy comprises only local spring energies (i.e. $w_{loc} = 1$, $w_{glob} = 0$). The rigidity vs. elasticity trade-off is not affected by this because w_{loc} and w_{glob} are complements of one another (equations 3.4 and 3.5).

Figure 47: Sequence of a camera zooming towards a textured surface (left) 1st, (middle) 5th and (right) 16th frames. The resolution is 250×250 pixels. Gaussian noise has been added with zero mean and variance 0.002

Figure 48 shows a uniform DDM initialised over the textured surface using the first image of the sequence. The DDM is then deformed throughout each frame of the sequence using a variety of values for w_{int}. The results of these are compared in Figure 49.

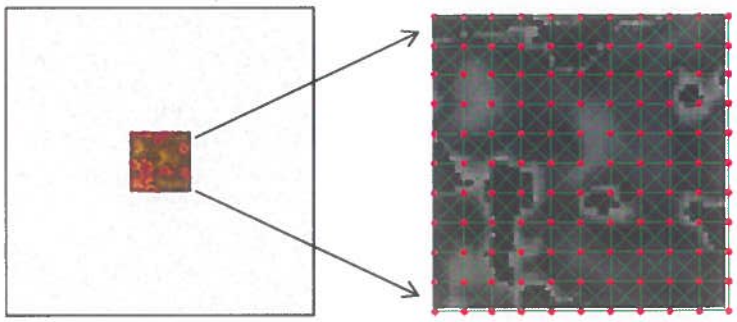


Figure 48: Initialised DDM using the first frame of a sequence. Red points indicate nodes and green lines indicate the mesh-like local spring networks. The separation between nodes is 7 pixels. The expanded view of the texture is in greyscale to improve clarity.

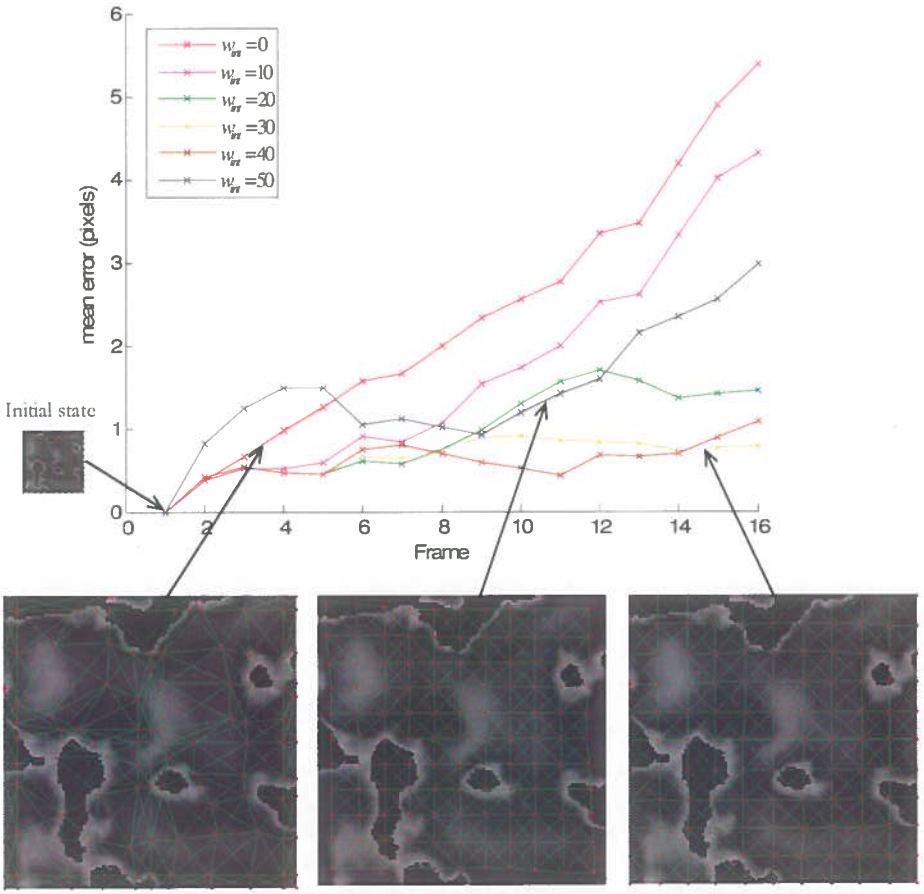


Figure 49: Rigidity/elasticity trade-off using various internal energy weights. (Top) plot of the mean errors of node positions as each frame is processed. Images at the bottom show states of the models after the 16th frame using $w_{int} = 0$ (left), $w_{int} = 50$ (middle) and $w_{int} = 30$ (right)

A weighting $w_{int} = 0$ is the case when only SSD energy us used. The benefit in using the DDM over pure intensity-based matching is very evident. This is clear by contrasting both the error rates over the sequence, and by inspecting the states of the DDM at the bottom of Figure 49. We also observe that for $w_{int} = 50$, the model is too tightly constrained because the internal energy weight is too high (Figure 49 bottom centre). This results in an under-deformed model. In this simple scene, a suitable internal energy weighting is within the range 20 to 40. For a weight of 30, the model's error is consistently under 1 pixel, which is close to the upper limit on

accuracy without using sub-pixel matching. The fact that this is a constant error rate also indicates that no drift is occurring. It was important to perform similar tests over a range of sequences in which different surfaces, textures, camera velocities and node densities were used (some of which are shown in the next few sections). This is because the optimal internal energy may be scene specific. A calibrated internal energy weight of $w_{int} = 20$ was found by taking the intersection of the best performing w_{int} values for each sequence.

It is worth point out that these models were deformed using the node prediction method discussed in section 3.5.3. The benefits of this are typically in a marked reduction of the hill climbing process at each frame, because the predicted positions of the nodes at this next frame are in general closer their true positions. Figure 50 shows an example of this reduction using the same image sequence described above using $w_{int} = 30$. The height of each spike in the two graphs indicates the mean Euclidean distance between the nodes' initial positions at a frame and the true positions. Clearly, a lower height indicates a more desirable initial configuration. The result of using node prediction is both a reduction in these spikes and a lower error after each optimisation terminates.

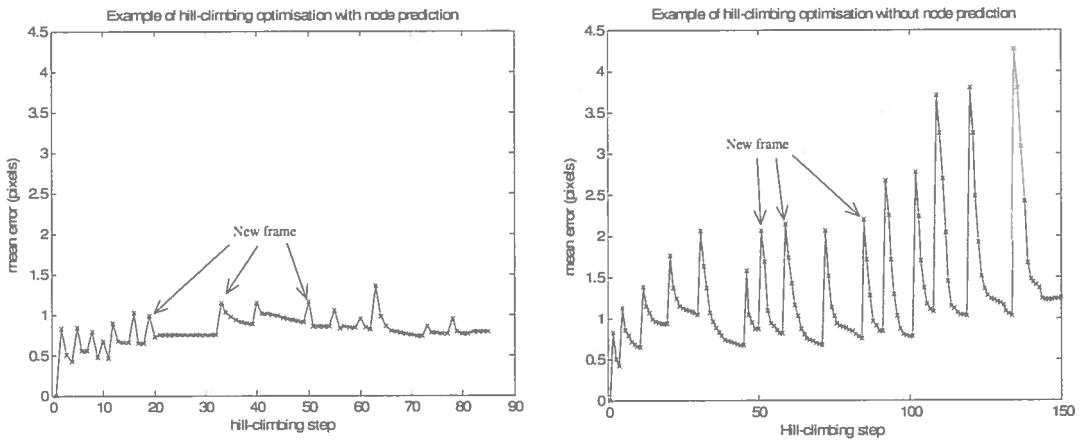


Figure 50: Hill-climbing process made easier by using node prediction. Spikes in mean error indicate the point where a new frame is presented to the model. The subsequent error decay indicates the model deforming to fit the new image data

It is also worth considering the state of the DDM as it deforms throughout the sequence using its spatiotemporal volume representation. This is shown in Figure 51.

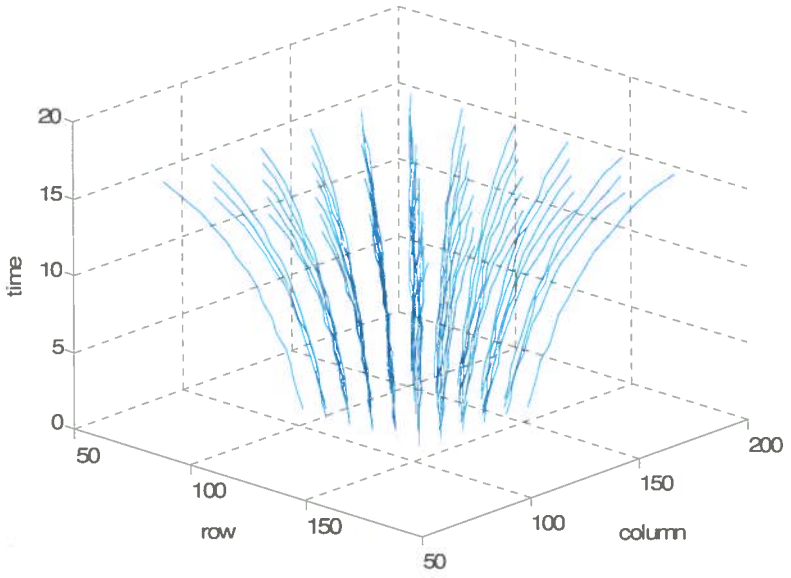


Figure 51: Spatiotemporal representation of DDM using a simple zooming sequence

It is clear that for this simple scene, the DDM has deformed to maintain coherent, non-intersecting paths throughout the spatiotemporal volume.

5.2.2 Performance gain using global spring energies

Recall that the internal energy of a node is comprised of both local and global spring network energies, weighted according to w_{glob} (equation 3.14). In this section a demonstration is presented of the gain in performance when incorporating global spring energy. This scene is again simple, although the results generalise to any scene which has expansive areas of very low texture, which is the case in many man-made scenes. In this example, a camera is translating above a coloured, but completely textureless surface. Figure 52 shows two frames from this sequence. Although simple, this type of scene would present great problems for area or gradient-based optic flow methods. Shown at the bottom of Figure 52 are the states of a DDM at the first and fourth frames in the sequence. This DDM is deformed without using global spring energy.

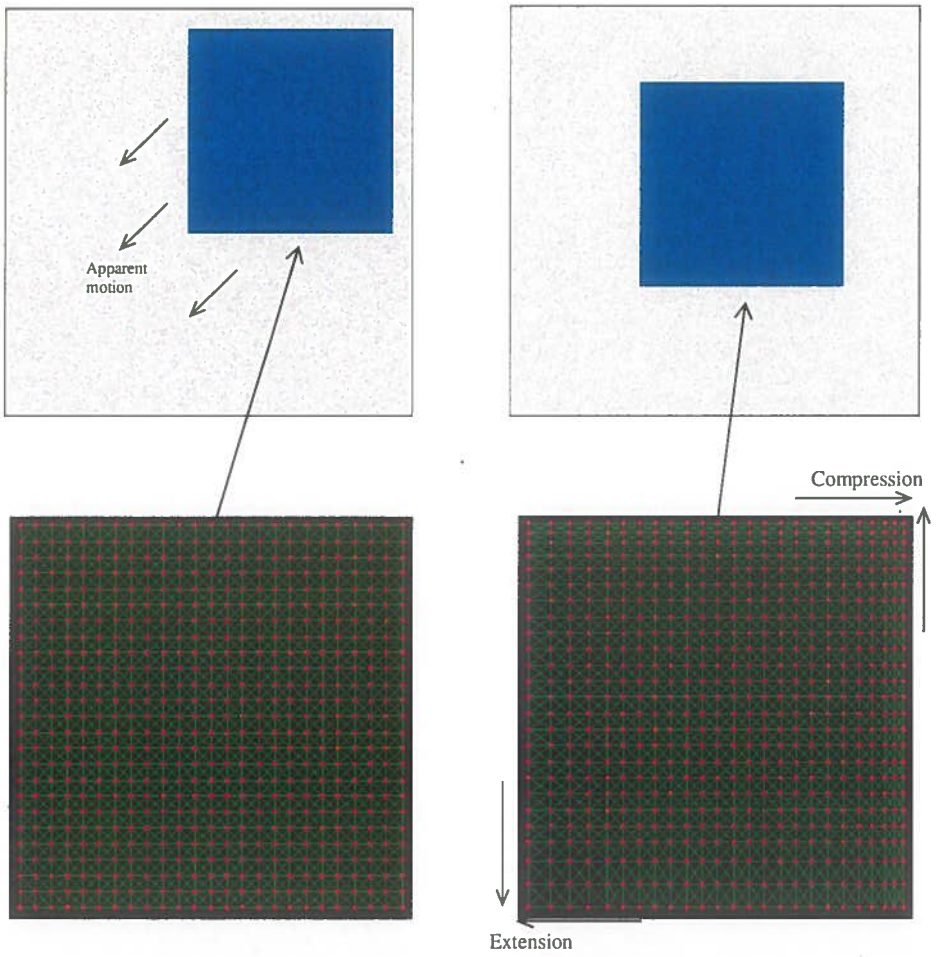


Figure 52: Frame 1 (top-left) and frame 4 (top-right) of a sequence in which a camera is moving above a blue surface of zero texture. The images are 250×250 pixels with added Gaussian noise. The states of the DDM at these frames are shown in the two images below

The model has successfully deformed at the edges and corners of the region, and the structure is preserved throughout the area. However, the spring forces at the edges and corners have failed to propagate throughout the model. This results in compressed springs in the top right corner and extended springs in the bottom left. The effect is even more apparent if we view the error surface over the DDM (Figure 53):

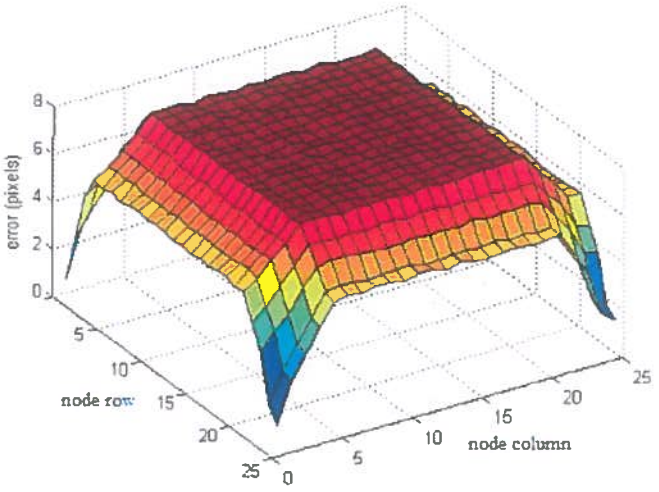


Figure 53: Example error Surface of DDM when using only local neighbours

As suggested by the extension and compression within the DDM, the errors within the centre of the DDM are considerably higher than at the corners and the edges. By contrast, the state of the DDM when using global spring energy weighted by $w_{glob} = 0.5$ after frame 4 is shown in Figure 54:

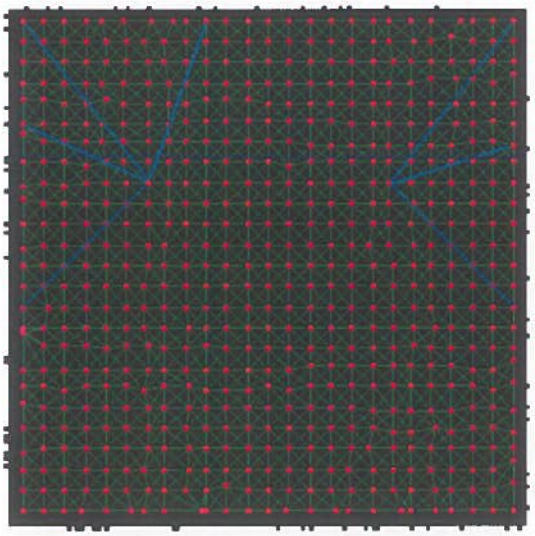


Figure 54: Improved deformation by introducing global spring energy. The global neighbours of two nodes are indicated by the blue lines. These have been selected based on the suitability function given in chapter 3

The resulting state of the model is much closer to the target state (which is where nodes are uniformly spaced). This is because many of the nodes have connections with global neighbours that have high image structure (i.e. those at the edges and corners). As a result, motion has propagated much more readily throughout the model. Given that there are only four regions in this scene where motion is completely resolvable (at the corners), the DDM has done well in disambiguating image motion throughout the textureless region. One of the repercussions for setting w_{glob} is for the local structure of the model to degrade, which is noticeable at several points in Figure 54. To preserve local structure, w_{glob} must be set accordingly. A good strategy for achieving this is to incrementally reduce w_{glob} to zero with each optimisation iteration. In particular, good results are achieved using:

$$w(i+1)_{glob} = w(i)_{glob} - 0.05 \times (i+1) \tag{5.1}$$

where i is the optimisation iteration number and $w(0)_{glob} = 0.5$. This enables motion to easily propagate throughout the model in earlier rounds, but in later rounds the emphasis is on maintaining local structure. Using this strategy, the model's state throughout this sequence is very close to perfect.

5.2.3 Within-plane rotation invariance

The within-plane rotation invariance of the DDM is now presented using a scene in which a camera rotates about its optical axis, which is directed towards a planar surface. Four example frames from this sequence are shown in Figure 55.

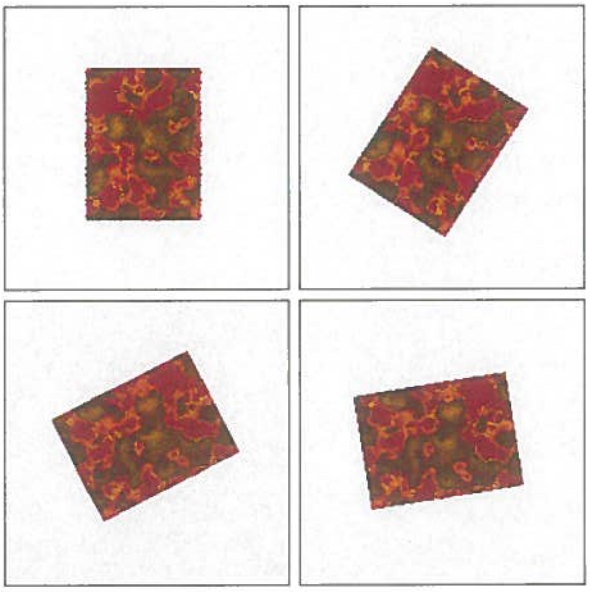


Figure 55: Frames from in image sequence used to demonstrate the within-plane rotation invariance of the DDM. The camera rotates about its optical axis, which is directed towards a planar surface. Frames 1 (top-left), 10 (top-right), 20 (bottom-left) 30 (bottom-right) are shown.

The DDM is initialised using the first frame and deform throughout the sequence. The model's structure in the 1st frame and 30th frame (after 30 deformations) is shown in Figure 56

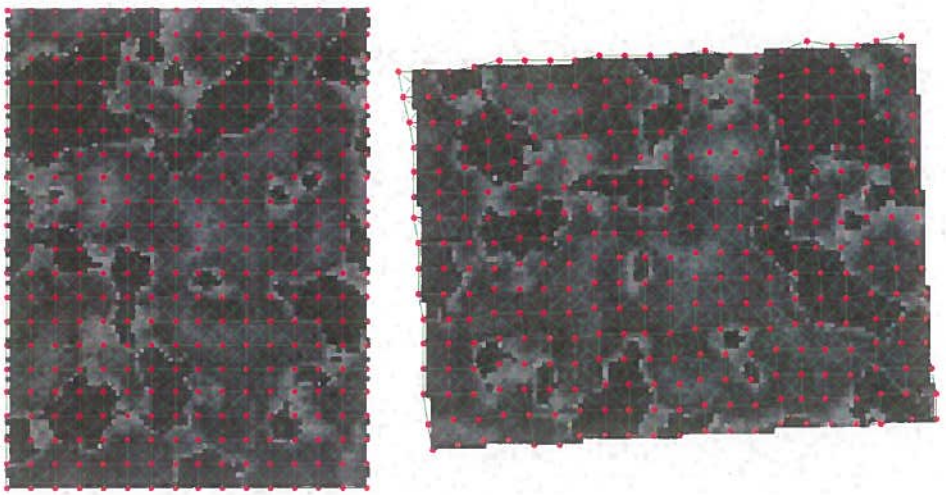


Figure 56: Within-plane rotation invariance of DDM. (Left) initial state. (Right) state after 30 frames

We can see that the DDM has deformed very successfully throughout the sequence. It has also performed well in the face of considerable pixel quantisation noise. A great strength of the DDM is that similar results are also attained when the surface has little or no texture (Figure 57)

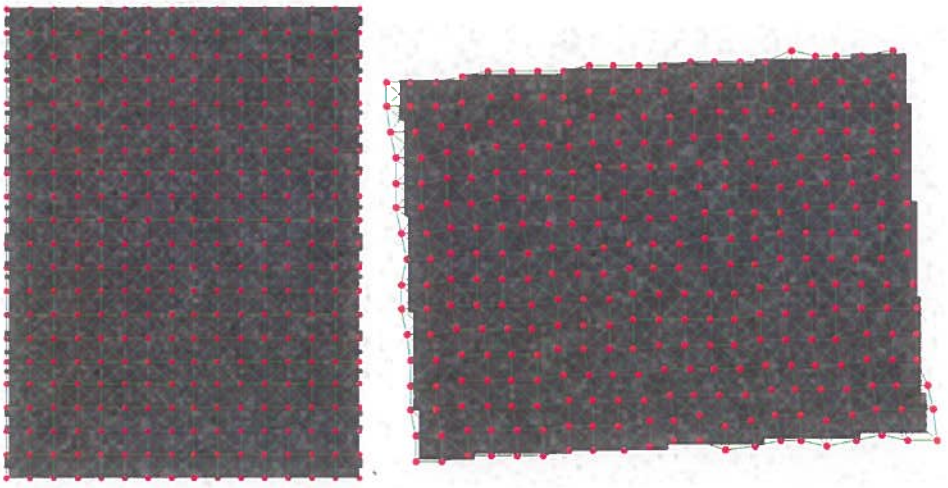


Figure 57: Within-plane rotation invariance of DDM for areas of very low texture

There is slight loss of accuracy at three corners, which is almost certainly a product of the relatively high quantisation noise. The spatiotemporal representation of the DDM is shown in

Figure 58, which shows the DDM has performed well in maintaining the relationships between nodes throughout the volume.

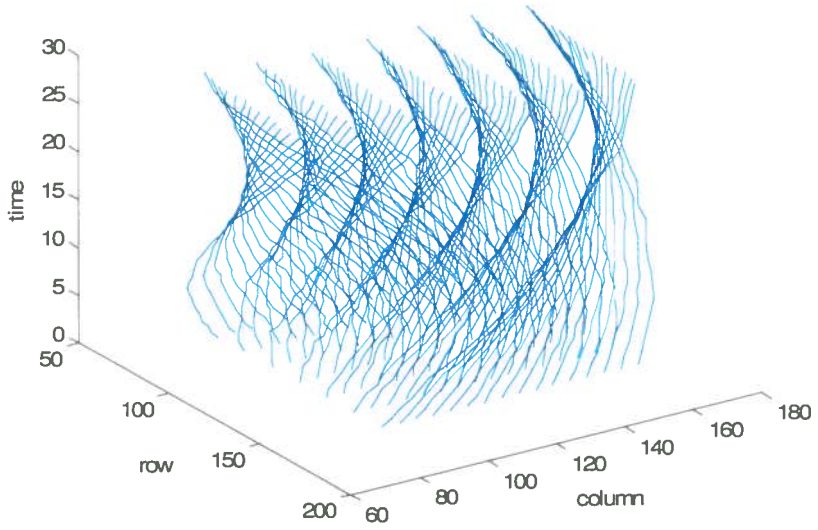


Figure 58: Spatiotemporal representation of a DDM where a camera rotates about its optical axis. Nodes have been under-sampled to clarify the visualisation

5.2.4 Out-of-plane rotation invariance

In the examples so far presented, motion between the scene and camera has been orthogonal to the optic axis. An example is now presented where this is not the case. In this scene, the camera is moving in an arc around a planar surface. Four frames from this sequence are shown in Figure 59.

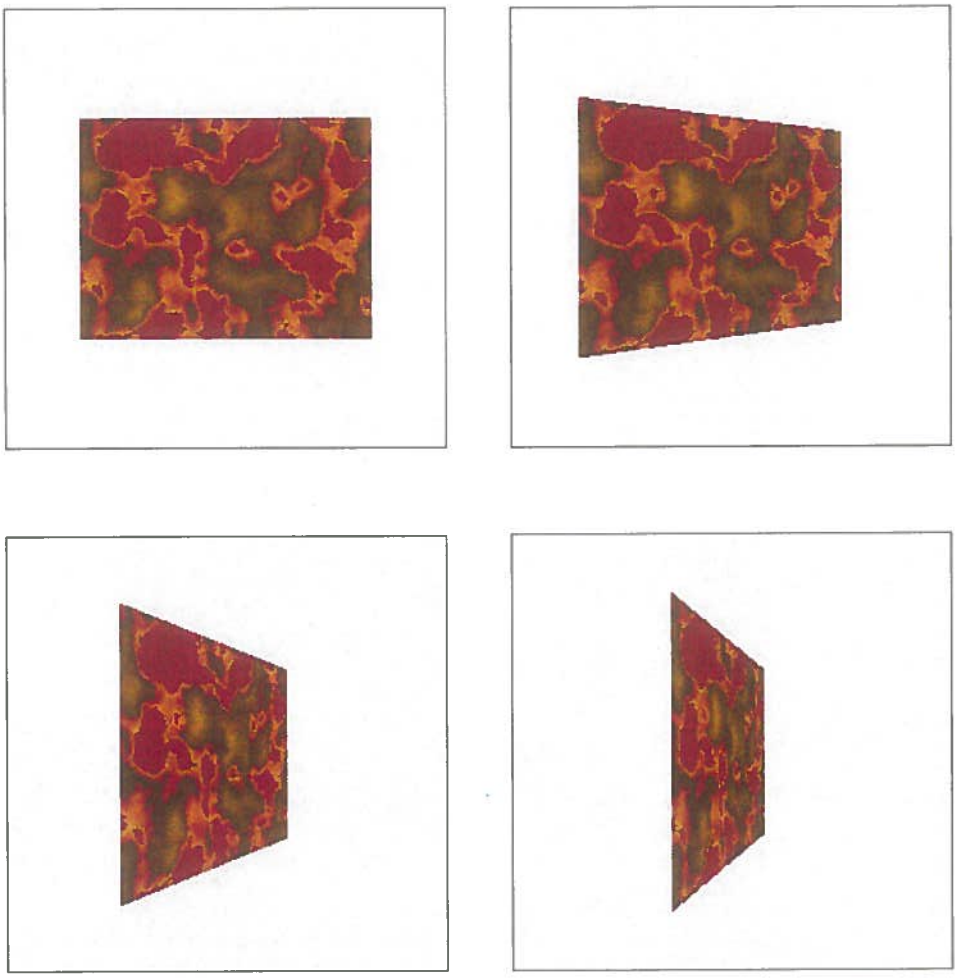


Figure 59: Example frames from the a sequence of a camera moving in an arc around a planar surface. Frames shown are (top-left 1, (top-right) 20, (bottom-left) 40,(bottom-right) 50. Frames are 250×250.

The reason why this sequence will be problematic is for the same reason which drives the motivation for the model-based energy term. The spring energies (and indeed any other image-based regularisation) work to maintain the structure of the model in 2D image space rather than in scene space. The results of deforming a DDM using this sequence are shown in Figure 60.

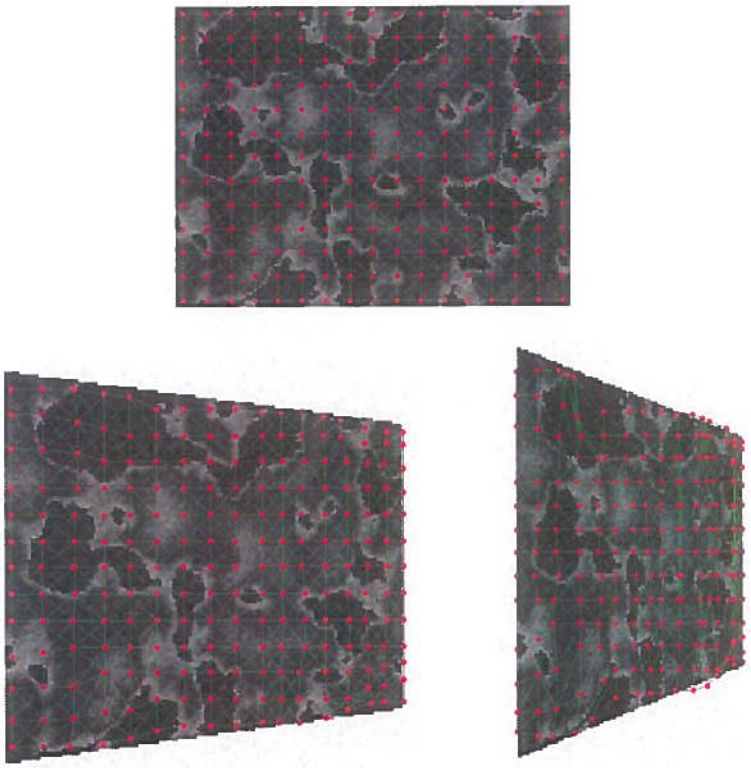


Figure 60: Images of the sate of the DDM at frame 1 (top), frame 20 (left) and fame 40 (right)

It is clear that the DDM has attempted to both fit the data but whilst trying to maintain its projected 2D structure in the first frame. The solution to this, outlined in section 3.2.2 is to have temporally discounted spring constraints. The discount factor λ is used which varies from 0, where only the first frame's spring state is used, to 1, where only the previous frame's spring state is used. The inherent trade-off in choosing λ is illustrated (using the sequence above) in Figure 61, where all other factors have been kept equal. When $\lambda = 0$ only the state of the DDM in the previous frame is used. This corresponds with a frame-pair approach for tracking throughout the video sequence, and as shown in Figure 61 results in the worst performance. It is clear that by incorporating spatiotemporal constraints, tracking performance is improved. On the other hand, when $\lambda = 1$ the result is again sub-optimal because the DMM is being constrained to preserve its 2D structure in the first frame. A good choice for λ in this scene is

approximately 0.8, which means that after 5 frames, the influence of the spring configurations in the initial frame has reduced from 1.0 to $(0.8)^5 = 0.33$.

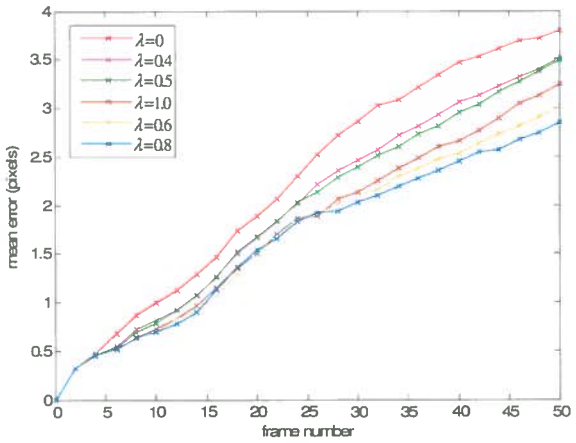


Figure 61: Performance trade-off regarding the proportion of past DMM spring state used to constrain next-frame deformations

It is worth visualising the improved deformation when using $\lambda = 0.8$ contrasted with $\lambda = 0$ (the frame-pair approach). The states of the two DDMs after frame 40 using these discount values are shown in Figure 62.

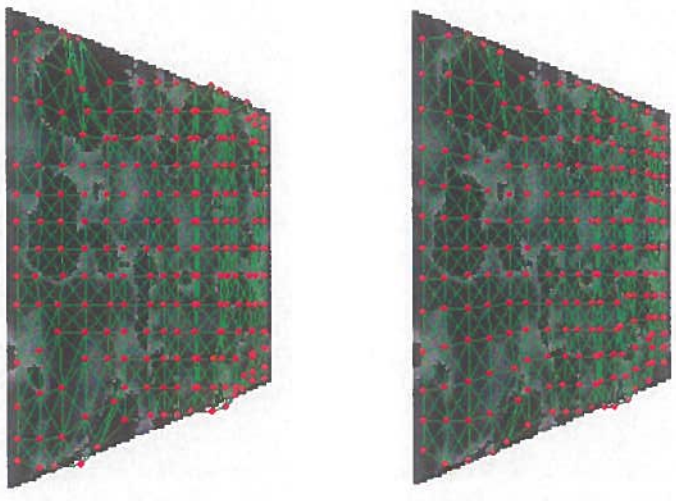


Figure 62: Improvement in tracking performance when incorporating historic spring constraints after 40 frames. (Left) $\lambda = 0.0$, (Right) $\lambda = 0.8$

Whilst $\lambda = 0.8$ is a good discount rate in this scene, it is difficult to guarantee an optimal choice of λ for points in an arbitrary scene. For local affine transformations, a smaller discount will give better results, but for projective transformations a higher discount is preferable.

Results are now presented for this sequence using the model-based energy. This is to demonstrate the potential performance gain once the model-based energy is incorporated into the DDM. The following sequence of images shows the deformations using this model-based energy, where a correct surface model is assumed (i.e. the surface normal of the textured plane is known).

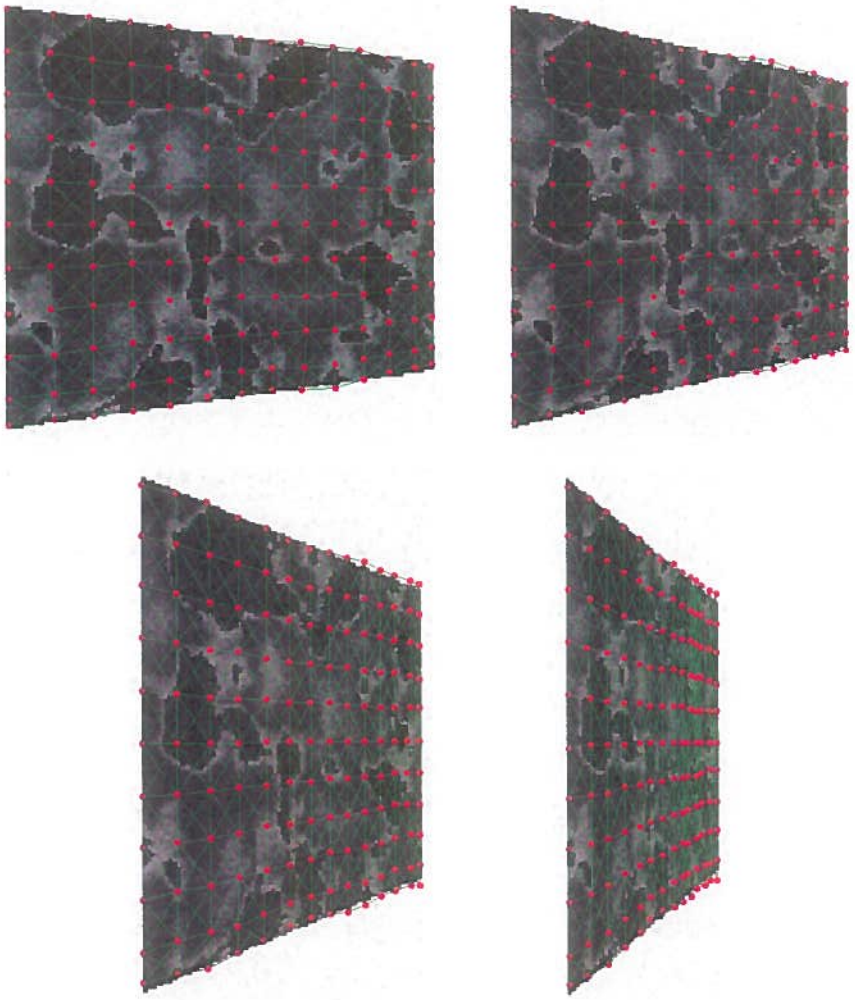


Figure 63: Performance gain using the model-based energy

It is clear that by using the model-based energy, excellent results are attainable for projective transformations. Furthermore, these results are not limited to textured surfaces. Figure 64 shows similar results in the final frame for a surface which has almost no texture.

Figure 64: Model-based energy for accurate deformations where there is very little

The slight loss of accuracy at the top far corner is a result of the discrete hill-climbing optimisation, which is not sub-pixel accurate. At this frame, the mean Euclidean error of the nodes is 0.62 pixels, which is close to the theoretical upper limit of 0.35 (the expected error in the discrete approximation of a point's projection onto the image.) Even further performance gains could be achieved using a sub-pixel post-deformation. This enhancement is further discussed in Chapter 6.

5.3 Reconstruction of a simple scene

The previous section has proven the worth of the model-based energy. In this section, a full multi-stage reconstruction is presented using a synthetic scene without prior knowledge of the surface normals. This scene comprises three planar surfaces which form a corner. The surfaces

have been textured by varying amounts to reflect the ability of the algorithm to reconstruct both high and low textured surfaces. The test sequence is captured by a camera rotating around the corner, although to make things interesting this is not about the corner point. The following images are four example frames taken from this sequence:

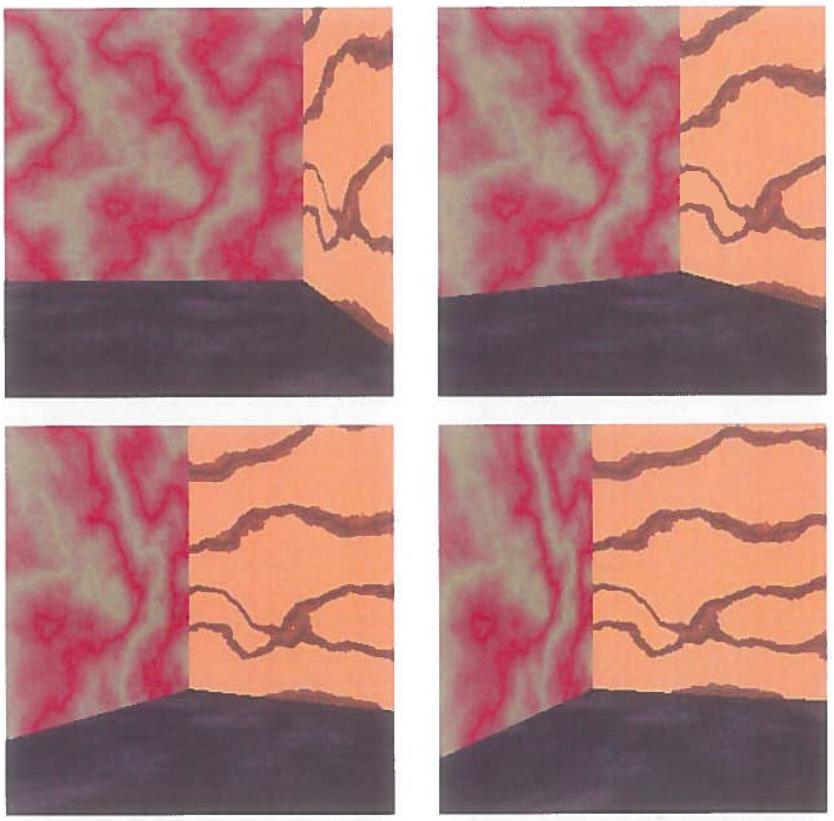
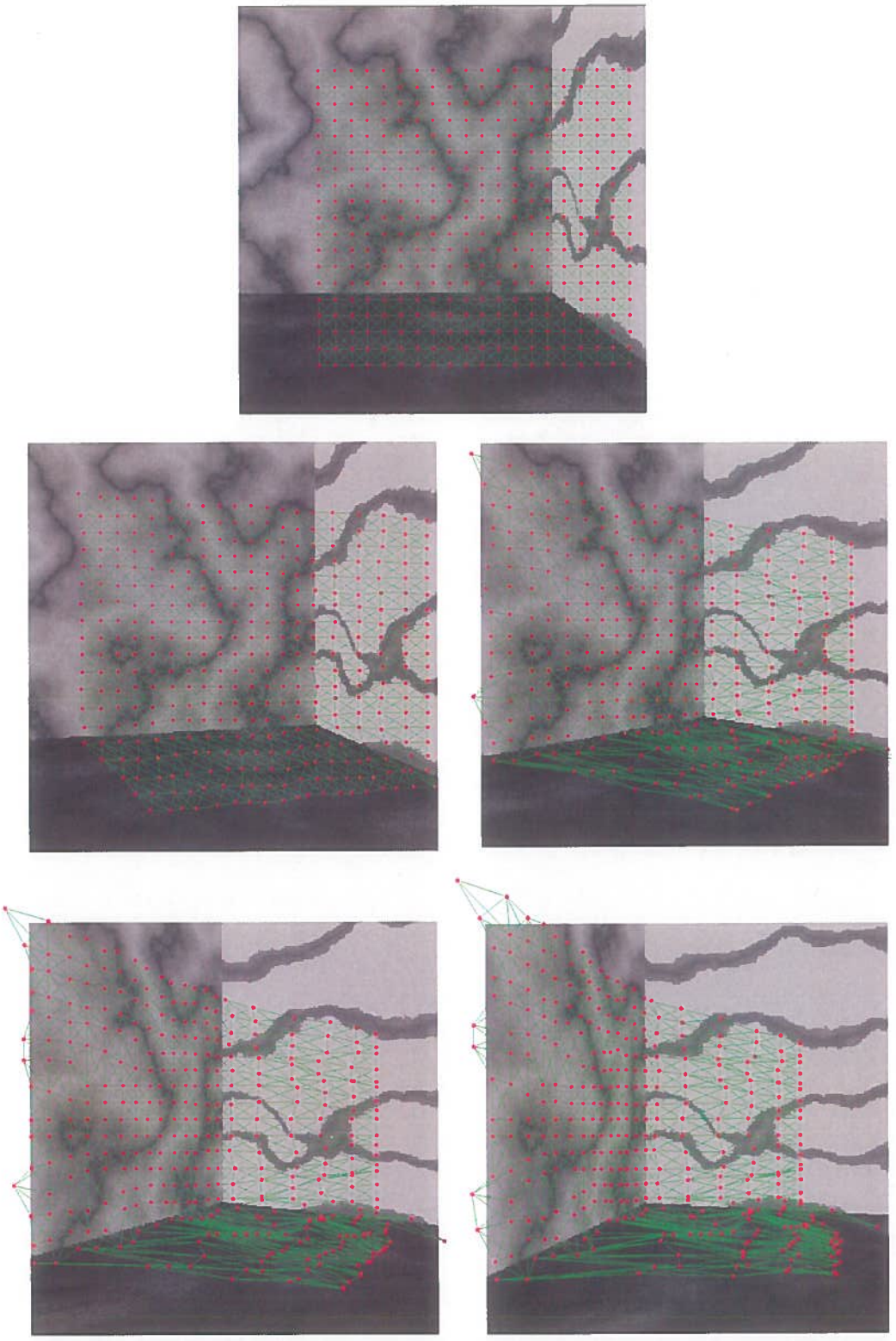


Figure 65: Frames 1, 20, 40 and 50 taken from a synthetic 50-frame sequence of a camera moving around a corner constructed from three textured surfaces. Frames are 250×250

5.3.1 Deformation

The next series of images shows the state of the DDM as it deforms throughout the sequence without using the model-based energy. States at frames 1, 10, 30, 40 and 50 are shown in left-to-right, top-down order.



A uniform DDM has been used in this demonstration because it is easier to interpret the tracking performance from the visualisations, although the algorithm in 3.3.1 for distributing sparse DDMs could equally have been used. Throughout the sequence, the DDM has deformed the most accurately for the two surfaces facing and to the right of the camera in the initial frame. Notice also the ability to maintain nodes which have moved beyond the viewable image plane, although this has been done fairly unreliably. For the surface to the bottom, deformation is less satisfactory as a result of the very low texture.

In addition to these visualisations, it is useful to gauge the mean tracking error as the DDM deforms throughout the sequence. Unless tracking is perfect, there will be a progressive loss in accuracy with each frame in the sequence. This is illustrated in Figure 66, which shows a roughly linear degradation in performance.

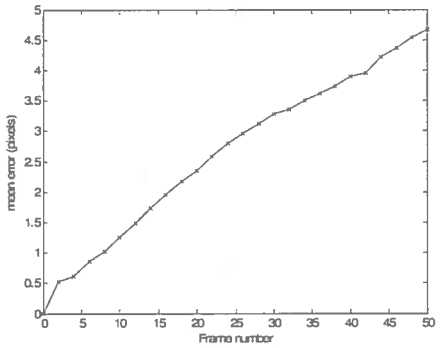


Figure 66: loss of model accuracy throughout the 'corner' video sequence

5.3.2 Reconstruction

We now consider the task of reconstruction using the deformed DDM. Triangulation offers one approach for this, although it raises the interesting problem of deciding which frames to use as the triangulating pair. Further into the future and the baseline may become wider which reduces the depth uncertainty, but the tracking error increases. Clearly, there is an optimal balance between baseline width and tracking error, although without knowing theses error this can be hard to determine [15]. This problem is essentially removed if we use the spatiotemporal curve matching method described in chapter 4, since depth is estimated using the positions of nodes in all the frames rather than a selected pair. A significant finding in this project suggests that in the presence of tracking errors, reconstruction using the spatiotemporal search outperforms

triangulation irrespective of the baseline. The reconstruction of the corner using the deformed DDM and the spatiotemporal search method is shown from two different views in Figure 67 and Figure 68. The optical axis of camera at the first frame is along the positive z-axis.

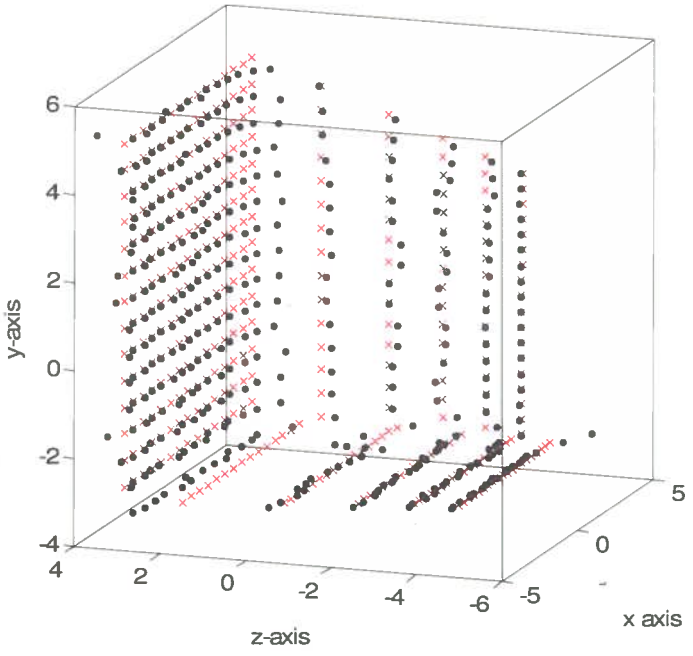


Figure 67: Reconstructed corner using the deformed DDM and the spatiotemporal search reconstruction method. Red crosses indicate the ground truths and black dots indicate the recovered 3D coordinates of the DDM nodes

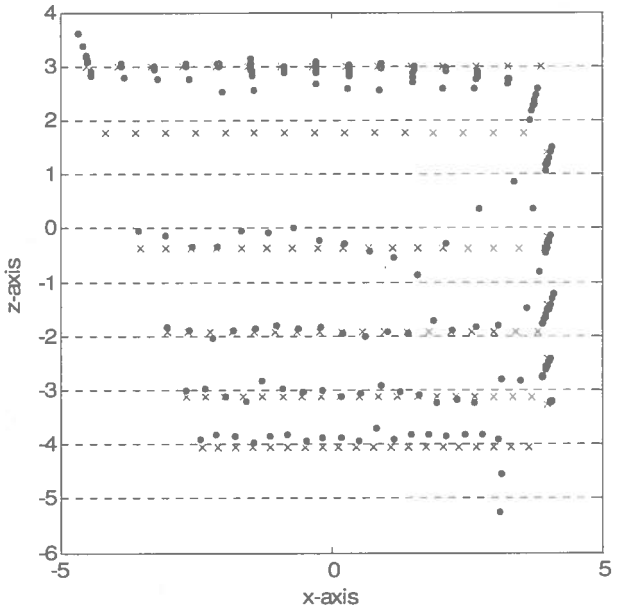


Figure 68: Reconstructed corner using the deformed DDM and the spatiotemporal search reconstruction method (Bird's eye view)

The results show a reasonably faithful reconstruction. The deformation of the DDM over the central and right faces in the first image have lead to the most accurate reconstructed sections of the corner, since these were most reliably tracked throughout the sequence. The outliers at the extremities of the corner are those nodes which moved beyond the viewable image plane during the sequence. Another problem is that nodes close to the edge joining the far and right surfaces in the first frame have been smoothed, and the sharpness of this edge has been lost. The two furthest rows of nodes on the ground surface have been the most poorly reconstructed. The reason why this is the case for the farthest row is because the DDM has tracked these nodes as part of the back facing surface rather than the ground surface. This is clear by looking at these nodes in the deformed DDM images. The problem is a result of the SSD comparison windows encompassing the hard edge intersecting these two surfaces. This strong intensity feature has prevented these nodes from moving away from the edge with the ground surface as the camera rotates around the corner. The second row has been poorly tracked simply because of insufficient data. The mean Euclidean error of the reconstruction in scene space is 0.205 units.

We can contrast this reconstruction with one using stereo triangulation. A view of the reconstruction using the first and last frames as a stereo pair (i.e. the widest baseline) is given in Figure 69.

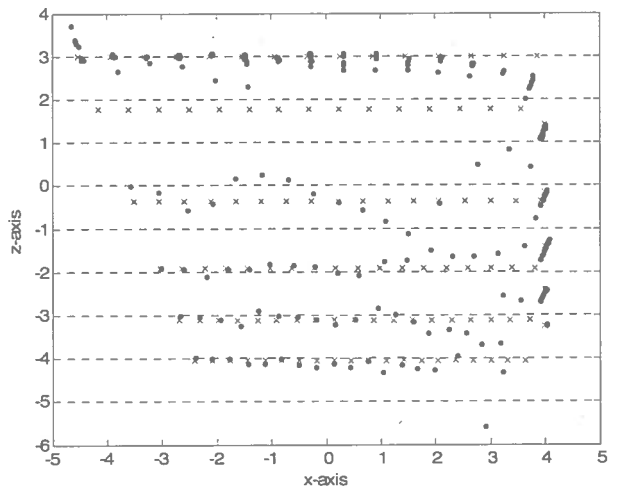


Figure 69: Reconstruction of corner using a wide triangulation baseline (bird's-eye view)

We notice that this reconstruction is in fact poorer than when the spatiotemporal curve matching method was used. This is particularly true of the two rows just discussed. The mean scene-space error is 0.24 units, which is a 22% error increase. It is also insightful to compare the spatiotemporal curve matching method against triangulation using a range of stereo pairs. Figure 70 shows the trend in mean error using the first frame, and each of the succeeding frames as a triangulation pair.

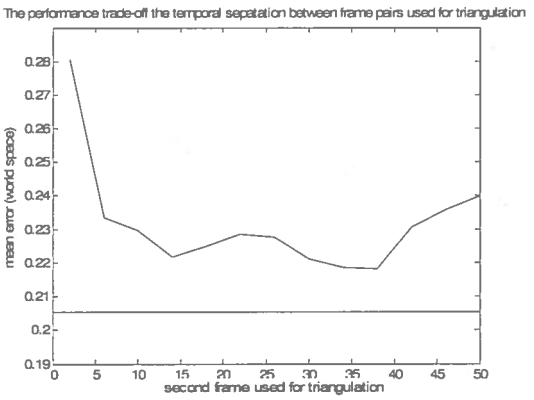


Figure 70: Performance comparison between triangulation and spatiotemporal curve matching reconstruction methods. The blue line is the triangulation error using frames 2,3,…,50 as the second triangulation frame. The black line is the spatiotemporal curve matching method error

Some interesting points can be drawn from this graph. The first is that the increased stereo baseline by using frames which are up to 37 frames apart outweighs the DDM tracking error, although beyond frame 37 the tracking error results in worse reconstructions. This reflects the stereo pair trade-off discussed above. The second point is that irrespective of the baseline, the spatiotemporal curve matching method always outperforms triangulation by a noticeable amount. This is a significant finding, particularly as these two methods have never been compared before. A reasonable hypothesis for this is that spatiotemporal depth curves time-integrate depth information over the entire duration of the sequence rather than only for a single frame pair, which results in more robust depth estimates. It would be very worthwhile to further investigate this finding. This is further discussed in the Future Work section of Chapter 6.

5.3.3 Surface normal estimation

The scene coordinates of the DDM nodes recovered using the spatiotemporal curve matching method can be viewed as a 3D Point Cloud Distribution (PCD) in scene space. This distribution can now be used to estimate the local surface normals of the corner using the PCD least squares plane fitting method described in section 4.3. The results of this are shown in Figure 71 and Figure 72. For each node, the *knn* set used to fit the planes comprises the points of that node and its local spring neighbours.

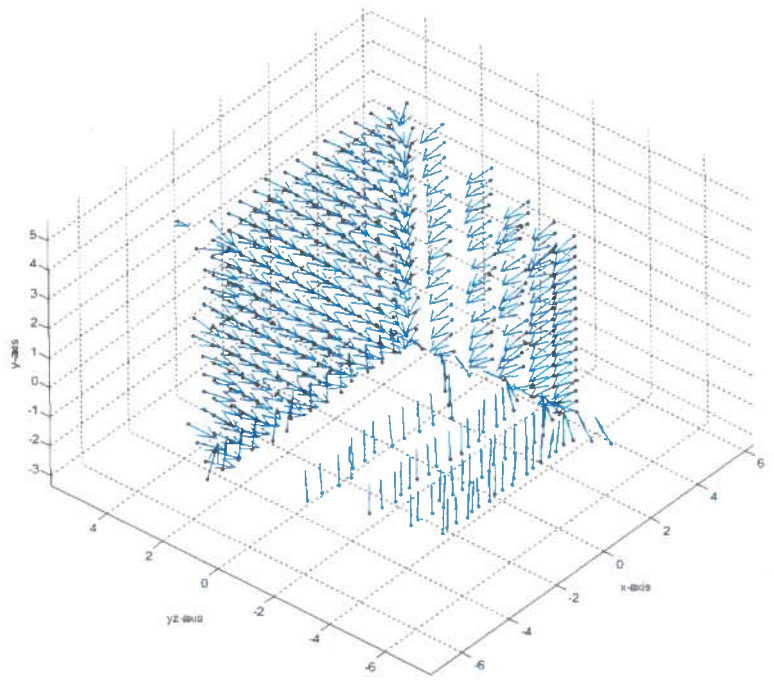


Figure 71: Plot 1 of surface normal estimates after first corner reconstruction

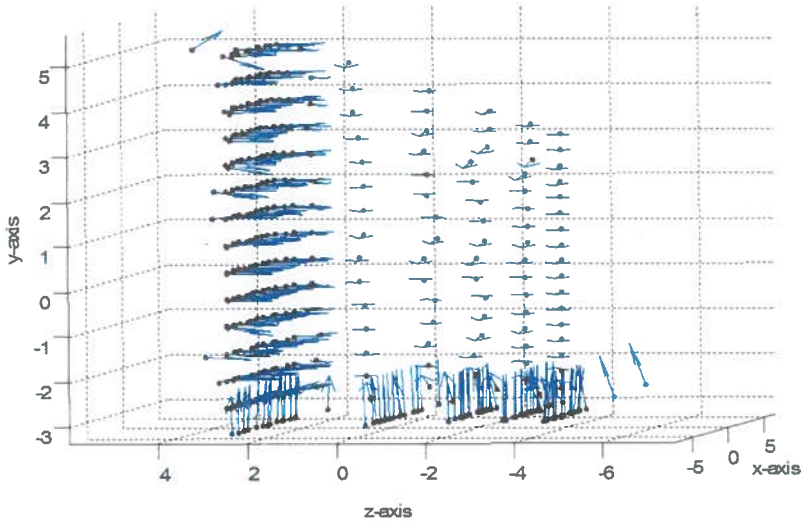
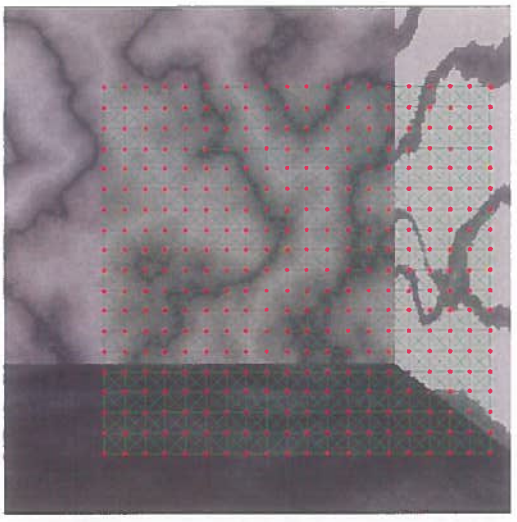


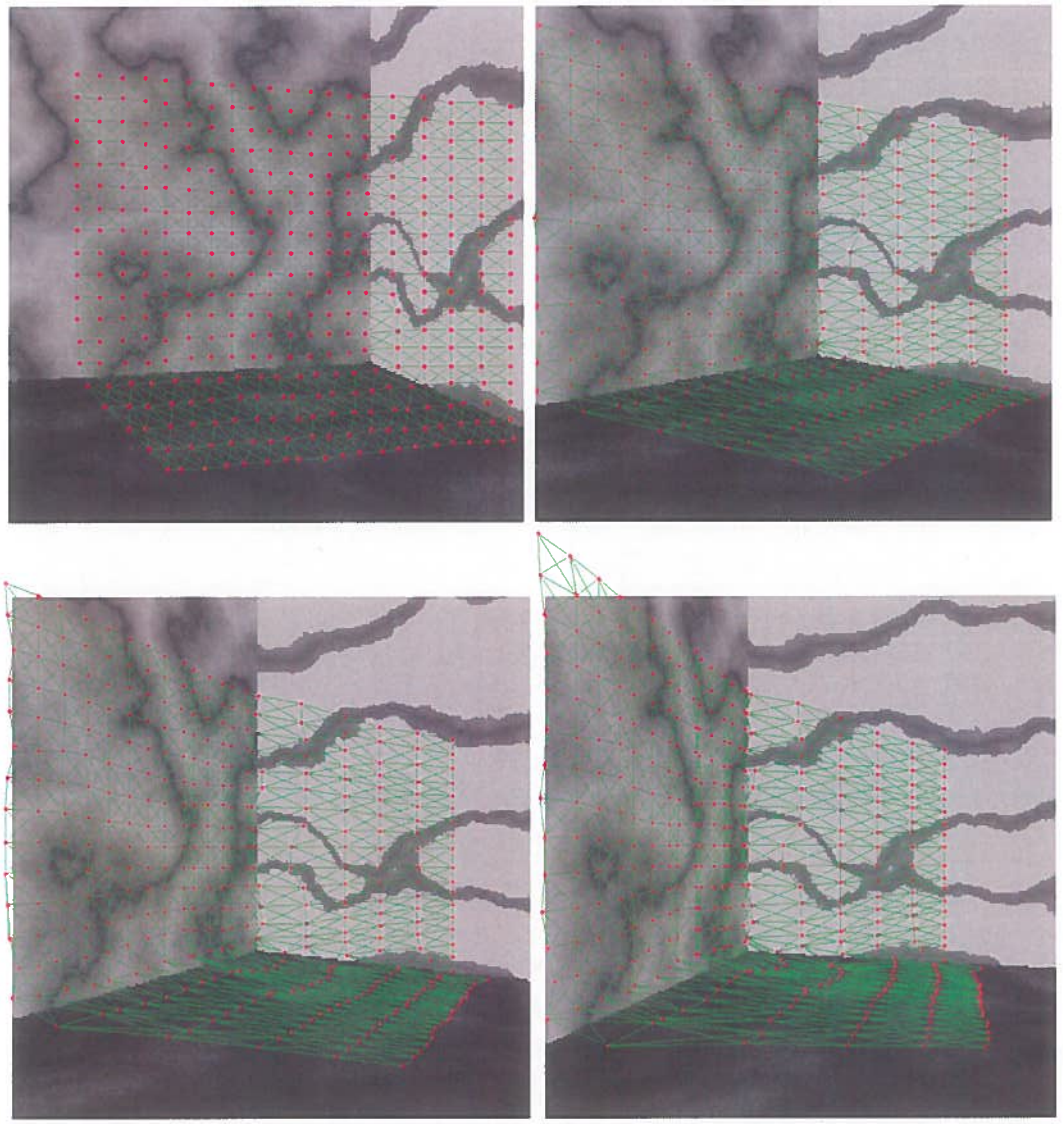
Figure 72: Plot 2 of surface normal estimates after first corner reconstruction

The true normals of the three surfaces are $\mathbf{n}_1 = (0,0,1)^T$, $\mathbf{n}_2 = (0,1,0)^T$ and $\mathbf{n}_3 = (1,0,0)^T$. It is clear that the estimates of these are in general quite reliable, although by inspecting Figure 71 problems have occurred at some of the surface boundaries. This is a result of the planes being fitted to neighbouring points which belong to different surfaces, which causes smoothed normals at these points. Normal estimates for the outlier points are also less accurate. The alternative view in Figure 74 shows that the fitting algorithm has performed very well at the intersection between the far and ground surfaces.

5.3.4 Reconstruction using model-based energies

The model-based energy E_{model} can now be phased into the DDM's energy equation by modifying the model-based/spring-based energy weighting parameter $0 < \alpha < 1$ (equation 3.3). It is important not to completely eliminate E_{spring} after the first reconstruction, since the model's internal constraints would be entirely determined by possibly inaccurate surface normal estimates. If α is set to 0.80 (i.e. we have both model constraints E_{model} and some 2D spatiotemporal smoothness constraints from E_{spring}) and the same DDM is deformed throughout the same sequence, its state after frames 1, 10, 30, 40 and 50. is shown as follows:





The incorporation of the model-based energy into the DDM energy has resulted in a substantial improvement in tracking performance. This is particularly true of the nodes which are on the ground surface. Also, the positions of nodes which have moved beyond the visible plane have been predicted with much greater accuracy. A clear comparison between the states of the DDM after frame 50 is given in Figure 73.

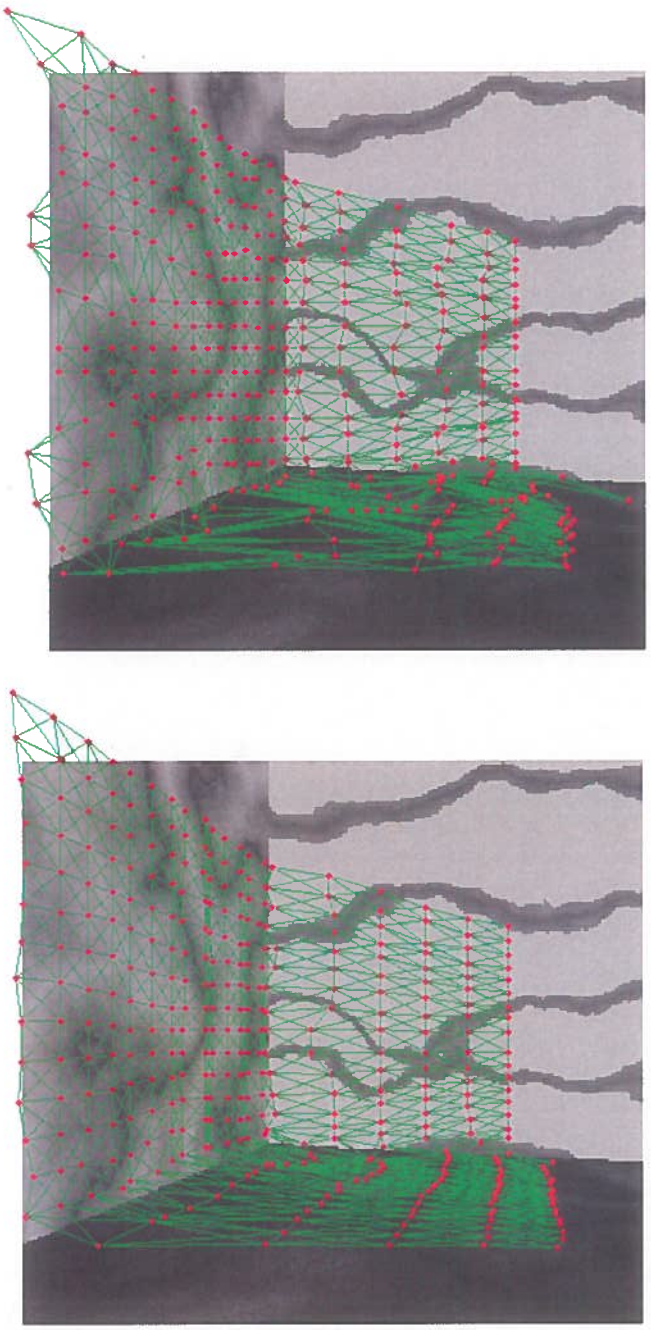


Figure 73: Comparison between the tracking performances using the 'corner test sequence with no model-based energy is used (above) and model-based energy used from first reconstruction (below)

This improvement can be quantified by comparing the tracking errors of the two DDMs as they deform throughout the sequence (Figure 74).

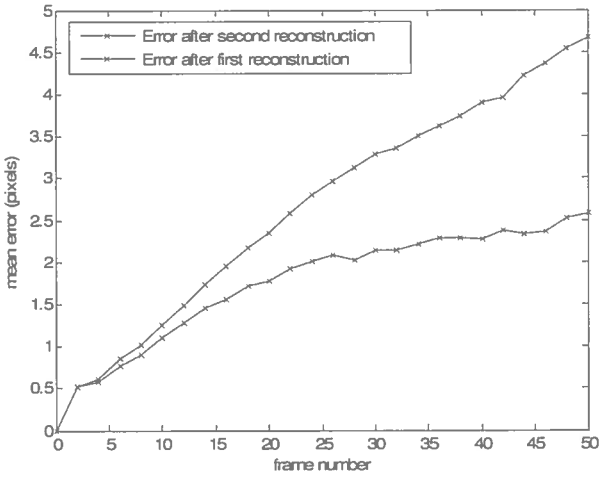


Figure 74 Improvement in deformation performance when using the model-based energy

After the 50th frame, the mean node position error of the new DDM is just over 2.5 pixels, which marks a 45% performance gain over the first DDM. It is also important to notice that the rate of node drift for this new DDM is somewhat less than linear in order. This is very desirable when the DDM is deforming over longer time scales.

We can now reconstruct the corner using this second DDM and contrast the reconstruction quality with the first. This is shown in Figure 75 and Figure 76.

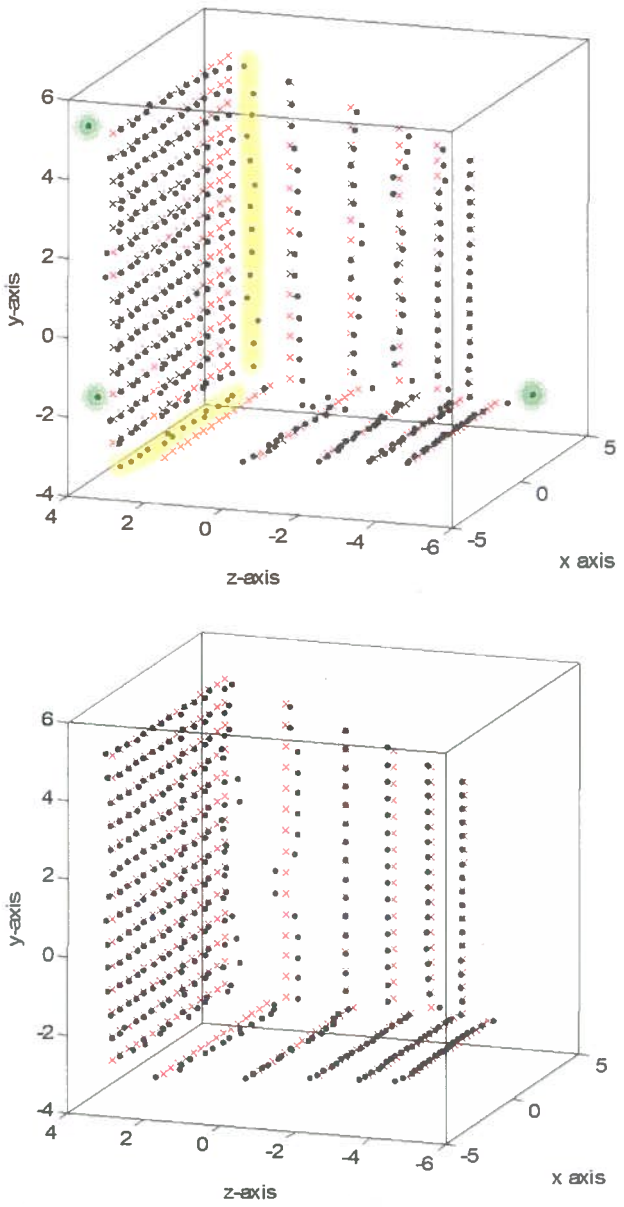


Figure 75: Improvement in reconstruction using DDM with model-based energy (asymmetric view). First reconstruction (top) and second reconstruction (bottom). Points highlighted in green are the outliers which have been resolved in the second reconstruction. Yellow points are error which have resulted from smoothing across surfaces, which have also been improved.

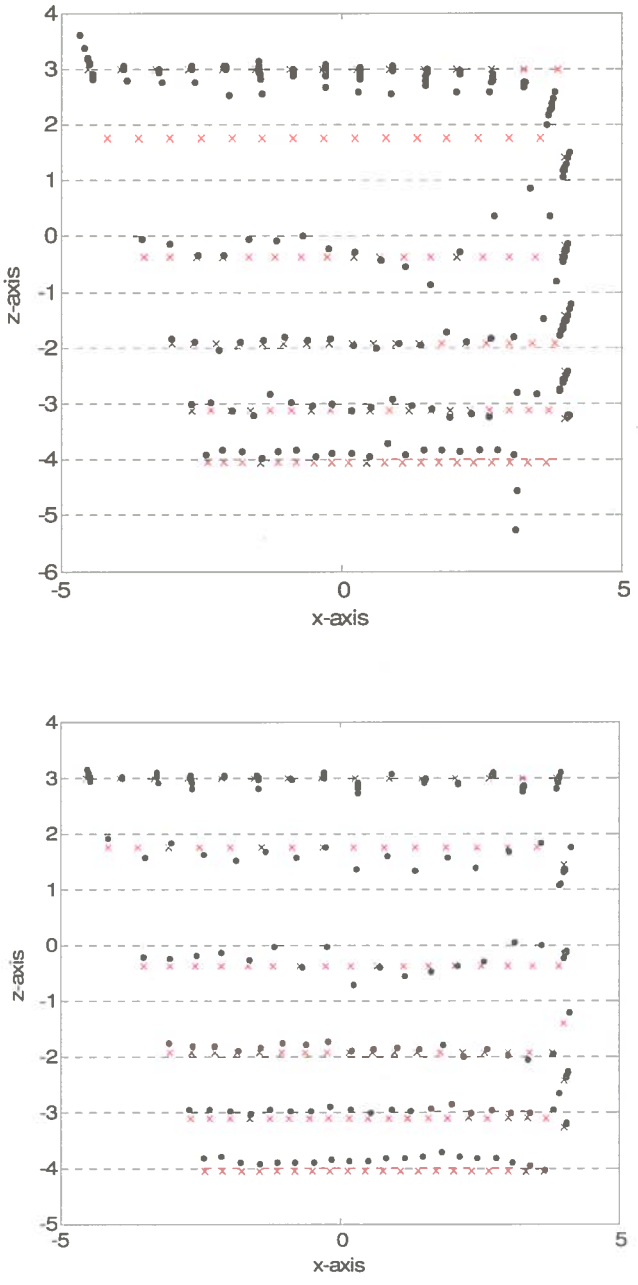


Figure 76: Improvement in reconstruction using DDM with model-based energy (bird's eye view). (Top) first reconstruction. (Bottom) Second reconstruction

These reconstructions show a good improvement when the second DDM using the model-based energy is used. The root mean squared error in scene space has reduced from 0.205 to 0.142, which is a 30.7% increase in accuracy. The points highlighted in Figure 75 show particular problems in the first reconstruction which have been considerably improved in the second. The green points are those which lost accuracy when they fell out of the camera's sight in one or more frames of the sequence. In such frames, the positions of these points are determined entirely by their internal energy. By using the model-based energy, these points are constrained with respect to the plane fitted locally to the surface at those points in the scene. This results in more accurate tracking in the absence of data. The points highlighted in yellow are those which have been incorrectly tracked due to the discontinuities at the edges. However, using the model-based energy, these problems have been resolved.

The cycle of surface normal estimation, deformation and reconstruction can then be repeated until the reconstruction converges. This has been performed for the corner sequence. The plot in Figure 77 shows the reduction in reconstruction error after five reconstruction cycles. It also shows the reduction in the mean angular difference between the true and estimated surface normals.

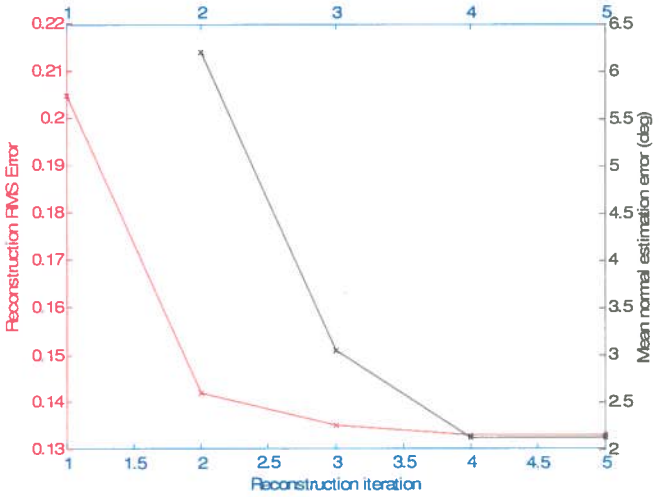


Figure 77: Improvement in reconstruction accuracy of corner after five iterations. Te black plot shows the mean error of the surface normal estimates at the start of each iteration before the DDM is deformed. The red plot shows the reconstruction RMS error after each reconstruction iteration.

The reconstruction converges after four reconstruction iterations with a RMS error of 0.133 units. Between iterations 2 and 4 the surface model is improved, which reflects the drop in mean

normal estimation error. This reduction results in an improved RMS error at the end of these iterations. The surface model, and consequently the RMS error converges after iteration 4. The reconstruction after iteration 5 is shown in Figure 78, and polygonised surface reconstruction using these points in shown in Figure 79.

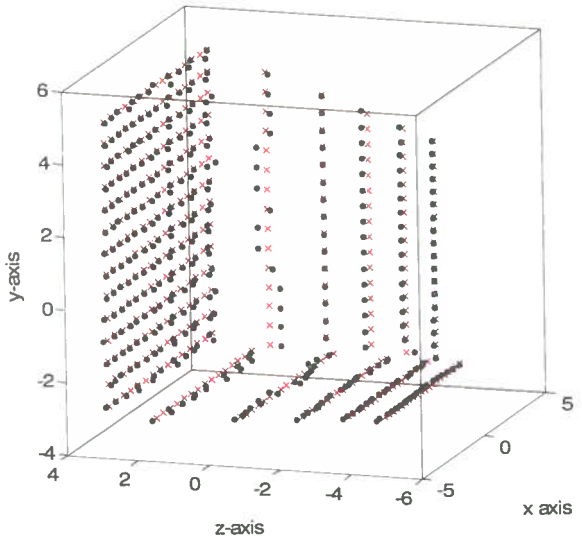


Figure 78: Reconstructed corner after the DDM deformation-reconstruction cycle converges

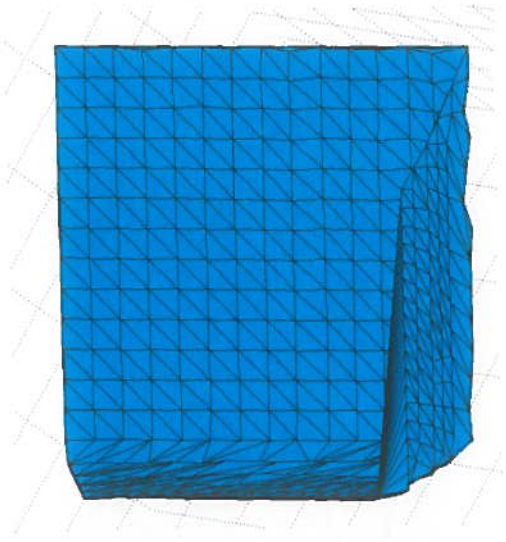


Figure 79: Reconstructed surface of corner by triangulating the recovered 3D scene coordinates

5.4 Reconstruction of non-planar scenes

In this section, the reconstruction of a simple scene involving a non-planar object is described. The purpose of this is twofold. Firstly, it is to demonstrate the ability of the DDM to successfully deform over non-planar surfaces. Secondly, we will show in this section that the reconstructed scene can still be improved using the model-based energy, using the assumption that the local surfaces of objects can be approximated by a plane. The scene used is very simple. It comprises a textured sphere raised above a low-textured planar background. The sequence is 20 frames long and is captured by a camera moving towards the sphere whilst undergoing a small amount of translation orthogonal to the optical axis. Frames 1, 6, 12, 16 and 20 are shown in the following sequence of images:

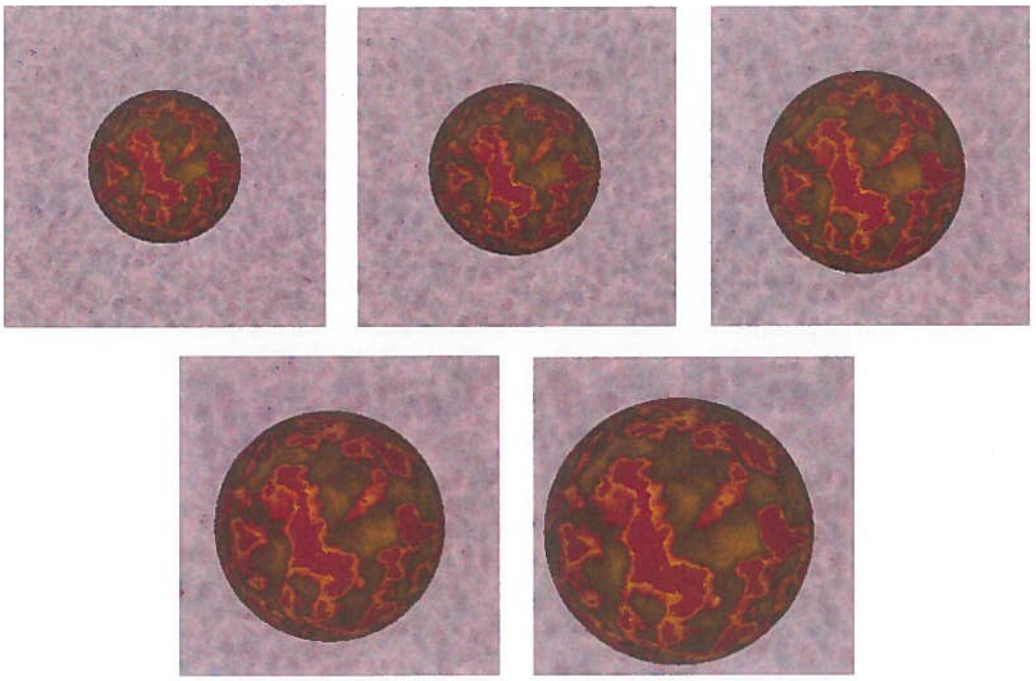


Figure 80: Example frames taken by a camera moving towards a textured sphere raised above a planar background

Similar to the previous section, a uniform DDM is initialised using the first frame, and deforms throughout the 20-frame sequence. The states of the DDM at each of these frames are shown in the following sequence of images:

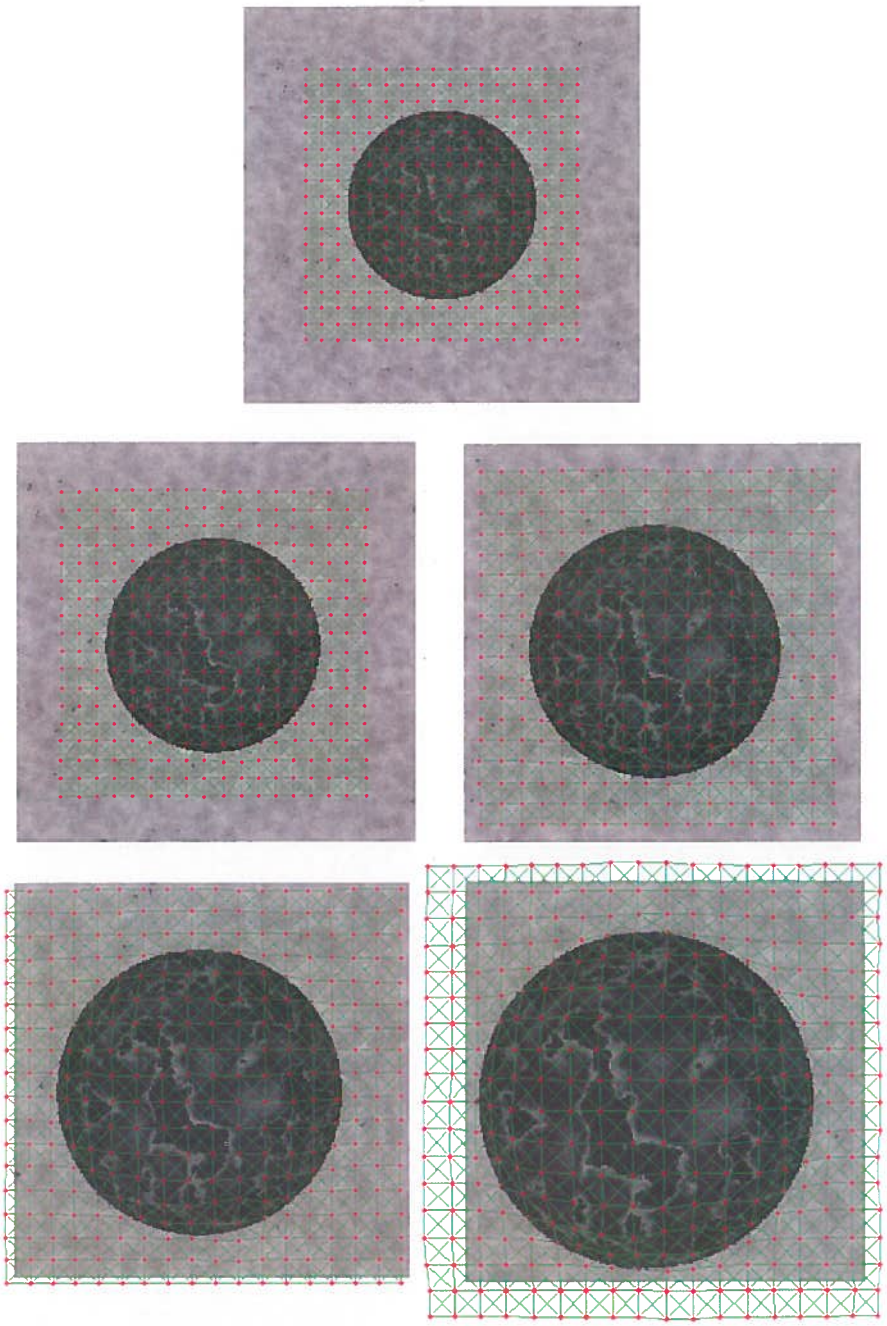


Figure 81: States of a DDM at various frames of an image sequence captured by a camera moving towards a non-planar object

The DDM appears to have deformed successfully throughout the image sequence. Notice that it has also predicted fairly accurately the positions of nodes which are beyond the viewable image plane towards the end of the sequence. It is perhaps clearer to gauge the error of the DDM as it deforms by plotting its RMS error as a function of time (or frame number). This is given in Figure 93.

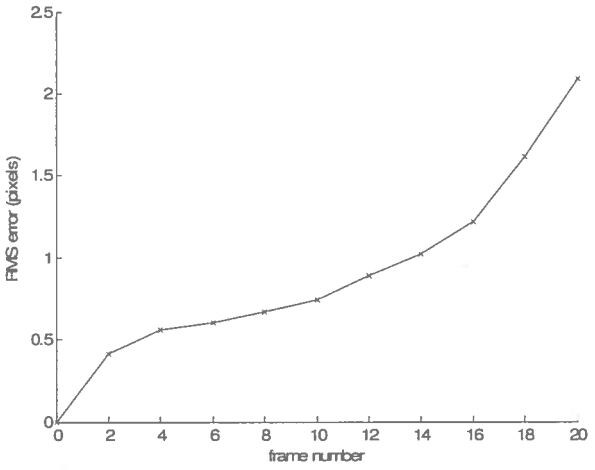


Figure 82: Growth in mean error of node positions as the DDM deforms throughout the image sequence in 5.4

The RMS error of the DDM grows over time to just over 2 pixels after frame 20. The reason for the accelerated growth is a consequence of the camera moving towards the scene, since the smaller the distance between the camera and scene, the more the projection of two consecutive frames will differ. Because the SSD comparison windows are not scale invariant, the SSD error becomes less reliable as a matching function the further into the sequence. Furthermore, because the spring energies constrain the deformation in 2D image space, these constraints also become less reliable.

Using this deformation, we can perform an initial reconstruction of the scene using the methods described previously. The next sequence of images shows this reconstruction compared with the scene's true 3D coordinate taken from three different views. The direction of the camera's optical axis in the first frame is along the z-axis.

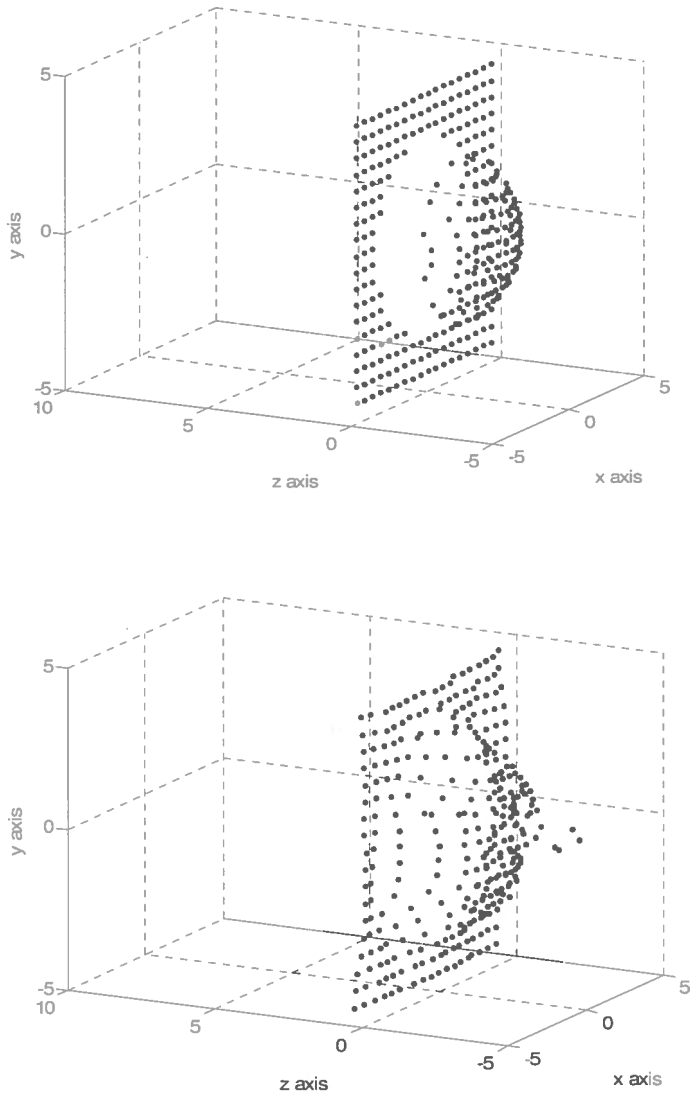


Figure 83: Asymmetric view comparing reconstructed and true scene coordinates of a non-planar scene. (Top) True coordinates. (Bottom) Reconstructed coordinates

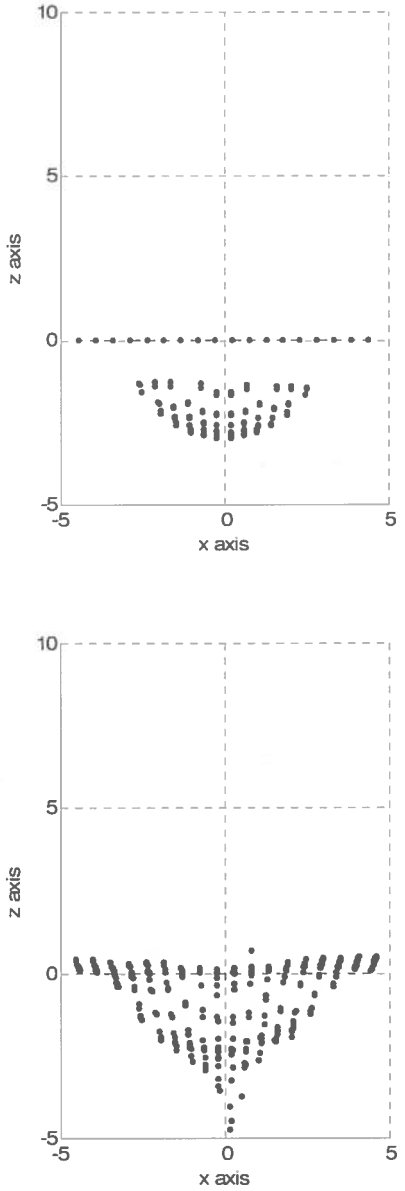


Figure 84: Bird's eye view comparing reconstructed and true scene coordinates of a non-planar scene. (Top) True coordinates. (Bottom) Reconstructed coordinates

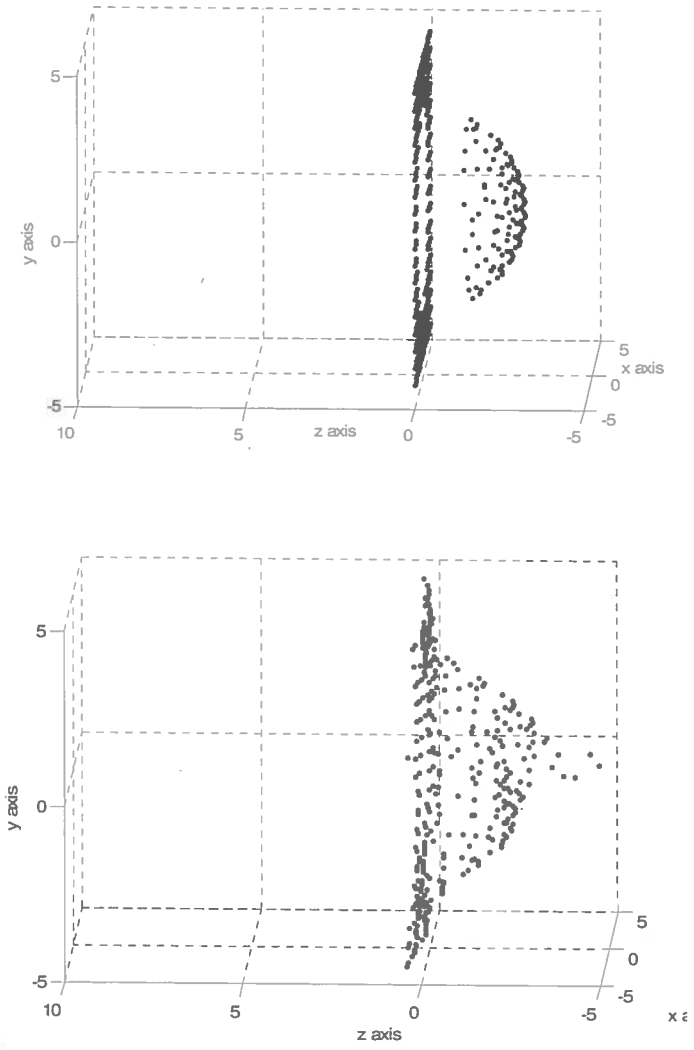


Figure 85: View along x-axis comparing reconstructed and true scene coordinates of a non-planar scene. (Top) True coordinates. (Bottom) Reconstructed coordinates

This series of views shows that the system has recovered the general structure of the scene. Although this is less accurate than the first reconstruction of the (planar) corner sequence, this provides evidence that the system can recover the surfaces of both planar and non-planar objects. However, it is clear that this reconstruction is not perfect. By looking at the three views, two problems are apparent. The first is related to the undesirable smoothing between the

background and the sphere. This has resulted in the separation in 3D between these objects (as shown in the ground truth views) being lost. The reason why this smoothing is occurring is because the depth discontinuity where the plane and the sphere intersect is relatively small, and this is not being detected by the disparity discontinuity preservation mechanism. This means that node connections crossing the object boundaries are not being severed, which results in the smoothing across the object boundaries. The second problem which can be seen is the loss of accuracy towards the centre of the sphere. This is because points in this region are very close to the centre of expansion, which results in very small disparities (Figure 86). When disparities are small, the depth estimates are much more sensitive to tracking errors caused by pixel quantisation and the discrete positioning of nodes for example. The result is worse reconstructions for nodes closer to the centre focus of expansion.

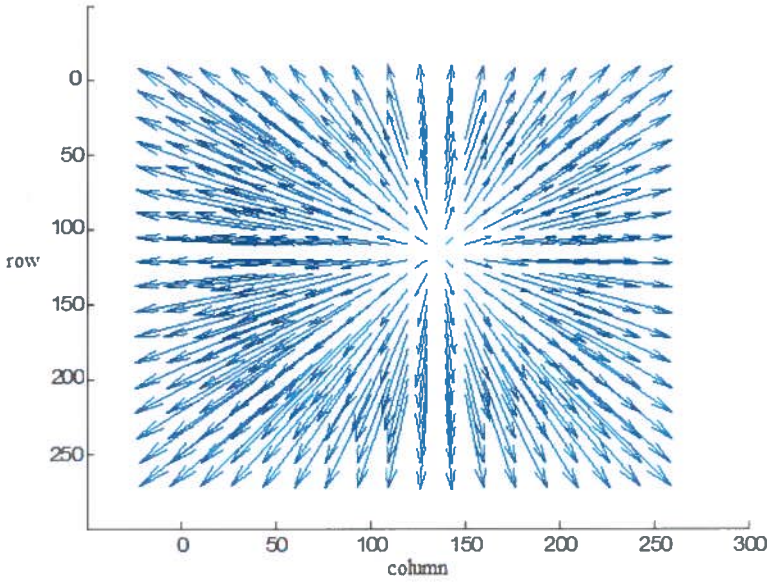


Figure 86: Plot of disparity vectors using the positions of nodes in the first and last frames

We can now use this initial reconstruction to estimate the surface normals at each point. An interesting problem which will affect the accuracy of these estimates is in choosing an appropriate neighbourhood size for the knn plane fitting algorithm. This problem is more significant in this scene than the previous because the sphere surface has higher curvature. Too

small, and the normal estimates become very sensitive to noise. However, too large and the local surface curvature is lost. Figure 94 illustrates this trade-off by plotting the mean surface normal estimation error against k, the number of nearest points used to fit the local plane.

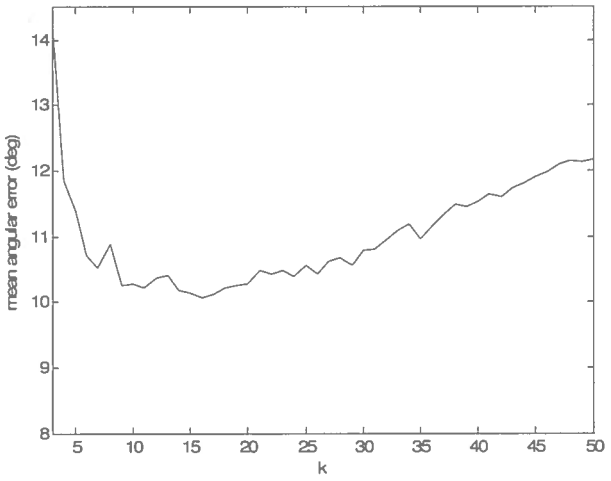


Figure 87: Trade-off in choosing the number of nearest neighbours used for fitting local planes to the reconstructed points

It appears that for this sequence a good choice for k is between 15 and 20 neighbours. When Hoppe *et al.* [29] used a similar method for fitting planes to Point Cloud Distributions, they manually selected k by inspecting the quality of the reconstructions. They also used a fixed value of k for all points. This is the approach taken in this work, although there are some obvious shortcomings. The first relates to the manual intervention, which is undesirable in any computer vision algorithm. This can be removed by using a k value which has been suitable in past reconstructions. The second problem is that a single, static k value will not be suitable for all points in the reconstruction. Points where there is a lower reconstruction noise should use fewer neighbours for fitting local planes than points where there is higher noise. Furthermore, the problem of smoothing normal estimates across depth discontinuities raises its ugly head again. These problems support the argument that k should be determined locally based on the certainty of the surrounding data and is an idea which warrants future investigation. A plot of the estimated surface normals using $k = 15$ viewed from above is given in Figure 88.

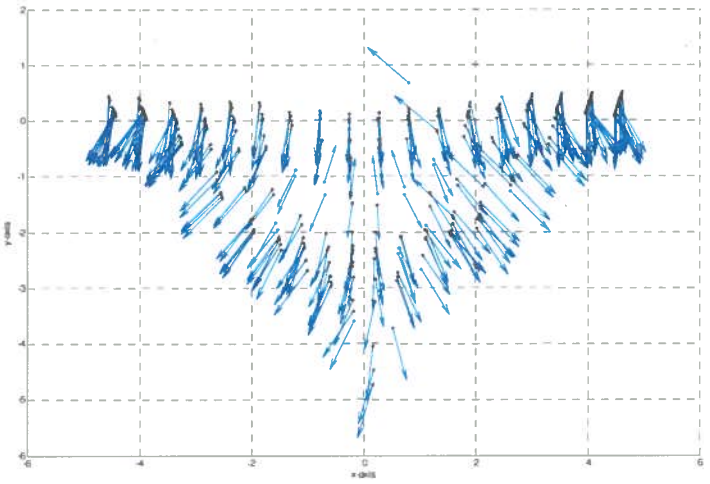
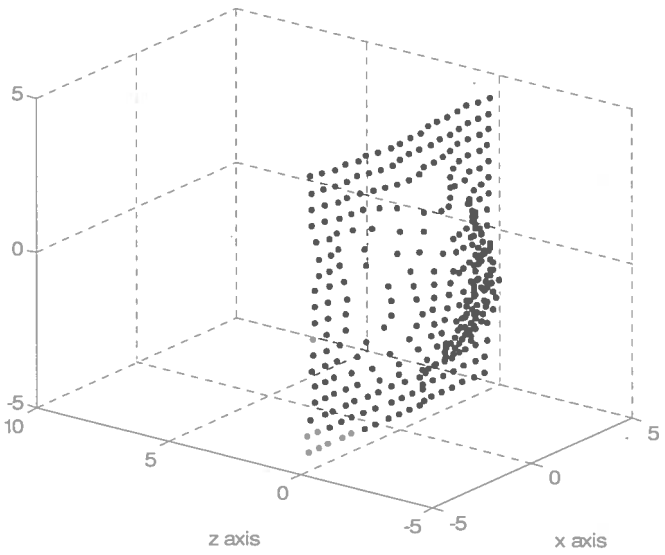
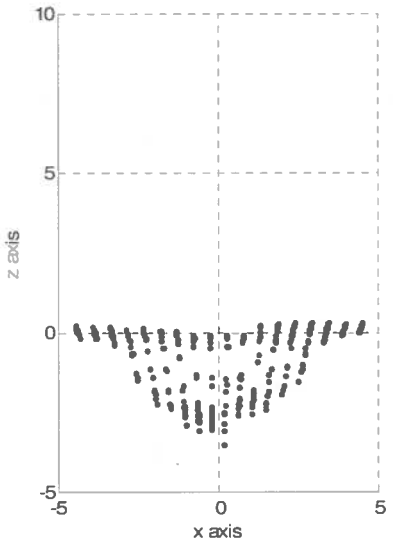


Figure 88: Estimate of surface normals after first reconstruction of a non-planar scene

We can see that the surface normals are reasonable approximations of the true surface normals, which differ by an average of 10°. Notice also that the normals at the poorly reconstructed points at the end of the sphere have been well estimated.

We can now deform the DDM for a second time using the model-based energy. By setting the model weight $a = 0.5$, the reconstruction from this second deformation process are shown from various views in the next three images:


5
0
-5
10
5
0
-5
z axis
-5
0
5
x axis


10
5
0
-5
z axis
-5
0
5
x axis

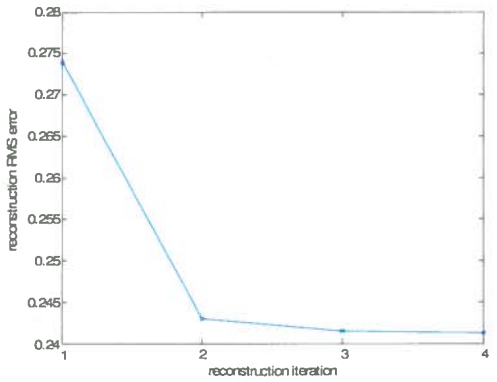


Figure 91: Decrease in reconstruction error with each reconstruction cycle

These plots show that as soon as the model-based energy is introduced in iteration 2, there is a clear performance gain; the reconstruction RMS error decrease from iteration 1 to iteration 2 by 10.6%. However, in subsequent iterations, this improvement is much less considerable. The reason for this is similar to the reconstruction in the previous section. This is due to a combination of the accuracy of the surface normal estimates reaching an upper limit, and the DDM deforming incorrectly even in light of better surface normal estimates. A side-by-side comparison of the reconstructions after iterations 1 and 4 is given in Figure 92.

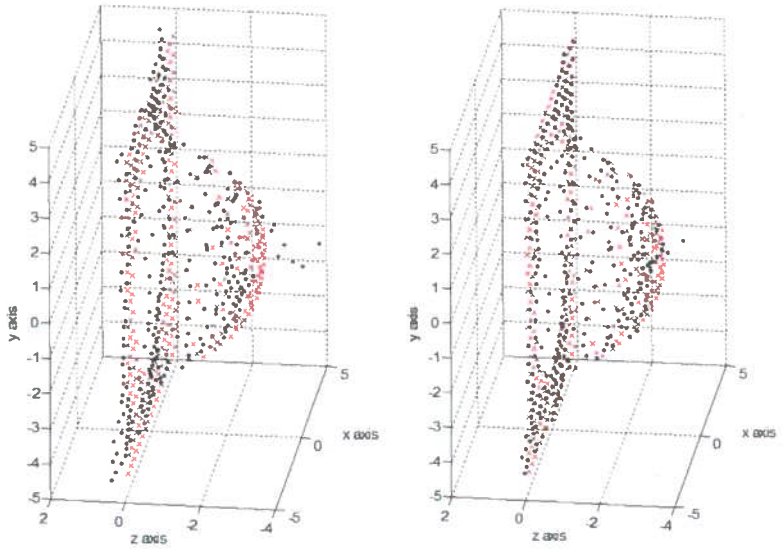


Figure 92: Improved reconstruction accuracy after four reconstruction iterations for a simple non-planar scene. (Left) Reconstruction after 1st iteration. (Right) Reconstruction after 4th iteration. Black dots mark predicted coordinates and red crosses mark true coordinates

It is worthwhile noting the effects on performance when the model energy weight α is increased too hastily (i.e. $\alpha = 1$). In this case, the internal constraints are only imposed in scene space, and are based on potentially inaccurate surface normal estimates. This is significant, since poorly estimated surface normals can result in reconstruction errors spiralling. This is because bad model constraints will encourage bad deformations, which will result in even worse reconstructions. The decision to use $\alpha = 0.5$ reflects a trade-off in realising the benefits of the model-based energy, but preventing this spiralling effect. Clearly, the model-based energy weight should be higher for those nodes whose normals have been well estimated. A mechanism is therefore needed for adapting α using estimates of the certainty of the surface normal estimates. This is further discussed in the future work section in the next chapter.

Chapter 6

Conclusion and Future Work

6.1 Summary of Work

This dissertation has outlined a new approach for reconstructing the 3D structure of a scene from a sequence of images captured by a moving camera. 3D Scene reconstruction is of great importance in the fields of computer vision, image processing and computer graphics, and finds many applications including the construction of 3D environments, the automatic construction of 3D CAD models from photographs and the creation of large photorealistic virtual environments. This approach has been primarily inspired by two fields in computer vision; Structure from Motion (SfM) and deformable (or active) models. A clear divide is present in the current SfM paradigm which separates those methods which are flow-based from those which are feature-based. While flow-based methods attempt to compute dense motion fields between image pairs or triples, these become unstable and computationally expensive when used over longer sequences. Indeed, Structure from Motion using dense image flow fields is notoriously unreliable. By contrast, feature-based motion methods provide more stable motion estimates although can only be used for sparse reconstructions. These sparse reconstructions must be interpolated if denser reconstructions are required, which itself remains a challenging and unsolved problem. In this work, a new approach has been developed which models motion through video sequences using a deformable model. The major contributions in this work are summarised as follows:

- Reliable semi-dense point correspondences throughout a video sequence using the Deformable Disparity Model. The model performs well using a number of test sequences at regions of both high and low texture.
- A new type of motion constraint imposed in scene space, rather than image space. When the normals to the surface at reconstructed points are reliably estimated, this can significantly improve reconstruction results.
- A process for iteratively refining reconstructions using by progressively more reliable scene-space constraints
- The estimation of surface normals directly from a 3D Point Cloud Distribution (PCD). This eliminates the usual cost of fitting a smoothed polygonised surface model to the PCD to estimate surface normals, and is inherently robust to noise.
- The reconstruction of a dense or semi-dense set of points by matching curves in the spatiotemporal volume. Evidence gathered has suggested that the spatiotemporal curve matching algorithm outperforms conventional triangulation
- A new way of detecting disparity discontinuities in an image sequence using the correlation between multiple SSD sub-window scores

6.2 Future Directions

In addition to the progress so far made in the reconstruction of 3D scenes using the DDM, there are a number of possible directions for future development. Several of these are now presented.

Camera auto-calibration

A very useful extension to the reconstruction system so far outlined is to incorporate camera auto-calibration, which removes the requirement for known camera parameters. Clearly, this addition would greatly increase the system's scope, since it could then handle free camera motion. The introduction of the autocalibration paradigm was a major breakthrough in the early nineties related to the intrinsic calibration, according to which calibration is achieved not with the aid of a calibration pattern but by observing a number of image features in a set of successive images. Maybank and Faugeras [51] proved that if intrinsic parameters are constant,

autocalibration can be used for the Euclidean reconstructions of a scene. A Euclidean reconstruction differs from a metric reconstruction by a ridged Euclidean transformation, which reflects the scene's arbitrary reference frame, and a uniform scale factor. Autocalibration is in general performed by first computing the fundamental matrix from a set of correspondences [43]. This is usually done by making use of the Kruppa equations [36], where with a minimum of three displacements we can obtain the internal parameters of the camera using a system of polynomial equations. The fact that the DDM provides semi-dense correspondences is an advantage, as the additional redundancy is likely to increase the robustness of the fundamental matrix estimate. From this, the essential matrix can be immediately obtained, from which we can decompose into the rotation and translation motion parameters. A good review of camera autocalibration can be found in [28].

Improving the surface normal estimates

The method described in section 4.3 for fitting least-squares local planes to reconstructed points in the Point Cloud Distribution (PCD) suffers several shortcomings which prevent more accurate surface normal estimates in later reconstruction iterations. It is useful to consider some of the factors which may affect the accuracy of the surface normal estimates. Four of these have been identified as follows:

- The amount of noise in the PCD
- The local curvature of the underlying surface
- The density and distribution of the points in the PCD
- The neighbourhood size used in the estimation process

It would be worthwhile to study the contribution of these factors on the normal estimation process, since we may then be in a better position to understand how best to choose the neighbourhood sizes. This analysis would then pave the way for computing the optimal neighbourhood size adaptively at each point, based on local information such as estimated noise, local sampling density and curvature. Another considerable benefit of this analysis is that we can use this information to adapt α, the model-based energy weight. For example, if a DDM node's potion in the PCD has high local curvature, or large amounts of noise, then α should be reduced accordingly.

Sub-pixel intensity matching

The Sum of Squared Differences, like most area based matching functions operates by comparing areas of discrete intensity. While this may be suitable for high resolution images, the loss of accuracy incurred in lower resolution images can be a considerable source of node drift. To estimate the sub-pixel displacements, two general methods have been used. The first searches for the maximum local intensity correlation using high spatial resolution images obtained by image interpolation. For example, Matthies *et al.* interpolated scanlines by a factor of 4 using a cubic interpolant before computing the SSD score [50]. Tian and Huhns have written a survey paper comparing a number of these algorithms for sub-pixel stereo matching [73]. The alternative approach is to estimate the maxima of the similarity function by fitting a parabola to the three pixels near its maxima [69]. The problem with the first approach is that they require a large amount of memory, and are computationally expensive [39]. With respect to the DDM, sub-pixel matching at every optimisation iteration is not computationally feasible. However, sub-pixel accuracy can be attained using either method applied as a post-processing operation once the DDM has fitted (discretely) to the new image data.

Denser models

The majority of the DDMs used throughout this dissertation have been initialised with a constant density. In many applications however, this density will be chosen to reflect the required reconstruction density, the computational resources available and the resolution of the image sequence. The density of the DDM does not affect its internal energy, because the spring networks are invariant to scale. This is useful because it means little tuning is necessary. The major difference is in the increased computational cost of deforming the model. An interesting application for sparser DDMs that would be worthwhile investigating is to use their recovered surface normals for denser DDMs. This equates to a coarse-to-fine reconstruction process; the surface models from coarser reconstructions are passed down to finer DDMs. The advantage of this is to essentially remove the first reconstruction iteration where the model-based energy is unused.

Does the matching of spatiotemporal curves always out-perform triangulation?

In section 5.3, evidence was presented to suggest that the matching of spatiotemporal curves can outperform conventional stereo triangulation irrespective of the frame pair sued. This is a

significant finding, particularly as these two methods have never been compared before. A reasonable hypothesis for this is that spatiotemporal depth curves time-integrate depth information over the entire duration of the sequence rather than only for a single frame pair, which results in more robust depth estimates. It would be very worthwhile to further investigate this finding.

Implementation efficiency

At present, the entire system has bee implemented in Matlab. During the developmental stages this has had considerable advantages over most other programming languages because of its excellent visualisation and prototyping capabilities. However, because Matlab is an interpreted language, it can be very inefficient. There is no doubt that a large processing speed improvement can be made by implementing the system in a compiled language such as C

More extensive evaluation

It is necessary to provide a more rigorous evaluation of its performance, particularly using real scenes. This has been hindered somewhat by the lack of benchmark sequences, and the system's current requirement for known camera parameters. This restricts the range of possible scenes and image sequences which can be used for the evaluation. This problem will be reduced once autocalibration has been introduced into the system.

Bibliography

[1] Aschwanden, P., Guggenb, W., Experimental results from a comparative study on correlation-type registration algorithms, *in Robust Computer Vision*, pp 268-289, 1992

[2] Baker, H. H., Binford, O., Depth from edge and intensity based Stereo, *in Proc. International Joint Conference Artificial Intelligence*, pp 631-636, 1981

[3] Barron. J., Fleet, D.J., Beauchemin, S.S, Performance of Optical Flow Techniques, *in Proc. International Journal of Computer Vision* (IJCV) 12(1):, pp 43-77, 1997

[4] Beardsley, P., Torr, P., Zisserman, A., 3D model acquisition from extended image sequences, *in Proc. European Conference on Computer Vision* (ECCV), pp 683-695, 1996

[5] Beardsley, P. A., Zisserman, A., Murray, D.W., Navigation using affine structure from motion, *in Proc. European Conference on Computer Vision* (ECCV), pp B:85-96, 1994

[6] Beauchemin, S. S., Barron, J. L., The Computation of Optical Flow, *ACM Computing Surveys*, 1995

[7] Blake, A. Brelstaff, G., Geometry from Specularity, *in Proc. International Conference on Computer Vision* (ICCV), pp 394-403, 1988

[8] Blake, A., Marinos, C., Shape from texture: estimation, isotropy and moments, *Artificial Intelligence*, Vol. 45, pp 323-380, 1990

[9] Bolles, R., Baker, H., Marimont, D., Epipolar-plane image analysis: An approach to determining structure from motion, *in Proc. International Conference on Computer Vision* (ICCV), pp 20637, 1987

[10] Boughorbel, F., Page, D., Abidi, M., Automatic reconstruction of large 3D models of real environments from unregistered data-sets, *in Proc. Conference on 3D Image Capture and Applications III* (CICA), Vol. 3958, pp 234-243, 2000

[11] Bourke, P., The shortest line between two lines in 3D, http://astronomy.swin.edu.au/~pbourke/geometry/lineline3d.html, 2004

[12] Canny, J., A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, 1986

[13] CMU CIL, The Computer Vision Homepage, http://www.cs.cmu.edu/~cil/vision.html, 2005

[14] Daum, M., Dudek, G., On 3-d surface reconstruction using shape from shadows, *in Proc. Conference on Computer vision and Pattern Recognition*, pp 461-468, 1998

[15] Davis, J., Ramamoothi, R., Rusinkiewic, S., Spacetime Stereo : A Unifying Framework for Depth from Triangulation, *IEEE Transactions on Computer Vision and Pattern Recognition*, 2003

[16] Faugeras, O., Luong, Q. T., Papadopoulo, T., Representing stereo data with the delaunay triangulation, *Artificial Intelligence*, 44:41-87, 1990

[17] Faugeras, O., Luong, Q. T., Papadopoulo, T., The Geometry of Multiple Objects, *MIT Press*, 2001

[18] Favaro, P., Soatto, S., Learning depth from defocus, *in Proc. European Conference on Computer Vision* (ECCV), 2002

[19] Fitzgibbon, A., Zisserman, A., Automatic 3d model acquisition and generation of new images from video sequences, *in Proc. European Signal Processing Conference*, pp 1261-1269, 1998

[20] Fusiello, A., Roberto, V., Trucco, E., Efficient stereo with multiple windowing, *IEEE Transactions on Computer Vision and Pattern Recognition*, pp 858–863, 1997

[21] Geiger, D., Ladendorf, B., Yuille, A., Occlusions and binocular stereo, *in Proc. European Conference on Computer Vision* (ECCV), 1995

[22] Grammalidis, N., Strintzis, M. G., Use of Dynamic Programming for Dense Depth Estimation from Uncalibrated Image Sets, *International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging*, 1999

[23] Grauman, K., Shakhnarovich, G., Darrell, T., Virtual Visual Hulls: Example-Based 3D Shape Inference from Silhouettes, *In Proc. of the 2nd Workshop on Statistical Methods in Video Processing*, 2004

[24] Hadziavdic, V., A Comparative Study of Active Contour Models for Boundary Detection in Brain Images, *University of Bergen*, 1998

[38] Leclercq, P, Assessing Stereo Algorithm Accuracy, *Image and Vision Computing*, New Zealand, 2002

[39] Lecordier, B., Lecordier, J.C., Trinite, M., Iterative sub-pixel algorithm for the crosscorrelation PIV measurements, *In Proc. 3rd International Workshop on PIV*, pp 37-43, 1999

[40] Lee, I., Curve reconstruction from unorganized points, *Computer Aided Geometric Design*, 17:161, 2000

[41]Lu, Y., Zhang, J.Z., Wu, Q.M.J., Li, Z.N., A survey of motion-parallax-based 3-D reconstruction algorithms, *SMC-C(34)*, No. 4,, pp 532-548., 2004

[42] Lucas, B. and Kanade, T., An iterative image registration technique with an application to stereo vision, *In Proc. Seventh International Joint Conference on Artificial Intelligence*, pp 674-679, 1981

[43] Luong, Q.T, Faugeras, O., Self-calibration of a moving camera from point correspondences and fundamental matrices., *in Proc. International Journal of Computer Vision* (IJCV), 22(3):261-289, 1997

[44] MacDonald, D., Worsley, K., Avis, D., Evans, A. C., 3D Mapping of Variability in Cortical Anatomy., *In Proc. of SPIE Visualization in Biomedical Computing*, volume 2359, pp 160-169, 1994

[45] Manessis, A., Hilton, A., McLauchlan, P., Palmer, P., A statistical geometric framework for reconstruction of scene models, *In Proc. British Machine Vision Conference* (BMVC), pp 222-231, 2000

[46] Marr, D., Poggio, T., Cooperative computation of stereo disparity, *Science*, vol. 194, pp 283–287, 1976

[47] Marr, D., Poggio, T., A computational theory of human stereo vision, *in Proc. Royal Society. London*, Vol. B 204, pp 301-328, 1979

[48] Marr, D., Hildreth, E. C, Theory of edge detection, *in Proc. Royal Society London.*, B-207:187--217, 1990

[49] Martin, W., Aggarwal, J. K., Volumetric description of objects from multiple views, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp 150-158, 1983

[50] Matthies, L. H., Szeliski, R., Kanade, T, Kalman filter-based algorithms for estimating depth from image sequences, in *Proc. International Journal of Computer Vision*, 3:209–236, 1989

[51] Maybank, S. J., Faugeras, O., A theory of self-calibration of a moving camera, *in Proc. International Journal of Computer Vision* (IJCV), 8(2):123-151, 1992

[52] McInerney, T., Terzopoulos, D., Deformable Models in Medical Image Analysis: A Survey, *Medical Image Analysis*, 1(2), pp 91-108, 1996

[53] Mencl, R.., Reconstruction of Surfaces from Unorganized 3D Points Clouds, PhD Thesis, *Dortmund University, Germany*, 2001

[54] Miller,J.V., Breen, D.E., Lorensen, W.E., O'Barnes, R.M., Wozny, M.J., Geometrically deformed models: A method for extracting closed geometric models from volume data., *Computer Graphics* (SIGGRAPH'91 Proceedings), 1991

[55] Mokhtarian, F., Suomela, R., Robust image corner detection through curvature scale space, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp 1376-1381, 1998

[56] Moravec, H. P., Toward automatic visual obstacle avoidance, in *Proc. International Joint Conference on Artificial Intelligence*, pp 584–584, 1977

[57] Nayar, K. K., Nakagaw, Y., Shape from Focus, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp 824-831, 1994

[58] Nobuhara, S., Matsuyama, T., Heterogeneous Deformation Model for 3D Shape and Motion Recovery from Multi-Viewpoint Images, *Second International Symposium on 3D Data Processing, Visualization and Transmission*, 2004

[59] Nüchter, A., Surmann, H., Hertzberg, J., Automatic Classification of Objects in 3D Laser Range Scans, *in Proceedings of the 8th Conference on Autonomous intelligent Systems* (IAS 2004), 2004

[60] Park, J., Metaxas, D., Axel, L., Analysis of left ventricular wall motion based on volumetric deformable models and MRI-SPAMM, *Medical Image Analysis*, 1(1), pp 53-71, 1996

[61] Park, S., Kim, M., Lee, C. W., Mobile robot navigation based on direct depth and color-based environment modeling, *in Proc. International Conference on Robotics and Automation* (ICRA), 2004

[62] Ponce, J., Lazebnik, S., Rothganger, F., Schmid, C., Toward True 3D Object Recognition, *Congrès de Reconnaissance des Formes et Intelligence Artificielle*, Toulouse, France, 2004

[63] Promayon, E., Baconnier, P., Puech, C., Physically-Based Deformations Constrained in Displacements and Volume., *Computer Graphics forum*, 15(3), pp 155-164, 1996

[64] Redert, P.A., Henriks, E.A., Biemond, J., Correspondence estimation in image pairs, *IEEE Signal Processing Magazine, special issue on 3D and stereoscopic visual communication*, Vol. 16, No. 3, pp 29-46, 1999

[65] Rodrigues, R., Fernandes, A., Overveld, K., Ernst, F., Reconstructing depth from spatiotemporal curves, *in Proc. International Conference on Vision Interface*, 2002

[66] Roma, N., Santos-Victor, J., Tomé, J., A Comparative Analysis of Cross-Correlation Matching Algorithms Using a Pyramidal Resolution Approach, *Evaluation Methods in Computer Vision, vol.50 of Series in Machine Perception and Artificial Intelligence*, pp 117-142, 2005

[67] Schaffalitzky, F., Zisserman, A., Viewpoint Invariant Texture Matching and Wide Baseline Stereo, *in Proc. International Conference on Computer Vision*, 2001

[68] Scharstein, D, Szeliski, R., A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms, *International Journal of Computer Vision* (IJCV), 2002

[69] Shimizu, M., Okutomi, M., Precise Sub-pixel Estimation on Area-Based Matching, in Proc. In *Proc. International Conference on Computer Vision* (ICCV), Vol. 1, pp 90-97, 2001

[70] Slabaugh, G., Culbertson, B., Malzbender, T., Schafer, R., A survey of methods for volumetric scene reconstruction from photographs, *Technical Report, Center for Signal and Image Processing, Georgia Institute of Technology*, 2001

[71] Smith, S. M., Brady, J. M., Susan—A new approach to low level image processing, *in Proc. International Journal of Computer Vision* (IJCV), vol. 23, pp 45-78, 1997

[72] Snow, D. Viola, P., Zabih, R., Exact voxel occupancy with graph cuts, *in Proc. Conference on Computer Vision and Pattern Recognition*, 2002

[73] Tian, Q., Huhns, M. N., Algorithms for subpixel registration, Computer Vision, *Graphics and Image Processing*, 35:220-233, 1986

[74] Tohka, J., Surface Extraction from Volumetric Images Using Deformable Meshes, *in Proc. European Conference on Computer Vision* (ECCV), pp 350 - 364, 2002

[75] Torresani, L., Hetzmann, A., Bregler, C., Learning Non-Rigid 3D Shape from 2D Motion, *Neural Information Processing Systems*, 2003

[76] Tu, X., Artificial Animals for Computer Animation: Biomechanics, Locomotion, Perception and Behavior, *New York: Springer*, 1999

[77] Veksler, O., Extracting Dense Features for Visual Correspondence with Graph Cuts, *IEEE Transactions on Computer Vision and Pattern Recognition*, pp 689-694, 2003

[78] Viola, P., Jones, M., Robust real-time face detection, *in Proc. International Conference on Computer Vision* (ICCV), 2001

[79] Weng, J., A theory of image matching, *in Proc. International Joint Conference on Computer Vision*, pp 200-209, 1990

[80] Williams, L.R., Thornber, K.K., A Comparison of Measures for Detecting Natural Shapes in Cluttered Backgrounds, *International Journal of Computer Vision* (IJCV), 1999

[81] Zhang,R., Tsai, P. S., Cryer, J. E., Shah, M., Shape from Shading: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp Vol. 21, No. 8, 1004