# Multi-Modal Tracking using Texture Changes

Christopher Kemp and Tom Drummond
{ck231|twd20}@eng.cam.ac.uk
Department of Engineering
University of Cambridge
Cambridge, CB1 2BZ, UK

**Abstract**

We present a method for efficiently generating a representation of a multi-modal posterior probability distribution. The technique combines ideas from RANSAC and particle filtering such that the 3D visual tracking problem can be partitioned into two levels, while maintaining multiple hypotheses throughout. A simple texture change-point detector finds multiple hypotheses for the position of image edgels. From these, multiple locations for each scene edge are generated. Finally we determine the best pose of the whole structure. While the multi-modal representation is strongly related to particle filtering techniques, this approach is driven by data from the image. Hence the resulting system is able to perform robust visual tracking of all six degrees of freedom in real time. Real video sequences are used to compare the complete tracking system to previous systems.

## 1 Introduction

Robust real-time visual tracking has important applications such as the guidance of robots or autonomous vehicles. Our ultimate goal is the control of a miniature aerial vehicle (MAV) such as a model helicopter, and this provides particularly challenging problems. To control the flight it is necessary to have a real-time estimate of the vehicle's position and orientation (pose). In previous work with large model helicopters, e.g. [6], this has been achieved using a combination of GPS measurements, inertial sensors and visual information. Our aim is to control much smaller indoor vehicles which have greatly reduced payload capacities. Accurately locating the vehicle using only the information from a sub-miniature transmitting video camera would therefore be a great advantage.

Visual tracking systems aim to follow motion in a sequence of images from a video camera. If the video camera is moving through a static world, this can be used to recover the motion of the camera in real-time. Although a prototype MAV system is not yet complete, initial experiments suggest that images obtainable are likely to contain large pixel noise and suffer from substantial motion blur. Further, MAVs are particularly unstable platforms and hence the tracking system must be able to handle large motion disturbances.

To address these problems, we propose a novel multiple hypothesis approach which can reliably track polyhedral surroundings for which an accurate 3D edge model is known. This paper makes two contributions. In Section 2 we show how ideas from RANSAC can be combined with particle filtering to generate multi-modal posterior distributions. A key benefit of this method is that it permits the tracking problem to be decomposed into two

BMVC 2004 doi:10.5244/C.18.88

steps. Firstly, given noisy edgel locations, a multi-modal posterior for the location of each scene edge can be generated. Secondly, these can be used to generate a multi-modal posterior for the camera pose. The system requires multiple edgels to be found along image scan-lines and so in Section 3, we describe the second contribution: a method of finding multiple texture change-points in a pixel sequence. The effectiveness of the resulting real-time tracking system is discussed in Section 4.

## 1.1 Previous Work

Existing real-time tracking systems can be split into two groups: those which rely on a predetermined model of the tracked scene and those which learn the surroundings as the sequence progresses. Some very encouraging progress has been made recently in the latter group [4] and since the construction of models is costly, systems which automatically create or adapt models are advantageous. However, in areas where the scene is reasonably static, the use of prior knowledge (in the form of a model) can only be beneficial. In particular, model based systems can more easily reject unmodelled features and so are more tolerant of non-static elements in the images, such as passing humans.

Many model based tracking schemes work by matching point features. Some methods sequentially match features from adjacent frames e.g. [14]. Recent contributions [16] have largely eliminated both drift and jitter through the use of inter-frame matching combined with precomputed keyframes. Whilst many scenes can be successfully tracked using point features, some indoor areas where our MAV may be used are surprisingly free of reliable point features. Also, point detectors can have difficulties with images with repetitive textures such as carpets or tiles and struggle with noisy and blurred images.

Another technique is to match image edges. Edge detection is more robust to illumination and aspect changes and it is more straightforward to detect edges in the presence of noise and blur. Typically however it is difficult to match image edges based on appearance and so a common assumption [1, 9, 11, 7] is that closer correspondences (to a predicted position) are more likely to be correct. However, if there is a high density of similar features, or large unpredictable image motions occur, this assumption is violated. Although robust estimators are used to tolerate a small number of false correspondences, in these conditions tracking often converges on an incorrect local minimum. It is also possible [12] to work directly with image intensities rather than first performing edge extraction. However, the same problems with local minima apply.

A number of techniques exist to deal with local minima. Particle filtering has proved successful for many applications [5, 10]. However, when tracking motion in all six degrees of freedom, it is difficult to achieve the required robustness in real-time. Alternative schemes which also consider multiple hypotheses have been used. In [2] the multi-modal representation used is a piecewise combination of Gaussian distributions and is used to track rapidly moving human figures, while in [3] image corners are used with a discrete, tree-like, representation of the multiple hypotheses.

## 1.2 Overview of the Approach

The proposed system follows the same basic structure as many frame-to-frame tracking systems, such as the early method described in [9]. For each new frame a 3D model of the surroundings is rendered into the imaging plane at the pose obtained from the previous

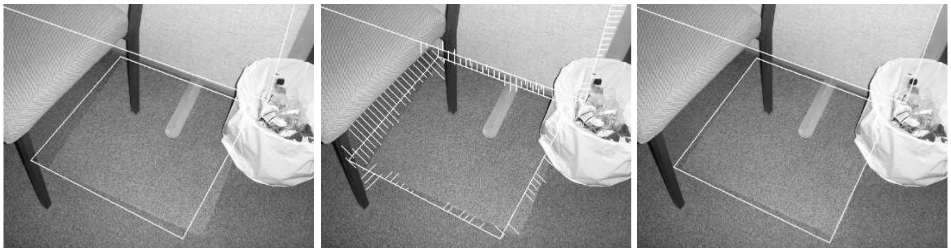

Figure 1: An overview of the system tracking a carpet tile and screen, despite occlusion. From the left: for each new frame the model is rendered in the predicted location. Next, perpendicular searches locate matches for each model edge in the image. Finally the best overall pose is found.

frame. The image is next searched to identify the new location of the rendered model edges. This is performed using 1D searches along regularly spaced scan-lines, perpendicular to each rendered model edge. Finally an optimisation is performed, adjusting the six pose parameters to align the rendered edges with those found in the image. These stages are shown in Figure 1.

The system proposed in [1] used RANSAC to improve edge based tracking by robustly estimating the location of each model edge in the image. Here, we extend this by admitting multiple hypotheses for each line, as described in the next section. This gives the resilience to local minima demonstrated by particle filtering approaches whilst still providing real-time operation.

## 2 Multi-Modal Posterior Representations

The Random Sample Consensus (RANSAC) paradigm, introduced by Fischler and Bolles [8], has been widely used throughout the computer vision field to obtain robust estimates of model parameters. The prototypical example of its use is to obtain the parameters of a straight line which best fits a set of noisy measurements, $M$, containing outliers. The parameters describing the straight line through two measurement points selected at random are obtained. A consensus count for this hypothesis is then found by comparing, to a threshold value, the distance from the line to all the measurement points. The process is repeated $N$ times and the hypothesis with the highest consensus is finally selected.

The consensus system proposed in [8] can be expressed as:

$$C(\theta) = \sum_{m \in M} f(\,\mathrm{dist}(m,\theta)\,) \quad , \tag{1}$$

where $\mathrm{dist}(m,\theta)$ is a function giving the distance from a measurement point $m$ to the straight line defined by $\theta$ and:

$$f(d) = \begin{cases} 0 & (d < t) \\ 1 & (d > t) \end{cases} \quad , \tag{2}$$

where $t$ is the consensus range or threshold. Defined in this way, the hypothesis with the smallest consensus score is chosen. In [15] it was observed that the performance of the system is improved if, at no extra computational cost, this last function is replaced with:

$$\tilde{f}(d) = \begin{cases} d & (d < t) \\ t & (d > t) \end{cases} \quad . \tag{3}$$

When the consensus score is expressed in this way, RANSAC is being used to find the minimum of a function defined continuously over model parameters. [15] shows how (under certain assumptions) $\tilde{f}(d)$ can be used to obtain a consensus score which is the negative log likelihood that the hypothesis is correct. From this the posterior probability $P(\theta)$ can be easily calculated, as used in Section 2.1.

The approach proposed here borrows an idea from CONDENSATION and retains *all* of the hypotheses tested. Each is then treated as a particle in the representation of the posterior distribution. Suppose that a set of measurements is obtained which contains noisy points from more than one straight line, along with some outliers. RANSAC applied to this data set would find the parameters of only the line with the most corresponding measurement points (given a sufficiently large $N$). Instead, if the consensus score from all $N$ RANSAC hypotheses are retained, these can be used to give an approximation to the multi-modal posterior. This representation is beneficial if the next task is to draw samples from the posterior since this can be approximated by drawing from the stored hypotheses. A good approximation can be obtained with a relatively low $N$ since there are typically many proposal points at or near the modes of $P$. However, this means that it is *not* correct to simply select hypothesis $n$ with probability $p_n = P(\theta_n)$.

If a posterior distribution to be represented is $P(\theta)$, and given a set of points $\Theta = \theta_1 \ldots \theta_N$ drawn from a (different) proposal distribution $Q(\theta)$, then draws from the posterior distribution can be approximated by resampling from $\Theta$, where the probability of selecting sample $n$ is:

$$p_n = \frac{P(\theta_n)}{Q(\theta_n)} \quad . \tag{4}$$

Provided that $Q(\theta) > 0$ for all $\theta$ where $P(\theta) > 0$ this process converges to sampling from $P$ as $N \rightarrow \infty$. In the case of multi-modal RANSAC, the proposals are those generated by random sampling of data tuples created by a process driven by $P$. The probability of obtaining the sample point, $Q(\theta_n)$, is a function of the consensus set (the number of data points within consensus range of $\theta_n$). If the size of the consensus set is $c_n$ and the number of data samples needed to form a hypothesis is $j$, then $Q$ is given by:

$$Q(\theta_n) = k \ {}^{c_n}\mathbf{C}_j = k \frac{c_n!}{j!(c_n - j)!} \quad , \tag{5}$$

provided that the same consensus set is obtained from any $j$-tuple subset. Although this is not strictly the case, in practice the approximation works well. Hence when sampling from a multi-modal representation of a posterior generated by RANSAC, each hypothesised sample, $\theta_n$ should be chosen with a probability, $p_n$, given by:

$$p_n = K \frac{P(\theta_n)}{{}^{c_n}\mathbf{C}_j} \text{ , with } K \text{ chosen such that } \sum p_n = 1. \tag{6}$$

## 2.1 Application to Finding Straight Edges in an Image

Given an initial estimate for the location of an edge in an image, the task is to generate the posterior distribution of the true position of that edge given the image data. This distribution is defined over the model parameters $r_1$ and $r_2$, which describe the motion of the two endpoints of the line (see Figure 2). First, edgels are obtained, either at points of maximal intensity gradient, or using the scheme described in Section 3. The ideas proposed in the previous section are then applied to find straight lines through these edgels.

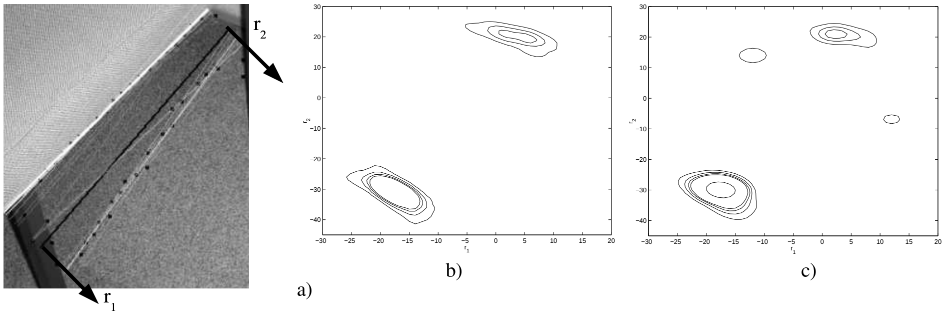

Figure 2: a) An closeup of Figure 1 showing a rendered edge (black), the two model parameters, the detected edgels, and samples from the multi-modal representation (white) b) contours of the true multi-modal posterior c) the sampled representation at the same scale.

The required posterior distribution describes the probability density that the corresponding straight edge in the image is at $\theta = \{r_1, r_2\}$. An image edge is modelled as causing edgels which are displaced from a straight line by additive Gaussian noise. If $d_{n,s}$ is the distance along each perpendicular scanline $s$, from the intersection with the hypothesis edge $\theta_n$ to the nearest edgel, a close approximation[1] to the negative log likelihood is:

$$C(\theta_n) = \text{const} + \sum_s \tilde{f}(d_{n,s}{}^2) \quad , \tag{7}$$

with $\tilde{f}()$ as in (3). Hence the posterior is given by:

$$P(\theta_n) = K\, e^{-\frac{C(\theta_n)}{2\sigma^2}} \quad . \tag{8}$$

In this case the $N$ samples are obtained by randomly selecting two edgels and calculating the parameters that describe the straight line through the two points. The multi-modal representation is completed by calculating and storing $\Theta_n = \{\theta_n, P(\theta_n)\}$ for each. Since in this case $j = 2$, samples can then be drawn from the representation by selecting $\theta_n$ with a probability proportional to:

$$p_n = \frac{P(\theta_n)}{c_n(c_n - 1)} \quad , \tag{9}$$

where $c_n$ is the corresponding consensus count as defined for (5).

The resulting system is demonstrated in Figure 2 which compares the full posterior distribution to that generated by sampling the RANSAC representation, with $N = 100$.

[1] This approach fixes the number of inliers to be constant and hence the inlier/outlier ratio has only a small effect on the true negative log likelihood.

## 2.2 Determining the Camera Pose

Having found a set of hypothesised correspondences for each model edge, a second level of RANSAC is used to find a new camera pose which maximises consensus of the original edgels. Three model edges are selected at random and for each a corresponding image line is chosen according to (9). This is usually sufficient to constrain a new hypothesised pose. It is straightforward to obtain a Jacobian matrix which describes the rate of change of each line parameter, $r_i$, with respect to each pose parameter, $\alpha_j$, about the current pose $\underline{\alpha}$. Hence a linear approximation to the new hypothesised pose is given by:

$$\underline{\hat{\alpha}} = \underline{\alpha} + J^{-1}\underline{r} \quad \text{where} \quad J_{i,j} = \frac{\partial r_i}{\partial \alpha_j} \ \ (i = 1..6,\ \ j = 1..6) \ . \tag{10}$$

The consensus score for the second stage RANSAC is found using a full Jacobian matrix to give a linear prediction for the new parameters of all the model lines. The probability density of each new line is evaluated from the original edgels using (8) and the product of these then gives the overall pose posterior. The second stage RANSAC is performed $L$ times and the pose with the highest posterior is retained as the estimate of the camera pose for that frame.

# 3 Finding Texture Change-points in a 1D Line Search

Traditionally (e.g. [9, 10]) an edgel is defined as a point of maximal intensity gradient along a 1D scanline. Here however we instead consider an edgel as a change-point in a 1D texture process. In our recent work [13] we described a detector which finds the single most probable location of a texture change. Here this is extended to allow multiple possible change-points. This allows the tracking system to additionally operate in simply textured environments yet adds little extra computational cost.

In [13] we proposed a tracking system similar to [9] but used a 1D texture change-point search instead of a simple gradient based edgel detector. Pixel intensities are first grouped into regularly spaced intensity bins. Consider a known zeroth-order texture generating process $T_1$ where pixel intensities are independently drawn from a probability distribution over $I$ intensity bins ($T_1 = \{p_i\}; i = 1..I$). Hence given a sequence of $N$ binned pixel intensities $S_0^N = (s_0, s_1, ...s_{N-1})$, the probability of obtaining that sequence, given process $T_1$ is:

$$P(S_0^N|T_1) = \prod_{n=0}^{N-1} p_{s_n} \ \ , \tag{11}$$

The key result from [13] is that if the texture process is instead unknown, but all possible texture processes are equally likely (i.e. there is a uniform prior on $T$), then given a sequence of $n$ pixel intensities $S_0^n = (s_0, s_1, ...s_{n-1})$ from $T$, the expected probability of a further sample, $s_n$ from $T$ is given by:

$$E(p_{s_n}|S_0^n) = \frac{o_{s_n} + 1}{n + I} \ \ , \tag{12}$$

where there are $o_j$ occurrences of the symbol $j$ in the sequence $S_0^n$. Hence the probability of the entire sequence given a single texture is:

$$P(S_0^N) = \prod_{i=0}^{N-1} E(p_{s_i}|S_0^i) \quad = \frac{(I-1)! \ \prod_{i=1}^{I}(o_i + 1)!}{(I + N - 1)!} \ \ , \tag{13}$$

In [13] this result was used to find the most probable location for a single texture change-point along a scanline and an efficient algorithm for this was presented. In this work however, multiple change-points must be considered. Instead of maximising the probability of the location of the single change-point given a sequence of intensities, we must find the set of change-points with maximum probability. If the sequence of $N$ pixels, $S$, is modelled by $m-1$ distinct textures between $m$ change-points, $M = (c_1, c_2 ... c_m)$:

$$P(S|M) = \prod_{j=1}^{m} P(S_{c_j}^{c_{j+1}}) \quad . \tag{14}$$

Finally using Bayes' rule we maximise:

$$P(M|S) = \frac{P(S|M)P(M)}{K} \quad . \tag{15}$$

For this work $P(M)$ was chosen simply as $P(M) = \lambda^m$, with $\lambda$ constant and $< 1$.

Evaluating (15) for all $2^N$ possible combinations of change-points would be computationally prohibitive. Fortunately it is possible to use a dynamic programming algorithm to improve efficiency greatly. The most probable set of change-points can be determined by considering the location of the last change-point in the sequence, $c_m$. Suppose that $\underset{M'}{\operatorname{argmax}} \; P(S_1^k|M')$ is known for all $k < l$. The most probable position of $c_m$ is then:

$$\begin{aligned} c_m &= \underset{c}{\operatorname{argmax}} \; P(S_0^l|\text{last change-point at } c) \\ &= \underset{c}{\operatorname{argmax}} \left[ \max_{M'} P(S_0^c|M') P(S_c^l|\emptyset) \right] \quad , \end{aligned} \tag{16}$$

since the probability of the sequence after $c_m$ is independent of the sequence up to $c_m$. $\emptyset$ is the empty set implying no change-points. Hence:

$$\begin{aligned} \underset{M}{\operatorname{argmax}} \; P(S_0^l|M) &= \{l\} \cup \underset{M'}{\operatorname{argmax}} P(S_0^{c_m}|M') \\ \text{and} \qquad \max_{M} \; P(S_0^l|M) &= \max_{M'} P(S_0^{c_m}|M') P(S_{c_m}^l|\emptyset) \quad . \end{aligned} \tag{17}$$

By induction these expressions can be evaluated sequentially for $l = 0..N$.

This means that, for $l = 0$, $P(S_0^0|0,1)$ is evaluated using $s_0$. At the second step both $P(S_0^1|0,2)$ and $P(S_0^1|0,1,2)$ must be evaluated using the value from the first step and $s_1$. Using these values, there are four potential sets of change-points to evaluate at the third step. However, the change from $P(S_0^1|0,2)$ to $P(S_0^2|0,2,3)$ is the same as that from $P(S_0^1|0,1,2)$ to $P(S_0^2|0,1,2,3)$ since by (14), the probability of the sequence after a change-point at 2 is independent of the sequence before it. Hence the most probable of $P(S_0^1|0,2)$ and $P(S_0^1|0,1,2)$ will lead to the most probable of $P(S_0^2|0,2,3)$ and $P(S_0^2|0,1,2,3)$. So at the third step we can simply pick the highest probability from the second step, and only evaluate three new probabilities. This means that when processing the $n^{\text{th}}$ pixel we must only evaluate $n$ possibilities, giving a total of $N^2/2$ operations for the entire sequence. The complete operation is shown in Algorithm 1.

Whilst this is not a major computational load, a second very simple speedup is possible. Since (15) will never be maximal if there is a change-point between pixels of the same binned intensity, there is no need to consider such cases. This offers a massive speed improvement in areas of the image which have a relatively constant intensity.

**Algorithm 1** Finding texture change-points along a 1D scanline

```
dim Observations[NUM_PIXELS][NUM_BINS]
dim LogLikelihoods[NUM_PIXELS], Changepoints[NUM_PIXELS]
BestLogLikelihood=0

for i=1..NUM_PIXELS do    // Loop for each pixel in the sequence
  for j=1..NUM_BINS do    // Set up a new node with a cleared histogram
    Observations[i][j]=1    // Seed with 1 sample per bin
  end for
  // and start with the best log likelihood to that change-point plus prior 'penalty'
  LogLikelihoods[i]=BestLogLikelihood-log($\lambda$)

  BestLogLikelihood=BIG_NUM
  for k=1..i do    // Loop round the nodes
    // update the node's log likelihood given the new pixel using (12)
    LogLikelihoods[k]+=log($\sum$ Observations[k][])-log(Observations[k][Pixels[i]])
    Observations[k][Pixels[i]]+=1
    // and store the best ready for the next outer loop
    if LogLikelihoods[k] $<$ BestLogLikelihood then
      BestLogLikelihood=LogLikelihoods[k]
      Changepoints[i]=k
    end if
  end for
end for

// The optimal changepoints can now be found by working back:
// $c_1$=Changepoints[NUM_PIXELS], $c_2$=Changepoints[$c_1$], etc.
```

# 4 Results

Figure 1 shows a simple example of successful tracking which demonstrates several benefits. Firstly, the system is able to track in the presence of simple textures, such as carpets. Secondly, the multi-modal posterior representation means that the system is able to consider multiple correspondences. In Figure 1, the chair produces a stronger edge than the carpet tile but, since both edges appear in the multi-modal posterior, the system sticks to the correct edge since this provides a better overall match. Finally, the scheme is relatively tolerant to occlusion. Although both the chair and the wastebasket in Figure 1 are unmodelled clutter, the system (given a large enough $L$) should find the correct pose as three of the modelled edges are visible in the image. Of course, clutter will reduce the overall pose posterior score, but typically the best score will be the correct pose.

The system was also tested on two full video sequences, one of a staircase and one of a corridor. Although both sequences were prerecorded, processing was performed at 10fps and 30fps respectively (on a single processor 2.4GHz Pentium) and selected frames are shown in Figure 3. In both cases the predicted location for each frame was simply that found in the previous frame. The staircase sequence demonstrates the system's ability to cope with textured surfaces (as shown in Figure 4) as well as rapid motion (leading to large inter-frame changes) and significant orientation changes. Tracking fails almost instantly if a simple gradient based edge detector is used instead of the scheme proposed

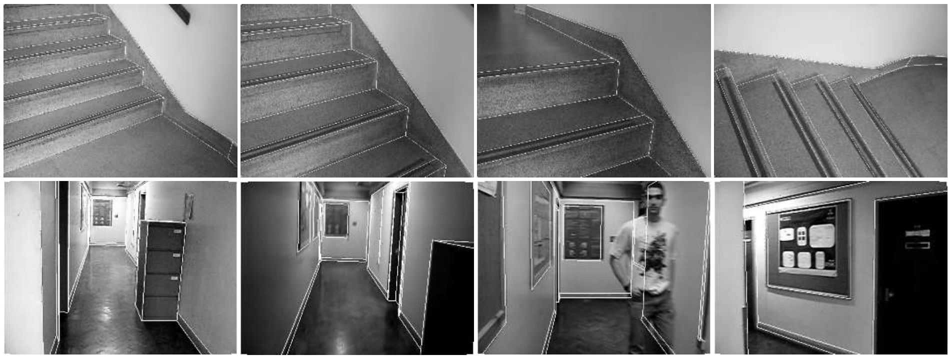

Figure 3: Frames 1,50,100,150 from a staircase video and 1,200,400,600 from a corridor sequence.

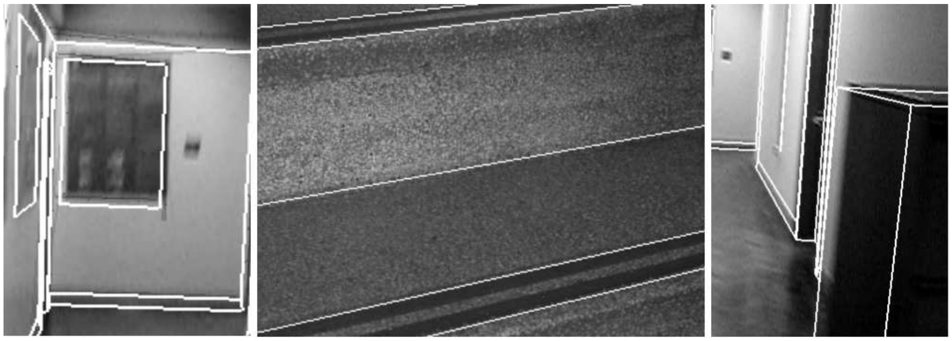

Figure 4: Left, a closeup from frame 200 corridor showing correct edge finding, despite significant image blur. Middle, a closeup from frame 100 of staircase showing edges being found on a textured surface. Right, an example of our single hypothesis system falling into an incorrect local minimum.

in Section 3 since the speckled steps produce a huge number of hypothesis edges. The corridor sequence uses relatively poor video with significant pixel noise and blurring and again, robustness to occlusion is demonstrated.

The corridor sequence was also tested on two other tracking systems, a demonstration version of ‘boujou’ and our earlier edge based system [7]. ‘boujou’ is a commercial product (`www.2d3.com`) which performs offline bundle adjustment of tracked interest points. The blur and pixel noise present in the images means that point features are difficult to match and tracking was regularly lost.

The corridor sequence also contains several times at which multiple edges align, an example of a local minimum which often causes single hypothesis systems to fail. Our earlier edge based system suffered from these failures (Figure 4) but the ability of the system proposed here to consider multiple hypotheses allows tracking to continue correctly.

# 5 Conclusions

This paper has presented a method for generating a representation of a multi-modal distribution. This is particularly beneficial when partitioning a problem such that the multiple hypotheses from one stage provide samples for further processing. The method has been successfully applied to the task of locating straight edges in an image using the results from a novel texture change-point detector and subsequently to perform visual tracking of known polyhedral objects. The system shows a significant improvement in motion robustness over previous techniques. Additionally, due to the data driven approach, it can be used to track all six degrees of freedom in real time.

# References

[1] M. Armstrong and A. Zisserman. Robust object tracking. In *Proc. Asian Conference on Computer Vision*, volume I, pages 58–61, 1995.

[2] T.J. Cham and J.M. Rehg. A multiple hypothesis approach to figure tracking. *Computer Vision and Pattern Recognition*, pages 239–245, June 1999.

[3] I. Cox and S. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(2):138–150, Feb. 1996.

[4] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. 9th Intl. Conference on Computer Vision*, pages 1403–1401, Oct. 2003.

[5] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. Intl. Conference on Computer Vision and Pattern Recognition*, volume 2, pages 126–133, 2000.

[6] J. Dittrich. Design and integration of an unmanned aerial vehicle navigation system. Master's thesis, Georgia Institute of Technology, Atlanta, GA 30332, July 2002.

[7] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):932–946, July 2002.

[8] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communcations of the A.C.M.*, 24(6):381–395, June 1981.

[9] C. Harris and C. Stennett. RAPID - a video rate object tracker. In *Proc. British Machine Vision Conference*, pages 73–77, 1990.

[10] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *Intl. Journal of Computer Vision*, 29(1):5–28, 1998.

[11] D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *Intl. Journal of Computer Vision*, 8(2):113–122, August 1992.

[12] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2D-3D model-based approach. In *Proc. Intl. Conference on Computer Vision*, volume 1, pages 262–268, Sept. 1999.

[13] A. Shahrokni, T. Drummond, and P. Fua. Texture boundary detection for real-time tracking. In *Proc. European Conference on Computer Vision*, volume 2, pages 566–577, 2004.

[14] G. Simon, A.W. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proc. Intl. Symposium on Augmented Reality*, pages 120–128, Oct. 2000.

[15] P.H.S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[16] L. Vacchetti, V. Lepetit, and P. Fua. Fusing online and offline information for stable 3D tracking in real-time. In *Proc. Intl. Conference on Computer Vision and Pattern Recognition*, volume 2, pages 241–248, June 2003.