Statistical Mosaics For Tracking

Simon Rowe[†] and Andrew Blake
Department of Engineering Science, University of Oxford
Parks Road, OX1 3PJ

Abstract

A method of robust feature-detection is proposed for visual tracking with a pan-tilt head. Even with good foreground models, the tracking process is liable to be disrupted by strong features in the background. Previous researchers have shown that the disruption can be somewhat suppressed by the use of image-subtraction. Building on this idea, a more powerful statistical model of background intensity is proposed in which a Gaussian mixture distribution is fitted to each of the pixels on a "virtual" image plane. A fitting algorithm of the "Expectation-Maximisation" type proves to be particularly effective here. Practical tests with contour tracking show marked improvement over image subtraction methods. Since the burden of computation is off-line, the online tracking process can run in real-time, at video field-rate.

Introduction

This paper presents a statistical treatment of background modelling for use in visual curve trackers. The new methods are tested using a real-time tracker based on snakes deforming over time [18, 10, 3], represented by B-spline curves [22, 8]. The tracker runs at video field rate (50Hz) and is stabilised using a shape template [14, 16, 5, 30] incorporated into the dynamical model used as a predictor. It runs at video field-rate (50Hz) in a cycle of prediction and measurement. The background modelling technique described here is not restricted to curves; it could also be applied to real-time trackers based on polygons or other geometrical representations [27, 17, 20]. Some tracking applications, surveillance for instance, call for a panoramic field of view which can be achieved by a pan-tilt head [9, 26, 25, 7]. Such a head is used in the experiments reported here.

[0]†The first author now works for Canon Research Europe Limited, Surrey Research Park, Guildford, UK, GU2 5YF

A major problem in achieving robust object tracking is the distracting effect of background objects—clutter. Strong features in the background compete for the attention of the tracked curve and may eventually succeed in pulling it away from the foreground object. This effect is clearly visible in figures 1a)–d) and 2a)–d). Immunity to distraction can be enhanced by both by modelling of the foreground and of the background. A foreground model may include a template, object dynamics [29, 12] and intensity profiles for certain object features [30, 11]. Such measures reduce the likelihood of erroneous matches of all or part of the tracked curve to spurious background features. The likelihood of such error can be reduced still further in the case that foreground events happen against a stationary background. Then the background also can be modelled and features which appear to match the background model can be ignored by the tracker.

This paper deals with modelling the background. It develops a statistical model of the distribution of intensities at *each* point in the background, which can then be used to discriminate the foreground object from the background. The model is applied to an image stream taken from a video camera mounted on a pan-tilt head—a situation where the previously used technique of image differencing proves ineffective.

It should be noted that there may be slight differences between the image sequences used to present results with the moving head in this paper, this is due to these tests being performed on *live* data. Ideally, recorded data would be used as standard tests but we currently have no facility for recording video *together with* head position data. Hence recorded benchmarks are currently only possible with a static camera.

The tracking process

Curve tracker

Our test task of curve tracking follows the method of Blake *et al* [6] and consists of a quadratic B-spline curve curve $(x(s), y(s))$ stabilised by a template curve $(\overline{x}(s), \overline{y}(s))$. Limited shape deformations of $(x(s), y(s))$ are allowed relative to the template and 2D Euclidean transformations are allowed to occur over time relatively freely. These dynamical constraints are used in the predictor of a Kalman filter [15] which constitutes the curve tracker.

The tracker is driven by a measurement process in which normal vectors to the tracked curve are constructed. The tradational tracker, performs one dimensional edge-searches along normals attempt to locate contrast features on the foreground object. When a candidate feature possessing plausible contrast is found, its position is added into the curve's estimated position and shape. It is at this juncture that background features may accidentally achieve a match and distract the tracking curve. This paper suggests

an approach whereby the edge-detector is replaced by a statistical test to determine if each point on each searchline is foreground or background. The boundary between the foreground region and the background is then used as the *feature* in the measurement process.

The virtual camera

The virtual camera is a single mathematical plane fixed in the world-frame onto which a physical image can be projected, in a manner somewhat akin to the recently developed technique of "image mosaicing" [28]. Ideally the centre of projection of the camera should coincide with the centre of rotation of the camera-head. In that case, for a given pan/tilt position, image pixels are projected along rays passing through the centre of projection, from the physical image onto the virtual one. This is illustrated in figure 3a. Note that a single virtual plane is sufficient where the union of all physical fields of view is contained within a hemisphere (otherwise several planes are required, forming a chart for the sphere). In practice there is some small misalignment of the two centres so that the projection process involves parallax errors, typically of a few mrad. The result is a panoramic image on the virtual image plane in which the parallax errors appear as blur, and this is shown in figure 4. The crucial point is that the (mean) image is accompanied by an overlaid probability distribution. In the simplest case this is a map of the *variance* of intensity, as shown in figure 4c.

The curve tracker now runs on the virtual rather than the physical image and this allows tracking to continue as if on a camera with a very wide field of view, but with the advantage of high resolution. Working in virtual camera coordinates means that the tracking process is quite decoupled from the effects of pan and tilt. In fact the controller for the position of the pan-tilt head can be quite slow and inaccurate, provided it is just agile enough to retain the foreground object within the field of view of the physical camera. A standard "Proportional-Integral-Derivative" (PID) controller [2] is quite sufficient. The head itself may then have substantial tracking lag, but this has no effect whatever on the curve tracker because the mapping from the physical to the virtual plane is computed using positional feedback signals directly from encoders on the motor shafts. Of course these encoders must be sufficiently fast and accurate but in practice such devices are routinely available. (Note that the physical camera must be calibrated, at least approximately, relative to the head.) This arrangement parallels the situation in animal vision in which slow head movements can be compensated by good proprioception, via the "vestibulo-ocular reflex" [1].

Background intensity variations

A number of researchers have used "image-differencing" to increase the robustness of tracking [23, 4, 19]. This uses a simple model of the background in which its mean intensity is represented as an image. Off-line estimation of this mean can be made robust to occasional moving objects by using a suitable filter — the median filter for instance. Once the mean image is obtained it is stored for repeated on-line subtraction from images in the incoming stream. This tends to cancel out background features, leaving features on moving foreground objects prominently exposed. A global threshold is applied to this differenced image to determine whether a point is part of a foreground object, or part of the background. Unfortunately, simple image differencing and thresholding has a somewhat limited power for rejection of spurious background clutter. This limitation is even more severe when there is additional variation introduced by viewing from a moving camera. The limitations of a simple scheme like this are shown below in figure 5. Murray and Basu[24] have managed to use image differencing with a moving camera, by applying a morphological opening filter to the thresholded difference image. This removes the small spurious features (such as edges) from the image—at a large computation cost. Their system does not run in real time—a necessary requirement for accurate tracking with a pan and tilt head—but off stored images.

In order to develop a system to discriminate foreground from background by using a model of the background, it is useful to think about the sources of variability in intensities of the background points. These sources include: variation of illumination over time, shadows and inter-reflection generated by moving foreground objects, parallax errors in mapping between physical and virtual image planes, mapping errors arising from any residual uncalibrated projective distortion of the physical camera's rectangular array as it appears on the virtual camera's rectangular array, mapping error due to the sub-sampling of the virtual plane needed to reduce physical memory requirements and sensor noise and spatial inhomogeneity of the camera array.

In some cases, illumination variation for example, partial compensation for error is possible, leaving only a residual uncompensated error to be modelled statistically. In other cases such as parallax error, the entire error is accounted for by the statistical model. It is not assumed that the errors in intensity are small—consider shadow-casting for instance, (see figure 1). Given that the system must in any case tolerate these gross errors, the pressure is removed for accurate camera calibration of the head/camera. Approximate calibration is sufficient since any residual error is relatively small and is comfortably absorbed into the statistical model.

In our system, the pan-offset is $< 0.02m$ and the tilt-offset is $< 0.1m$. This means

that the maximum error due to parallax is within ±2 pixels in the X axis, and within ±5 pixels in the Y axis[1].

Fitting to a normal distribution

The simplest reasonable model of the intensity variation at a single pixel is a univariate normal distribution. Given a training set consisting of a set of N readings $\mathbf{z} = [z_1, z_2 ... z_N]$. The mean μ and variances σ^2 are given by $\mu = \frac{1}{N}\sum_{i=1}^{\mathbf{N}} z_i$ and $\sigma^2 = \frac{1}{N-1}\sum_{\mathbf{i=1}}^{\mathbf{N}}(z_i - \mu)^2$.

In many areas of the image the univariate normal is adequate but the data is contaminated by foreground objects moving during data-collection, as in figure 6a, and this calls for a fitting method that is robust to outliers. The centre of the Gaussian μ can be located as the mode of the data distribution and an initial estimate of its standard deviation obtained as the width at half the modal frequency. With that initial estimate of σ, the relative proportions $\alpha, 1-\alpha$ of contaminant lying respectively to the left and right of the Gaussian peak can be estimated:

$$N_{lower} = |\{x \,|x \in \mathbf{z} \wedge x < (\mu - 3\sigma)\}| \quad (1)$$

$$N_{higher} = |\{x \,|x \in \mathbf{z} \wedge x > (\mu + 3\sigma)\}| \quad (2)$$

$$\alpha = \frac{N_{lower}}{N_{lower} + N_{higher}} \quad (3)$$

A variable proportion η of the data can be then be trimmed, $\alpha\eta$ from the left tail and $(1-\alpha)\eta$ from the right tail. As the trim-level varies, a χ^2 test detects when the remaining data cannot be distinguished from an uncontaminated Gaussian (The χ^2 test is a statistical test which tests the goodness of fit of a set of data).

Unfortunately, the trimming removes not only the contaminating dataset, but also the *tails* of the Gaussian. This will mean that equations given previously underestimate the Gaussian's variance. A solution to this problem is to use an Expectation-Maximisation (EM) algorithm which corrects the tendency to underestimate iteratively.

Using an EM Method

Expectation-Maximisation [13] is a technique for obtaining a maximum likelihood estimate (MLE) of a family of model parameters given some observed data. It is essentially an iterative two stage technique. In stage one, the **E**xpectation step, sufficient statistics are estimated based on the observed data. In stage two, the **M**aximisation step, takes this estimate of the sufficient statistics and estimates the model parameters by maximum likelihood as though complete data were observed. A more complete explanation of the general EM algorithm is given by Dempster *et al*[13].

[1] A more detailed analysis of parallax error is presented in [21].

The derivation of an EM estimation scheme for fitting a single Gaussian is not presented here, instead it can be thought of as a special case of the two Gaussian EM algorithm described in the appendix, where the probability of one of the Gaussian's is 0. Using this assumption leads to the update equations:

$$\mu_{i+1} = \frac{1}{N(1+q)}\left(\sum_{n=1}^{N} x_n + Nq\mu_i\right) \quad (4)$$

$$\sigma^2_{i+1} = \frac{1}{N(1+q)}\left(\sum_{n=1}^{N}(x_n - \mu_{i+1})^2 + Nq(\sigma_i^2 + \mu^2_{i+1})\right) \quad (5)$$

where μ_i is the i'th estimate for the mean of the distribution, and σ_i^2 for its variance. The dataset consists of N measurements of intensity $x_1 ... x_N$, and q is a scale factor defined in the appendix.

The iterative application of these equations will converge [13] onto an unbiased, MLE of μ and σ for the gray level distribution for a point. In order to have fast convergence to the correct answer in the presence of clutter, it is essential to have a good initial estimate of both μ and σ. Such an estimate could be obtained by using the repetitive trimming technique described in section . The improvement obtainable using this EM algorithm over the standard estimation technique when fitting a single Gaussian distribution in the presence of clutter is shown in figure 6.

Kalman Filtering

It has been proposed [19] that a Kalman Filter be used to track the intensity of a point over time, to derive a filtered estimate of mean intensity. An attraction of this approach is that it can be applied iteratively, online, giving a natural means of continuously updating the background distribution. Unfortunately it does not extend naturally to the statistical framework proposed here, in which not only mean but also variance of intensity is required.

Of course, the Kalman filter does generate a time-varying estimate of variance, but this is the variance of the *estimated mean* not the variance of the raw intensity process itself. Indeed, to underline this distinction, note that the steady state variance of the mean P_{ss} is given [15] by:

$$P_{ss} = \frac{1}{2}\left(-Q + \sqrt{Q^2 + 4QR}\right) \quad (6)$$

where the model noise has assumed covariance Q, and the measurement noise covariance is assumed to be R. (The dynamical model here specifies constant intensity.) This means that P_{ss} converges to a fixed value, depending only on prior assumptions (Q, R) and quite *independent of the data*. Thus P_{ss} cannot possibly be informative as to the nature of the underlying intensity distribution.

Limitations of the single Gaussian model

The greater part of the *area* of the image is adequately modelled by the single (contaminated) Gaussian. Unfortunately that minority of pixels near clutter edges needing modelling by a Gaussian mixture, are *exactly* the pixels that cause false tracking. Modelling these pixels accurately is crucial to good tracking performance. Although the single Gaussian model fails to do justice to the underlying distribution near high-contrast edges, as figure 7 shows, a two-Gaussian mixture would appear to be adequate. Either the trimming technique or the single Gaussian EM algorithm might be expected, at best, to converge to one of the Gaussian's. The remaining unmodelled Gaussian will generate false foreground features and cause the tracker to stick on the background clutter.

The single Gaussian model is also inadequate when the foreground and background interact with each other — when the target casts a shadow on the background for instance (as in figure 1). In this case points in the background can be expected to have two intensity distributions associated with them – one for direct illumination and one from the ambient illumination. This means that the PDF for the point will again comprise two separate Gaussian's.

Fitting a two-Gaussian mixture

The problem is to fit a two Gaussian mixture to data which is possibly contaminated by outliers. Both approaches mentioned in the previous section can be applied to this problem with only slight modification, the trim and fit method by applying it recursively, and the EM algorithm by re-formulating it for a two Gaussian mixture.

Unfortunately applying the *Trim and Fit* algorithm recursively would still suffer from the same problem as it does when applied to the single Gaussian case—it will produce an underestimate of the variance of the distributions. It also suffers from additional problems when two Gaussian's overlap, since it takes no account of their interactions. Both these problems can be eliminated by using an properly formulated EM algorithm. The appendix derives one such algorithm.

The use of a two Gaussian mixture model

It can be shown that using a single Gaussian model instead of a Gaussian mixture, leads to a tracker less sensitive than one based on the correct underlying model. This means that in situations where a large proportion of the image requires a two Gaussian mixture, a tracker which utilises the correct model will track significantly better than one using the one Gaussian model. This is shown below in figure 9, where the image has been

heavily sub-sampled (by a factor of 4×4). The insensitivity caused by using the wrong model means that a tracker based around fitting a single Gaussian to each points intensity distribution loses track of the target when it is subject to high accelerations. The tracker based on the two Gaussian model however manages to continue tracking the target even in these situations. Figure 8 is a key to this sequence.

Unfortunately, correctly fitting a two Gaussian model to points in the image takes a long time — the current implementation on a Sun Sparc IPX, takes of the order of 1 second for each point. In a static image of 768×512 pixels, sub-sampled by a factor of 4×4, there are still of the order of 25000 points, meaning that it will take approximately 7 hours to fit a two Gaussian model to them. Fortunately however, not all points need a two Gaussian mixture to represent their intensity distribution, in the example in figure 9 only about 8000 of the points appear[2] to require the more complex model, meaning that this model can be learnt in about 2.5hours. The single uncontaminated Gaussian model can be fitted to all the points in the image in about 1 minute. There may however be situations when this long start-up time is perfectly acceptable, such as a security camera looking down corridor night after night. Certainly as the computational speed of computers increases, this time will become acceptable for more and more cases.

A further problem when attempting to fit a background distribution both in direct lighting and in shadow is that in normal situations the shadow may only be present for a small, but highly significant, proportion of the time. This can make collecting representative background data difficult unless it is done by deliberately casting shadows onto the background without allowing the foreground object to appear in the image too often. The result of this is that the background modeller is forced to model the intensity variability due to shadows, but the foreground object appears only as a contaminant and is not modelled.

The improvement in tracking accuracy obtainable by using a two Gaussian mixture is shown in figure 1 — the contour tracks the hand rather than the shadow—an improvement that cannot be obtained by simple image-differencing.

Given the above arguments for sub-sampling the image, (namely that the resultant intensity distribution can still be modelled by a Gaussian mixture), it might at first appear that we could go increasing the sub-sampling factor indefinitely. While it is indeed true that we can model any intensity distribution by an N-Gaussian mixture[3], it is not true that

[2]Points *which appear* to need a two Gaussian model are defined as those points which, when fitted by a uncontaminated single Gaussian model, have an unusually large variance, typically one greater than about 15 gray levels. It is not a problem to attempt to fit a two Gaussian model to a single Gaussian distribution, as the EM algorithm will correctly deduce that the probability of one of the Gaussian's which it is attempting to model is zero.

[3]Since the intensities returned by our frame grabber are quantised in the range 0–256, any intensity distribution can be modelled by 256 Gaussian mixture, with each Gaussian centred on a different intensity,

this helps us to discriminate foreground from background. As the sub-sampling factor is increased (to take in a larger area of the background), the resultant intensity distribution will allow more of the range of possible intensities to be explained by the background model—as there are a larger range of intensities to be explained by the background model. This means that there is less discriminatory power left to recognise points which are not background—we loose the ability to recognise foreground points as not belonging to the background.

Modelling Gradient of Intensity

One obvious way to extend the intensity modelling described earlier in this paper, would be to model higher order features of the background than raw intensity, such as the gradient of intensity. Doing this would have advantages in terms of robustness against lighting changes.

The model fitting theory described earlier can be applied exactly as before with ΔI substitued for I. The histograms of gradient of intensity at each point in the background have very similar shapes to those of the raw intensities, compare figure 11 with figure 7.

Unfortunately however since the vast majority of both the background and foreground have very low image gradients, discriminatory information is only really available at image *edges*. This means that over most of the image, the only points on the target that can be discriminated from the background are the edges of the target. This is shown in figure 12((a)–(f)). Although these are the points that we need to track the object, if we only have information at these points then we will be very susceptible to noise in our detection process. This is in sharp contrast to the situation described in the previous section where almost the entire target was discriminated from the background, giving very good resistance against thresholding noise. The differences in the two methods are shown below in figure 12. These images show the points in the thresholded images which could be selected by the feature search mechanism of the active contour, when the image has been thresholded by intensity and gradient of intensity. It can be seen that when the target thresholded by intensity the *feature map* is much cleaner and more complete than that provided by the gradient threshold. This means that a tracker based on the model of background intensity will track correspondingly better than one based on a model of background gradient.

---

and having a variance of around 0.25gray levels. (The figure 0.25 gray levels was chosen here so that $\pm 2\sigma$ covers a range of 1 gray level).

Discussion

Extensions have been proposed to improve and extend the tracking ability of active contours so that they can successfully robustly track in a wider range of applications. Use of a virtual image-plane has been proposed, enabling an active contour tracker designed for a static camera to operate transparently with a pan/tilt head. Results have been shown for a hard tracking sequences which demonstrate the improvements in tracking performance possible by statistically modelling the distributions of points in the background.

Future work will address more efficient ways to fit the background model to the intensity distribution. An interesting possibility, worthy of investigation, is to extend the statistical modelling the background beyond modelling intensity to include also the gradients of the intensity field — both in space and in time.

References

[1] D.B. Arnold and D.A.. Robinson. A neural network model of the vestibulo-ocular reflex using a local synaptic learning rule. *Phil. Trans. R. Soc*, 337:327–330, 1992.

[2] K. J. Astrom and B. Wittenmark. *Computer Controlled Systems.* Addison Wesley, 1984.

[3] N. Ayache, I. Cohen, and I. Herlin. Medical image tracking. In A. Blake and A. Yuille, editors, *Active Vision*, pages 285–302. MIT, 1992.

[4] Adam Baumberg and David Hogg. Learning flexible models from image sequences. In Jan-Olof Eklundh, editor, *Computer Vision - ECCV '94*, volume Volume I, pages 299 – 308. Springer-Verlag, 1994.

[5] A. Bennett and I. Craw. Finding image features for deformable templates and detailed prior statistical knowledge. In P. Mowforth, editor, *Proc. British Machine Vision Conference*, pages 233–239, Glasgow, 1991. Springer-Verlag, London.

[6] A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *Int. Journal of Computer Vision*, 1993.

[7] C.M. Brown, D. Coombs, and J. Soong. Real-time smooth pursuit tracking. In A. Blake and A. Yuille, editors, *Active Vision*, pages 123–136. MIT, 1992.

[8] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 616–625, 1990.

[9] J.J. Clark and N.J. Ferrier. Modal control of an attentive vision system. In *Proc. 2nd Int. Conf. on Computer Vision*, pages 514–522, 1988.

[10] Laurent D. Cohen and Isaac Cohen. A finite element method applied to new active contour models and 3D reconstruction from cross sections. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 587–591. IEEE Computer Society Conference, December 1990. Osaka, Japan.

[11] T.F. Cootes, C.J. Taylor, A. Lanitis, D.H. Cooper, and J. Graham. Buiding and using flexible models incorporating grey-level information. In *Proc. 4th Int. Conf. on Computer Vision*, pages 242–246, 1993.

[12] R. Curwen, A. Blake, and A. Zisserman. Real-time visual tracking for surveillance and path planning. In *Computer Vision - ECCV '92*, pages 879–883, 1992.

[13] A.P. Dempster, M.N. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc.*, B 39:1–38, 1977.

[14] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE. Trans. Computers*, C-22(1), 1973.

[15] Arthur Gelb, editor. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.

[16] U. Grenander, Y. Chow, and D. M. Keenan. *HANDS. A Pattern Theoretical Study of Biological Shapes*. Springer-Verlag. New York, 1991.

[17] C. Harris. Tracking with rigid models. In A. Blake and A. Yuille, editors, *Active Vision*, pages 59–74. MIT, 1992.

[18] M. Kass, A. Witkin, and D. Terzopoulos. Snakes:active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, pages 259–268, 1987.

[19] Dieter Koller, Joseph Weber, and Jitendra Malik. Robust multiple car tracking with occlusion resoning. In Jan-Olof Eklundh, editor, *eccv94*, volume Volume I, pages 189 – 196. Springer-Verlag, 1994.

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

[21] P. F. McLauchlan and D. W. Murray. Active camera calibration for a head/eye platform using a varia ble state dimension filter. Oxford University Engineering Library report number OUEL 1975/93. Accepted for PAMI, 1993.

[22] S. Menet, P. Saint-Marc, and G. Medioni. B-snakes:implementation and application to stereo. In *Proceedings DARPA*, pages 720–726, 1990.

[23] M.Kilger. Video-based traffic monitoring. In *IEE 4th International Conference on Image Processing and its applications*, 1992.

[24] Don Murray and Anup Basu. Motion tracking with an active camera. *IEEE Trans. Pattern Analysis and Machine Intell.*, 16(5):449–459, May 1994.

[25] D.W. Murray, F. Du, P.F. McLauchlin, I.D. Reid, P.M. Sharkey, and M. Brady. Design of stereo heads. In A. Blake and A. Yuille, editors, *Active Vision*, pages 303–336. MIT, 1992.

[26] K Pahlavan and J-O Eklundh. A head-eye system for active purposive computer vision. Technical Report CVAP-80, Dept of Numerical Analysis and Computing Science, Royal Inst of Tech, Stockholm, 1991.

[27] G.D. Sullivan. Visual interpretation of known objects in constrained scenes. *Phil. Trans. R. Soc. Lond. B.*, 337:109–118, 1992.

[28] R. Szeliski. Image mosaicing for tele-reality applications. Technical report, Digital Equipment Corporation, Cambridge, USA, 1994.

[29] D. Terzopoulos and R. Szeliski. Tracking with Kalman snakes. In A. Blake and A. Yuille, editors, *Active Vision*, pages 3–20. MIT, 1992.

[30] A. Yuille and P. Hallinan. Deformable templates. In A. Blake and A. Yuille, editors, *Active Vision*, pages 20–38. MIT, 1992.

A Fitting a contaminated two Gaussian mixture using an EM algorithm

The problem we are faced with is to fit an unbiased model to a contaminated two Gaussian mixture. One class of algorithms suitable for this problem are the EM algorithms. These algorithms extend Maximum Likelihood Estimation (MLE) algorithms, allowing them to deal with missing or unseen data. They work by maximising the *Expectation* of the likelihood[4] of some observed (or seen) data, rather than by maximising the likelihood of this data directly as is done in MLE. It is usual to maximise the log-likelihood of the data as opposed to the likelihood, as it simplifies the mathematics and has maxima at the same place.

The probability distribution function (PDF) of a Gaussian with mean μ and variance σ^2 is defined:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{7}$$

The probability of a point x, having come from a distribution with mean μ_i, variance σ_i^2, when the probability of distribution i is P_i is given by $f_i(x)$, where:

$$f_i(x) = p_i(x)P_i \tag{8}$$

In the case of a two component Gaussian mixture, P_1 and P_2 are the mixing constants, and $P_2 = 1 - P_1$.

The data we observe can be thought of as coming from two datasets, X_1 and X_2, where the data in X_1 is within $\pm\lambda\sigma_1$ of μ_1 (and similarly for the data in X_2). These datasets can be treated separately, as long as we remember that data in X_1 might actually be associated with the Gaussian centred at μ_2. A diagrammatic explanation of the datasets used is given below in figure 13a.

Now, if all the data in X_1 actually originated from Gaussian 1, then the likelihood $\mathcal{L}$ that the dataset X_1 came from the Gaussian centred at μ_1 is given by

$$\mathcal{L}(X_1|\mu_1, \sigma_1) = \prod_{x \in X_1} p(x) \tag{9}$$

and, dropping a constant term, the log-likelihood, $L(X_1)$ is:

$$L(X_1) = -\sum_{x \in X_1} \left(\log\sigma_1^2 + \frac{1}{\sigma_1^2}(x-\mu_1)^2\right) \tag{10}$$

however, since each datum is unlabelled as to which distribution it comes from, this expression cannot be evaluated. We can however take the expectation of it:

$$E\left(L(X_1)|\Phi\right) = -\sum_{x \in X_1} \frac{f_1(x)}{f_1(x)+f_2(x)} \left(\log\sigma_1^2 + \frac{1}{\sigma_1^2}(x-\mu_1)^2\right) \tag{11}$$

where $\Phi = \{\mu_1, \sigma_1^2, \mu_2, \sigma_2^2, P_1\}$. We can develop a similar expression for $E\left(L(X_2)|\Phi\right)$. The quotient term arises because points in the distribution X_1, are not independent of Gaussian 2 (since we do not have labelling information about which Gaussian a particular intensity value is from).

We still have to deal with the missing tails of the Gaussian's. This is done by *hyphosising* data in the two tails. This hypothesising data can be thought of as having come from two

[4]The likelihood of a set of data, given a model is defined to be the product of the probability that each individual datum can be explained by the model.

datasets, Y_1 and Y_2, where Y_1 is associated with Gaussian 1, and Y_2 with Gaussian 2. Note that while Y_1 and Y_2 are independent of each other since they consist totally of hypothesising data, their ranges may overlap each other, as in figure 13c.

The log-likelihood of the distribution Y_1 coming from Gaussian 1 is simply given by:

$$L(Y_1) = -\sum_{y \in Y_1} \log \sigma_1^2 + \frac{1}{\sigma_1^2}(y - \mu_1)^2 \tag{12}$$

and similarly for the distribution Y_2 and Gaussian 2. The expectation of this is given by:

$$E[L(Y_1)|\Phi] = E\left[-\sum_{y \in Y_1} \log \sigma_1^2 + \frac{1}{\sigma_1^2}(y - \mu_1)^2\right] \tag{13}$$

Now, with a slight abuse of notation let us call the expectation of both the seen data(X_1, X_2) and unseen data (Y_1, Y_2), $E[L(\Phi)]$ where:

$$E[L(\Phi)] = E\left[L(X_1)|\Phi\right] + E\left[L(X_2)|\Phi\right] + E\left[L(Y_1)|\Phi\right] + E\left[L(Y_2)|\Phi\right] \tag{14}$$

In order to fit the data, we would like to find the set of parameters, Φ' which maximises this expectation of log-likelihood. This is done by setting each partial derivative of equation (14) to zero. Let us call the values of Φ which are our next estimate $\bar{\Phi} = \{\bar{\mu}_1, \bar{\sigma}_1^2, \bar{\mu}_2, \bar{\sigma}_2^2, \bar{P}_1\}$. Setting $\frac{\partial L(\bar{\Phi})}{\partial \bar{\mu}_1} = 0$ gives:

$$0 = \frac{\partial L(\bar{\Phi})}{\partial \bar{\mu}_1} = E\left[\frac{2}{\bar{\sigma}^2}\sum_{x \in X_1}(x - \bar{\mu}_1)\right] + E\left[\frac{2}{\bar{\sigma}_1^2}\sum_{y \in Y_1}(y_i - \bar{\mu}_1)\right] \tag{15}$$

$$0 = \frac{2}{\bar{\sigma}_1^2}E\left[\sum_{x \in X_1} x\right] - \frac{2}{\bar{\sigma}_1^2}E\left[\sum_{x \in X_1} \bar{\mu}_1\right] + \frac{2}{\bar{\sigma}_1^2}E\left[\sum_{y \in Y_1} y_i\right] - \frac{2}{\bar{\sigma}_1^2}E\left[\sum_{y \in Y_1} \bar{\mu}_1\right] \tag{16}$$

$$0 = \sum_{x \in X_1} E\left[x\right] - E\left[\sum_{x \in X_1} \bar{\mu}_1\right] + E\left[\sum_{y \in Y_1} y_i\right] - E\left[\sum_{y \in Y_1} \bar{\mu}_1\right] \tag{17}$$

Rearranging this gives

$$\bar{\mu}_1 E\left[\sum_{x \in X_1} 1\right] + \bar{\mu}_1 E\left[\sum_{y \in Y_1} 1\right] = \sum_{x \in X_1} E\left[x\right] + E\left[\sum_{y \in Y_1} y_i\right] \tag{18}$$

Noting that the probability of a point, x, in the dataset X_1, actually belonging to Gaussian 1 and not to Gaussian 2 is $\frac{f_1(x)}{f_1(x)+f_2(x)}$, and that we, therefor, effectively have N_1 actual points in X_1, and qN_1 points in Y_1, we get:

$$qN_1\bar{\mu}_1 + N_1\bar{\mu}_1 = \sum_{x \in X_1}\left(\frac{f_1(x)}{f_1(x) + f_2(x)}x\right) + qN_1\mu_1 \tag{19}$$

$$\text{And finally,} \quad \bar{\mu}_1 = \frac{qN_1\mu_1 + \sum_{x \in X_1}\left(\frac{f_1(x)}{f_1(x)+f_2(x)}x\right)}{(1+q)N_1} \tag{20}$$

$$\text{where} \quad N_1 = \sum_{x \in X_1}\frac{f_1(x)}{f_1(x) + f_2(x)} \tag{21}$$

$$q = \frac{p}{1-p}$$

$$p = 2(1 - erf(\lambda)) \tag{22}$$

Note that q is related to the area of the trimmed tails of the Gaussian, relative to the untrimmed area of the Gaussian.

Equating the other partial derivatives to zero leads to the following set of parameter update equations:

$$\bar{\mu}_2 = \frac{qN_2\mu_2 + \sum_{x\in X_2} \frac{f_2(x)}{f_1(x)+f_2(x)} x}{(1+q)N_2} \tag{23}$$

$$\bar{\sigma}_1^2 = \frac{\sum_{x\in X_1} \frac{f_1(x)}{f_1(x)+f_2(x)}(x-\bar{\mu}_1) + qN_1(\mu_1-\bar{\mu}_1)^2 + qN_1\sigma_1^2(1+2\frac{\lambda}{p\sqrt{2\pi}}e^{\frac{-\lambda^2}{2}})}{(1+q)N_1} \tag{24}$$

$$\bar{\sigma}_2^2 = \frac{\sum_{x\in X_2} \frac{f_2(x)}{f_1(x)+f_2(x)}(x-\bar{\mu}_2) + qN_2(\mu_2-\bar{\mu}_2)^2 + qN_2\sigma_2^2(1+2\frac{\lambda}{p\sqrt{2\pi}}e^{\frac{-\lambda^2}{2}})}{(1+q)N_2} \tag{25}$$

$$\bar{P} = \frac{N_1}{N_1+N_2} \tag{26}$$

The *best* estimate of Φ can then be found by iteratively applying the above equations and setting $\Phi = \bar{\Phi}$ at the end of each update loop.

The difference that applying this EM algorithm makes compared to the straight Gaussian fitting is shown below in figure 14. Note that the trim and fit algorithm fails almost totally to fit the distribution in this case as the two Gaussian's overlap very significantly. The EM algorithm however correctly identifies and fits the two components of the distribution.

Convergence of the EM algorithm

The EM algorithm appears to be fairly robust at fitting a two Gaussian mixture, provided that it is given an intelligent starting point. We have tested the EM algorithm with a wide variaty of input data and initial estimates. In all cases the algorithm converges quickly to very close to the right distribution, provided that it is given an intelligent first guess at the distributions. Adding 30% noise into the dataset affects the convergence very little—this is due to the EM algorithm only accepting data-values within $\pm\lambda\sigma$ of the estimated mean of the distributions (a value of $\lambda = 1.8$ was used in these experiments). If the algorithm is given a poor initial estimate of the distributions, unsurprisingly it fails to fit the distributions properly. Typically this fails when the initial guess totally misses one of the peaks in the distribution.

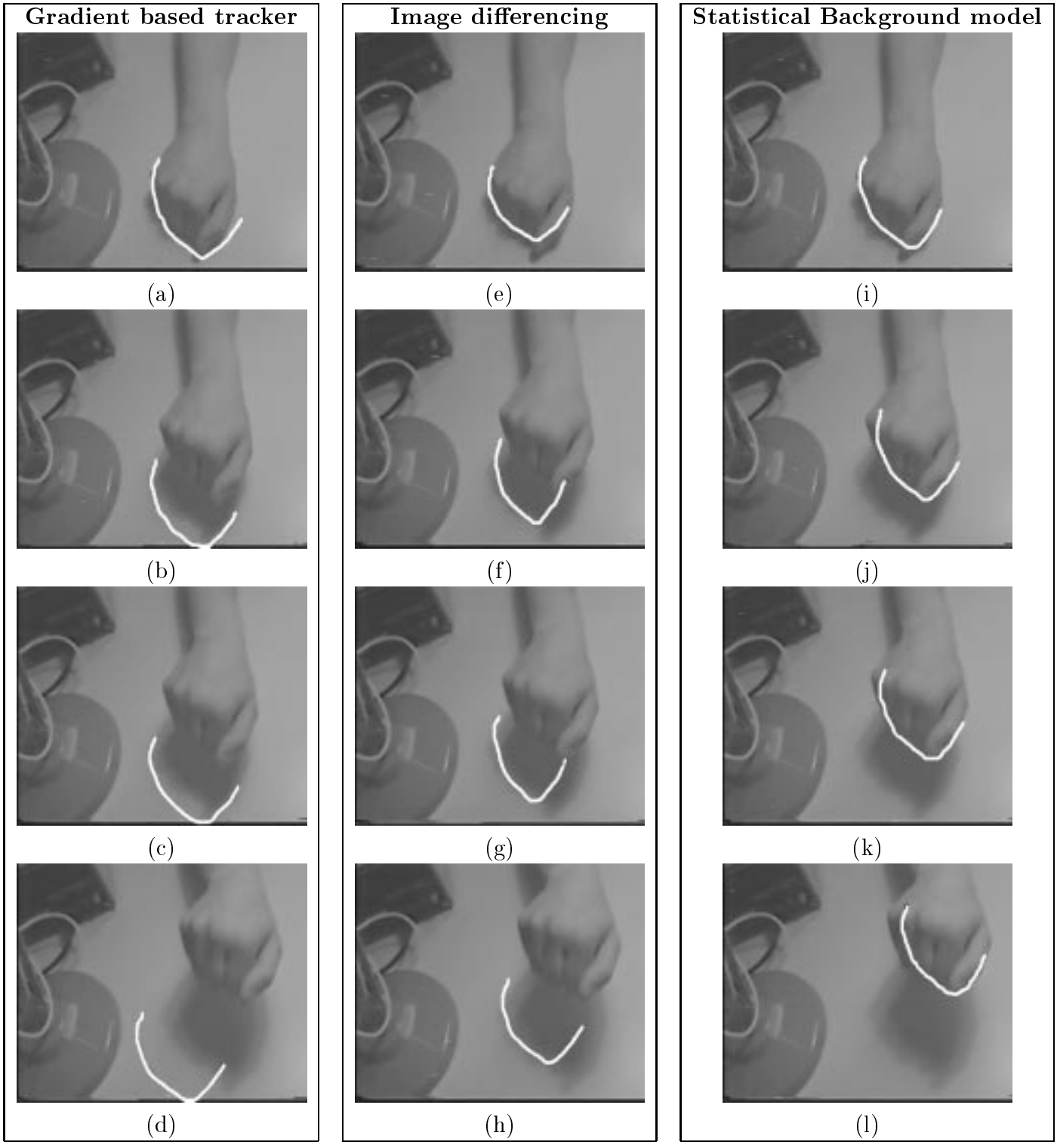


Figure 1: In the sequence shown in the left column ((a)-(d)), a hand is being tracked by a static camera. Since the edge of the shadow created by the hand has a stronger gradient than the edge of the hand, a gradient-based tracker tracks the shadow rather than the hand – eventually losing track of the hand (d). The middle column ((e)-(h)) shows a the same sequence but using image differencing to locate the target. The hand's shadow again distracts the tracker. The right-hand column ((i) - (l)), shows contour tracking with background modelling on the sequence. This allows the tracker to ignore both desk and shadow and to correctly track the hand. Tracking fails slightly on the left hand side of the hand due to this being a very low contrast image, and the shadow on the hand being almost exactly the same as the shadow on the desk.

Figure 2: In the left-hand sequence ((a) - (d)), a gradient-based feature detector is used to track a target as it moves across a room. The camera is mounted on a pan-tilt head. Because the foreground is fixated it appears stationary, but note how the background moves relative to the target. As the target passes some strong clutter the contour is distracted ((c) and (d)) and loses track of the target. The middle column((e)-(h)) shows a similar tracking sequence, but using image differencing. The contour is still distracted by the edges of objects in the background and loses track of the target (h). Finally, in the right hand column, the background has been modelled statistically on the virtual image plane. Edges in the background are ignored (k) enabling tracking to continue past the clutter (l).

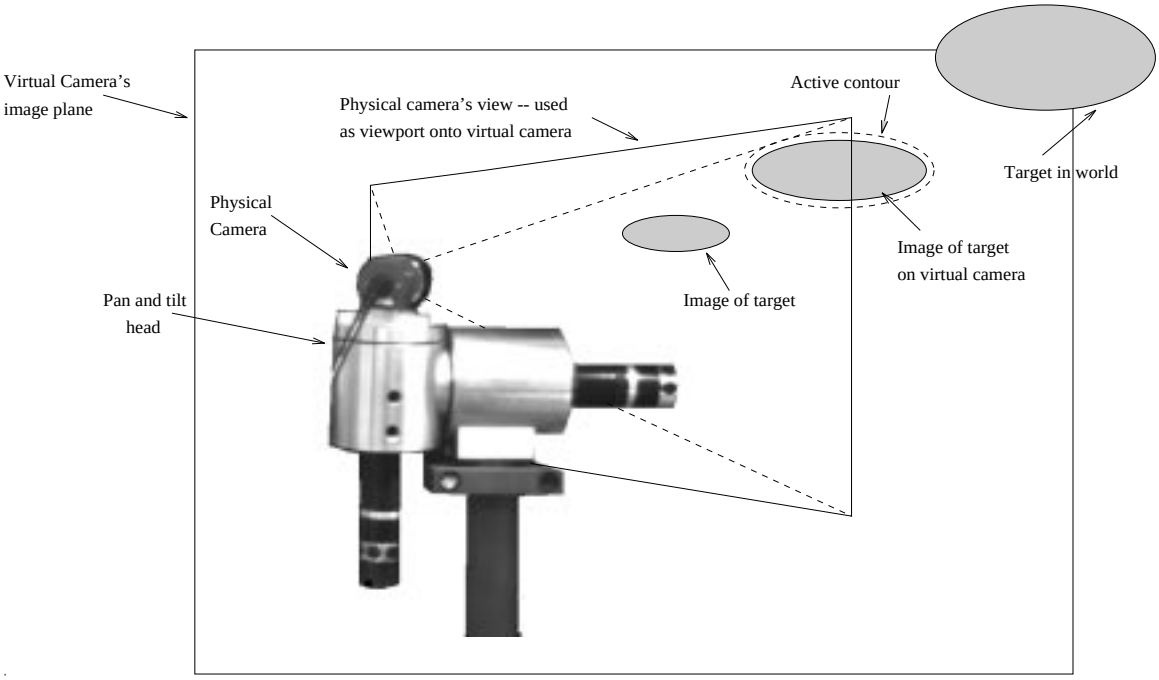
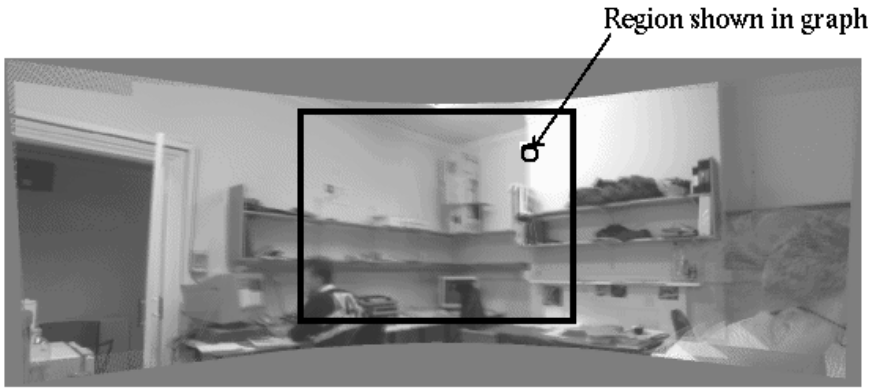
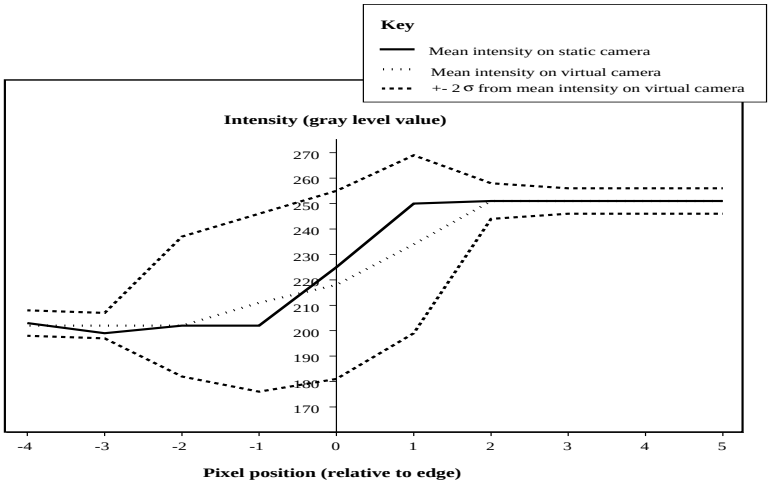


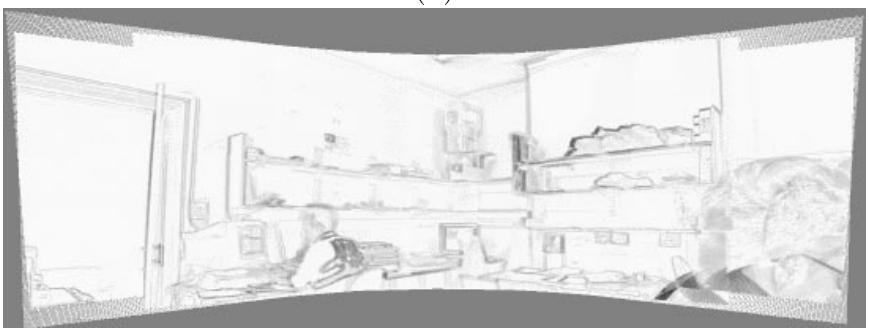
Figure 3: A target is viewed in the world using a real camera mounted on a pan and tilt head. The view from the physical camera is projected onto a static virtual camera plane where an active contour tracks a target.



(a)

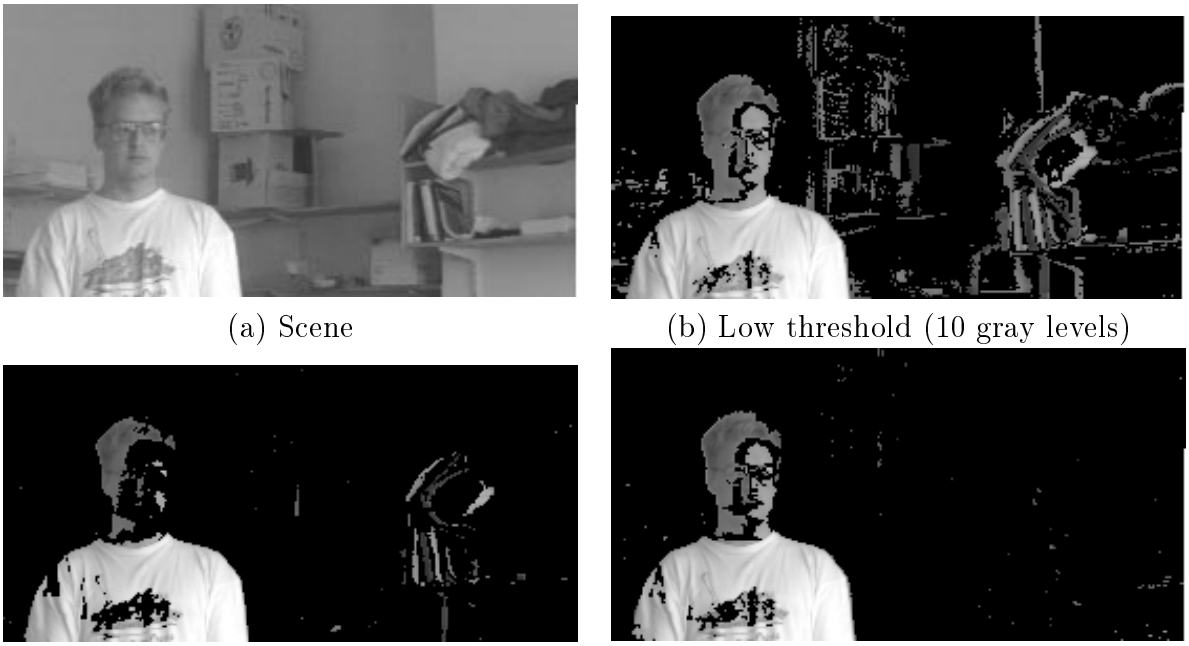


(b)



(c)

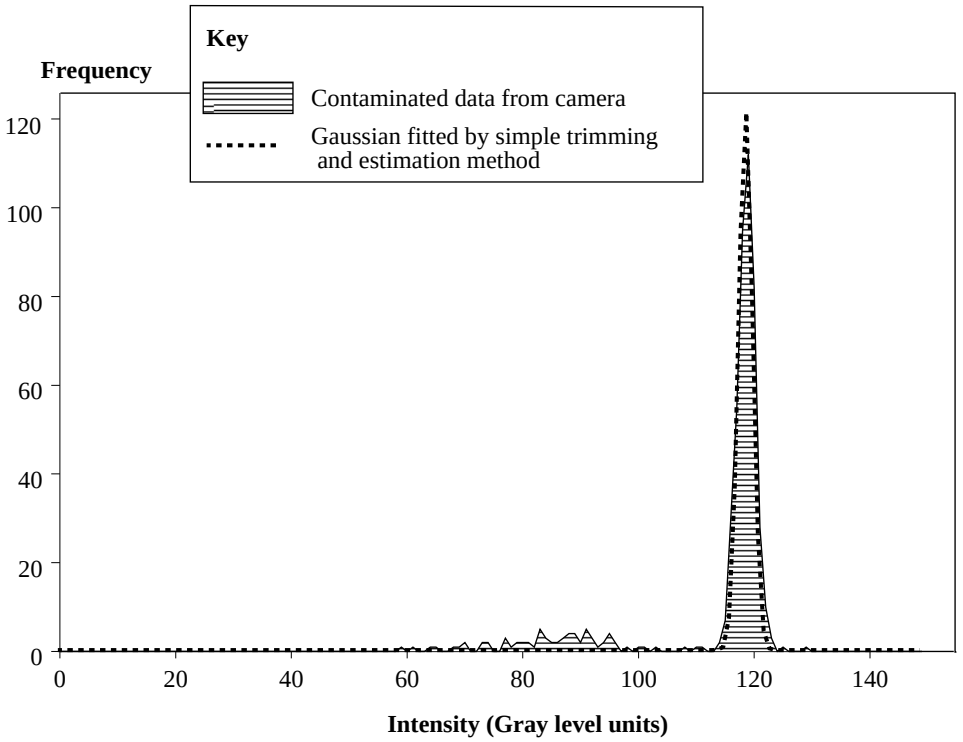
Figure 4: The virtual image plane. The image in (a) was obtained from a physical camera mounted on a pan and tilt head, mapping its image onto the virtual image plane as the it is swept round the room. The instantaneous field of view of the physical camera is shown as the black rectangle in the image. Calibration errors in the system mean that the image is slightly blurred. (b) shows the gray levels across the edge of the chimney on both the virtual image plane (which is the average of several views), and the physical image plane. It can be seen that the edge on the virtual image is more blurred (spread out) than the corresponding edge on the physical image. This blur is within the range that the mounting of the camera can be expected to produce ($\pm$2.2mrad). Although the apparent effect of blur is small, it is significant for background modelling because of the consequent variability of intensity I. The variance (c) of intensity over the virtual image is particularly great where ∇I is large (i.e. at edges).



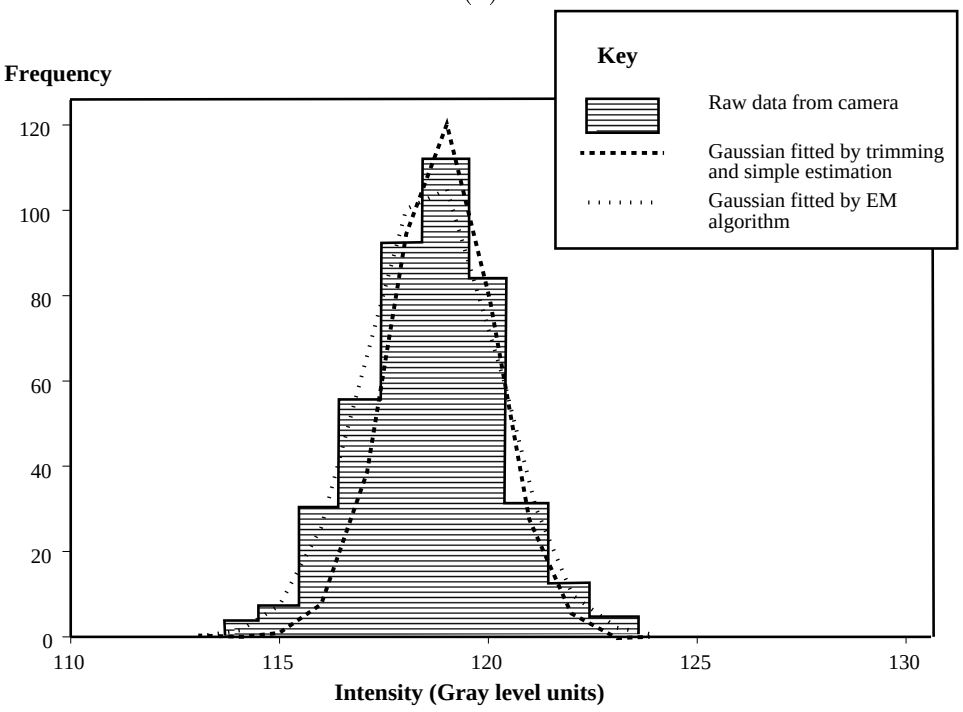
(a) Scene (b) Low threshold (10 gray levels)

(c) High threshold (27 gray levels) (d) Statistical background removal

Figure 5: **Statistical background removal outperforms simple thresholding.** Using a constant threshold to compare a scene containing a target (a) to the same scene without the target. The intensities from (a) have been used to show areas which differ by more than the threshold from the reference image. (b) shows the result of using image differencing with a low threshold, representing the intensity noise typical of the walls—note how clutter edges are segmented along with the target. (c) shows the result when a higher threshold is used so that most of the edges in the image are no longer segmented. Unfortunately this threshold results in a high level of drop out of the target, while still segmenting some edges along with the target. Setting the threshold even higher to remove these edges would mean that even more of the target is lost. Finally, (d) shows the result of segmenting using independent pixel based thresholds. These thresholds were set at $\pm 2\sigma$ of the Gaussian fitted to the pixel's intensity during the learning phase. The target is better segmented with this approach than with the image differencing approach.



(a)



(b)

Figure 6: Figure (a) shows the fitting of a Gaussian to the intensity distribution at a point, with contamination by clutter. The trimming/fitting of the dataset allows a Gaussian to be fitted to the background data; however note that the variance of the Gaussian has been underestimated. (b) shows an expanded view of a section of (a), with the fitted Gaussian obtained by the EM algorithm overlaid. Note that the Gaussian fitted by EM is a better approximation to the real data than the simply fitted one. Simple fitting gave a standard deviation of 1.175 for the main peak, whereas the EM algorithm gave 1.523 – a 30% increase.

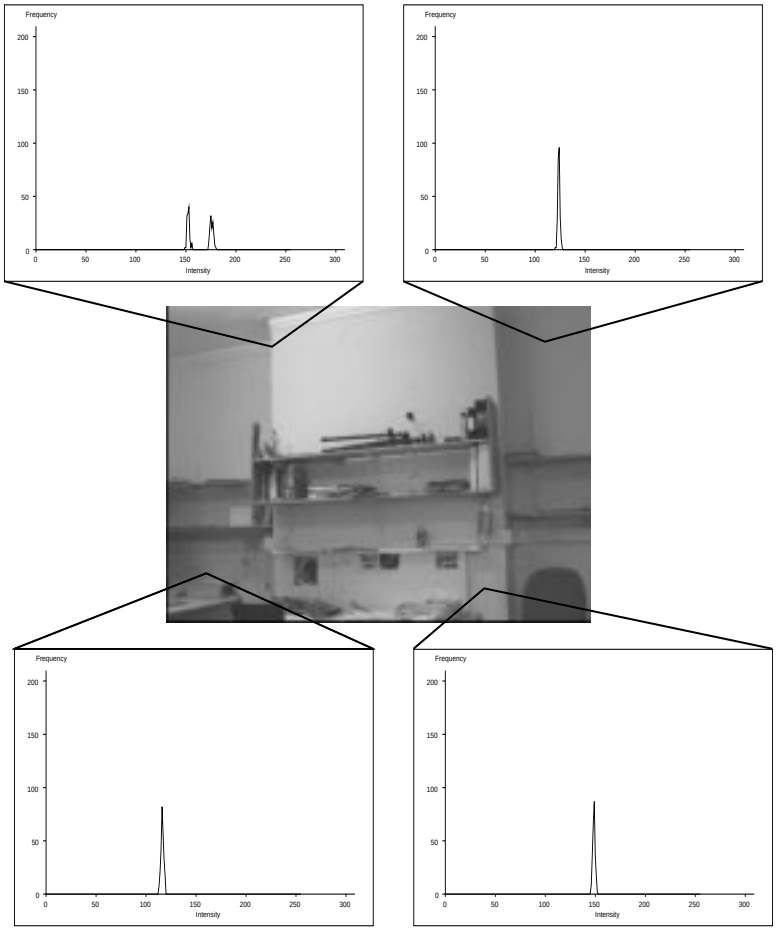


Figure 7: Intensity histograms for points on the image plane. The graphs show the frequency of occurrence for a particular gray level for a particular pixel over 500 frames. The image has been sub-sampled by mapping each 2×2 pixel block onto a single point. Near an edge a two Gaussian mixture will be necessary to model the intensity as can clearly be seen from the upper left graph. Note also how the widths of the distributions are different in different parts of the image.

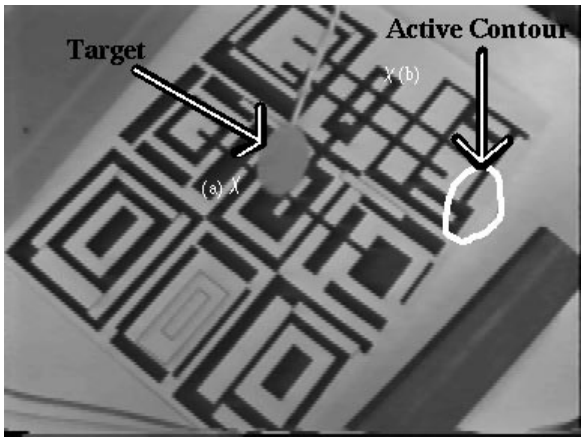


Figure 8: Key for figure 9. The gray oval is a target to be tracked, by an active contour (the white line) against a complex cluttered black and white background. The intensity profiles of points (a) and (b) are shown in figure 10

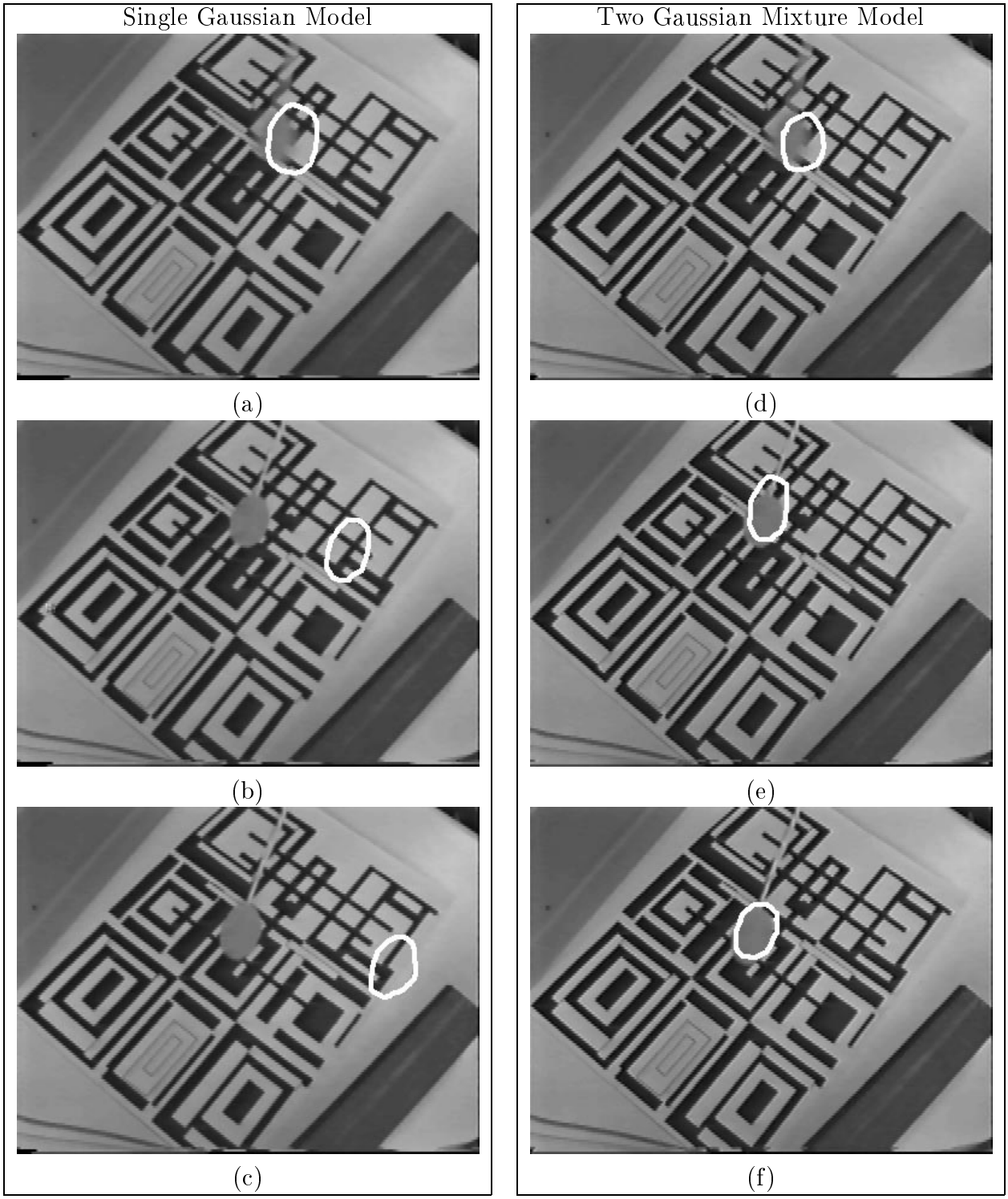
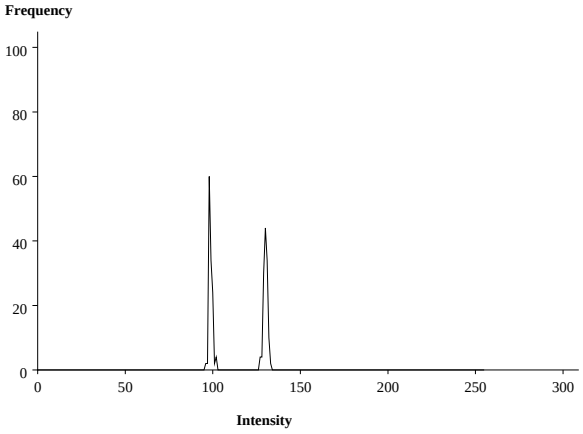
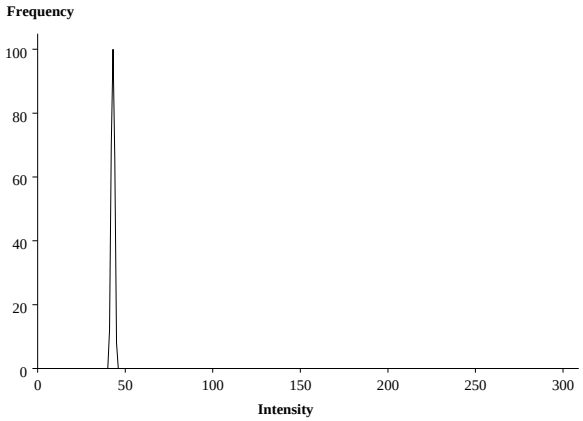


Figure 9: Tracking a fast moving target against clutter in a 4×4 sub-sampled image when the background is modelled by a single Gaussian (a)–(c), and a two Gaussian mixture (d)–(f). Note how the tracker using the single Gaussian model is more insensitive to the edge of the target, and looses track of it at the high acceleration between images (b) and (c). The improved discrimination of the two Gaussian model allows tracking to continue past this point (e)–(f).



Profile of point (a)



Profile of point (b)

Figure 10: Intensity profiles for points (a) and (b) in figure 8. Note that a two Gaussian mixture is necessary to model the profile of point (a), which lies on an edge in the background.

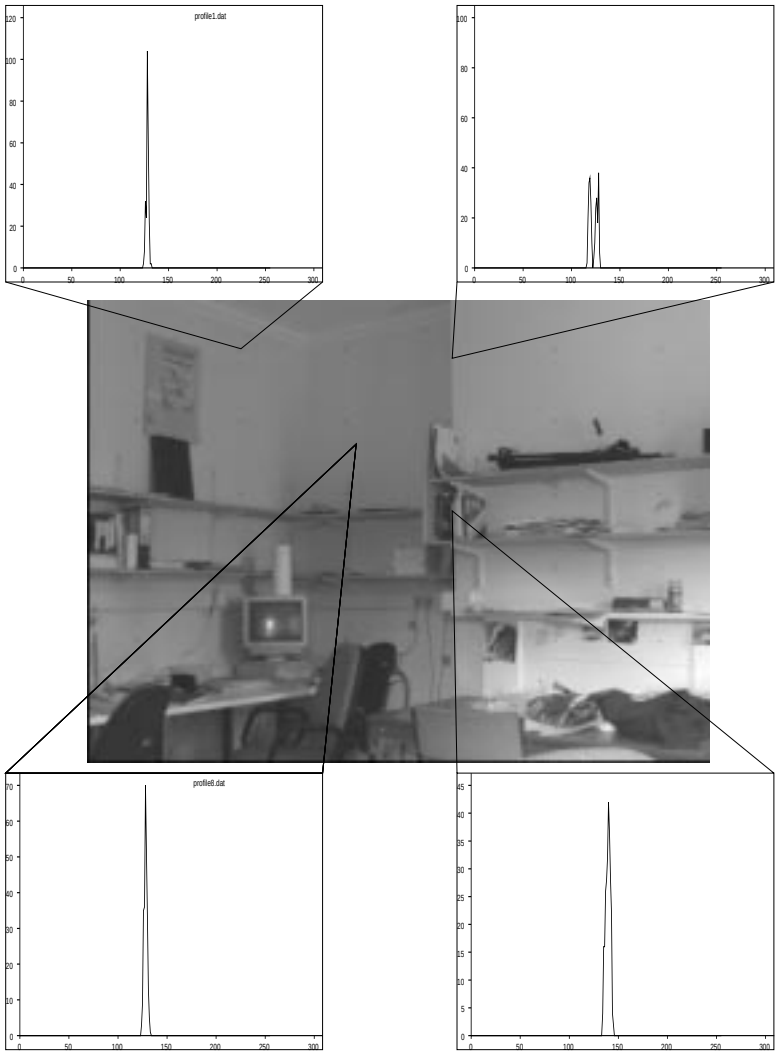


Figure 11: Background gradients for various parts of the room. Note that the widths of the gradients vary considerably, and that a two Gaussian model is necessary to fit one of the gradients.

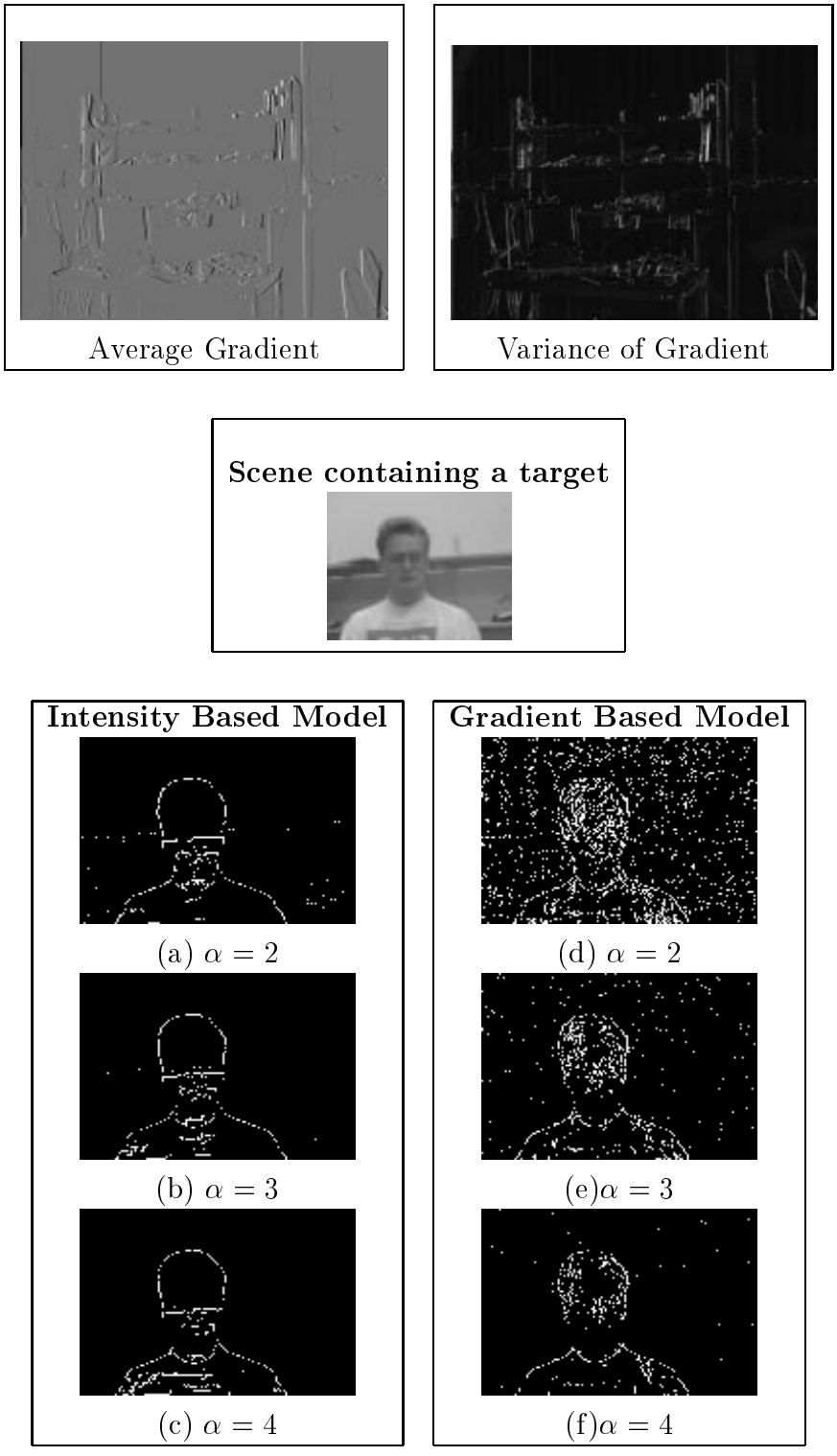


Figure 12: **An statistical intensity based background model provides a *cleaner* feature-map than a gradient based one.** The upper two figures show the average and variance of the gradient of intensity of a view of the room. Note how it is only edges in the background which are highlighted in either the average or variance maps. A target is then introduced into the scene. The lower figures ((a)–(f)) show a zoomed in view of the edges of the regions around the target which have failed the $\mu \pm \alpha\sigma$ test for inclusion into the background—the feature-map on which the tracker is searching. Note that the outlines of the target (the correct feature) given by the statistical intensity models are much cleaner and more complete than the corresponding ones produced from the statistical gradient based model. The tracker will track much better based on the intensity based model as it is operating in a much less cluttered space.

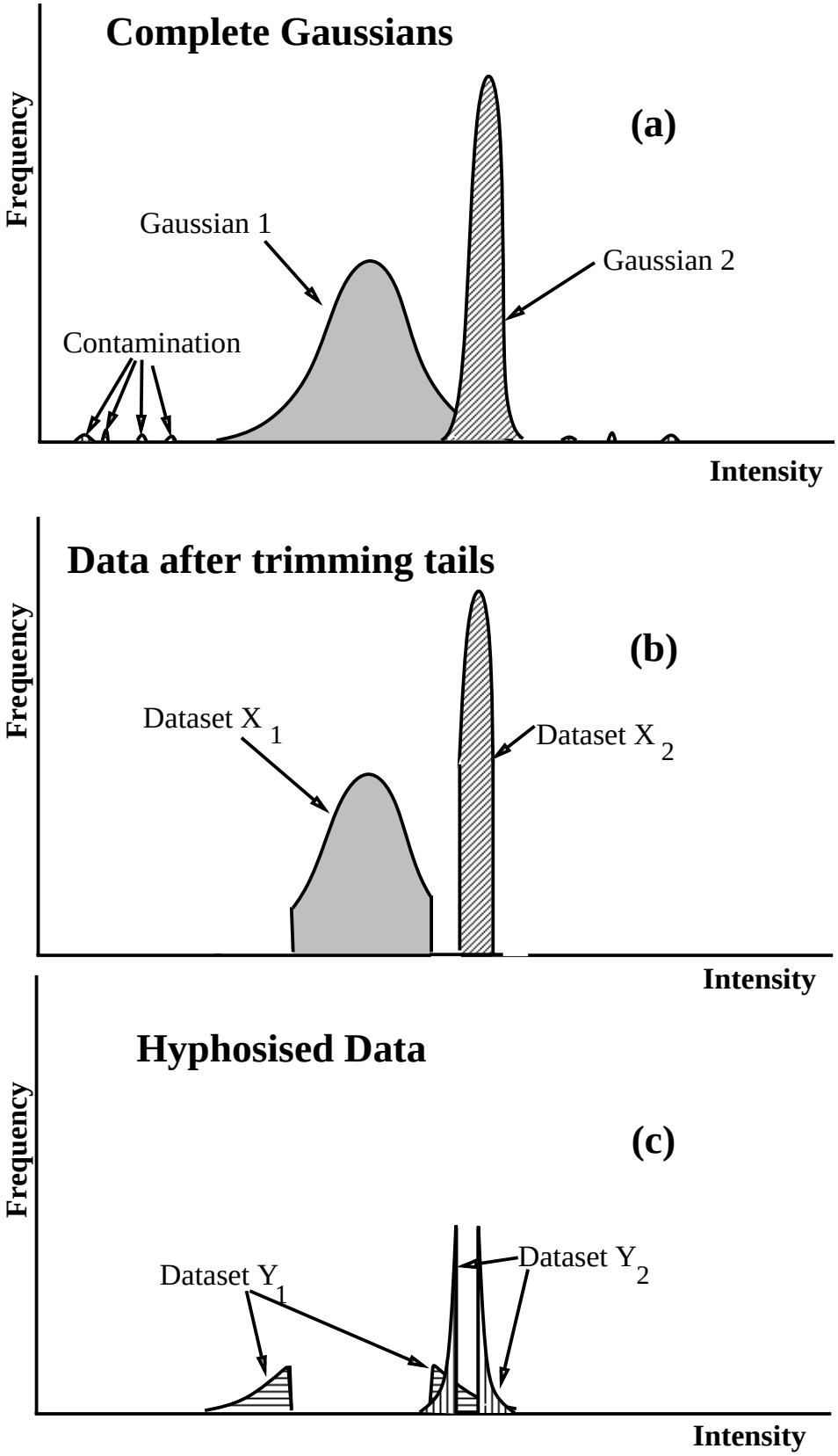


Figure 13: Datasets used in EM fitting of two Gaussian mixture

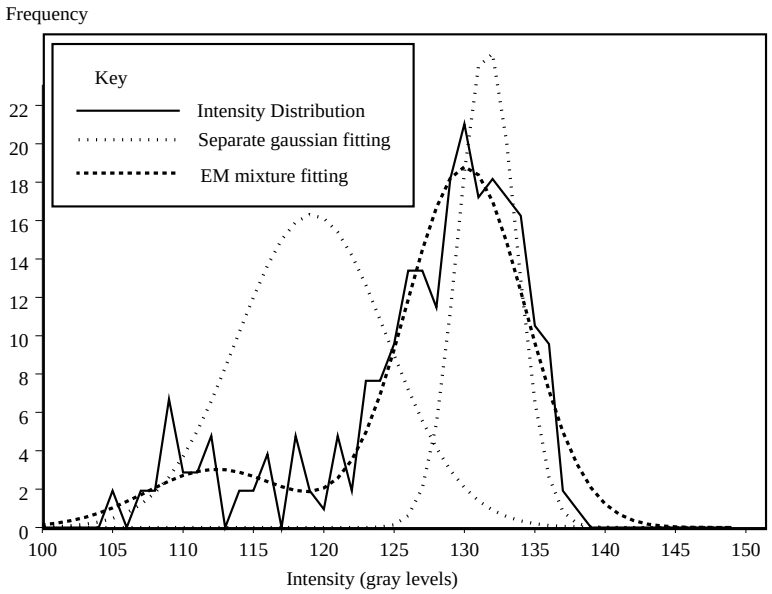


Figure 14: Use of EM for fitting a two Gaussian mixture. Note how the distribution fitted by the algorithm is much closer to the underlying distribution that that obtained by using the trim and fit algorithm (indeed the trim and fit algorithm was only able to find one correct Gaussian in this case, the other one displayed is the closest approximation to a Gaussian that it could find).