

Evaluation of a robust least squares motion detection algorithm for projective sensor motions parallel to a plane

Fabian Campbell-West and Paul Miller¹

Institute of Electronics, Communications and Information Technology (ECIT)
Queen's University Belfast
Belfast, BT3 9DT

ABSTRACT

A robust least squares motion detection algorithm was evaluated with respect to target size, contrast and sensor noise. In addition, the importance of robust motion estimation was also investigated. The test sequences used for the evaluation were generated synthetically to simulate a forward looking airborne sensor moving with translation parallel to a flat background scene with an inserted target moving orthogonal to the camera motion. For each evaluation parameter, test sequences were generated and from the processed imagery the algorithm performance measured by calculating a receiver-operating-characteristic curve. Analysis of the results revealed that the presence of small amounts of noise results in poor performance. Other conclusions are that the algorithm performs extremely well following noise reduction, and that target contrast has little effect on performance. The system was also tested on several real sequences for which excellent segmentation was obtained. Finally, it was found that for small targets and a downward looking sensor, the performance of the basic least squares was only slightly inferior to the robust version. For larger targets and a forward looking sensor the robust version performed significantly better.

Keywords: Motion estimation, least squares, small moving objects.

1. INTRODUCTION

At the heart of many surveillance and reconnaissance systems are the electro-optic sensor suite and the image analyst whose combined role is to determine if there are targets present and provide information on them. These tasks are difficult for the analyst due to the high degree of clutter in the imagery and the small target size. The analyst's efficiency will also decrease with time. Since one of the strongest cues for analysts is motion, their role can be made easier by automated target detection technology. However, the problem is made difficult by the small target size, motion of the platform sensor and also by parallax effects.

Relevant approaches to motion detection compensate for the sensor-induced motion by assuming that the velocity field can be modelled by a parametric transformation¹. The transformation parameters are determined by least squares regression that uses the brightness constancy constraint². There are several flaws with this single component model, which are exposed in many commonly occurring circumstances^{3,4}. To overcome these problems, various approaches have been developed. Burt *et al.*³ observed that the least squares technique has two modes of operation. In the first mode, the motion estimates are an average of motions in the image. In the second mode, the algorithm 'locks-on' to a single motion. Algorithms obtaining motion estimates over multiple frames⁴ and using a hierarchical image representation⁵ were designed to increase motion estimation accuracy. In addition, the basic least squares technique was

¹ {f.h.campbellwest, p.miller}@qub.ac.uk

made more robust by removing outliers in the scene. Following compensation, any residual motion can only be due to moving objects¹. Further attempts at achieving robustness generalized the least-squares approach to use M-Estimators⁶. An alternative version of least squares, designed to deal with outliers, has been formulated using robust statistics, allowing assumption violations to be detected⁷. Summaries of regression and robust regression are also available^{8,9}. Such approaches only work in certain conditions and fail when there is significant depth in the 2D images. One of the techniques¹ was developed to manage 3D scenes by decomposing the optical flow field into the plane and parallax components¹⁰.

More recent approaches to motion estimation^{11,12} solve the gradient based motion estimation problem using non-linear optimisation techniques. One system¹¹ first uses intensity segmentation to create patches. It then calculates a dense motion field by estimating a number of local parametric models, each of which describes the motion field of a local patch. Neighbouring patches with small intensity differences are merged - the aim is to group in the same patch pixels that by themselves yield unreliable motion constraints due to low-intensity variation around them. Occlusion problems are overcome by modelling the motion in both temporal directions, using a three frame approach.

Though arguably robust and more sophisticated than previous techniques, these recent algorithms preclude real-time implementation. Lim and Gamal¹³, as part of the ‘Programmable Digital Camera Project’ at Stanford, use a special sensor (CMOS) to capture video at very high frame rates – i.e. 1000s of frames per second. The optical flow is estimated, using the Lucas-Kanade method¹⁴, at the standard frame rate (30 fps) in real-time. In this paper we evaluate another technique¹ which could feasibly be implemented in real time using COTS hardware. We are interested in the effects of target contrast, target size and sensor noise on the performance of both the motion parameter estimation process and the segmentation of independently moving objects.

An overview of the robust least squares algorithm is given in section 2. The test image data set and the testing methodology are described in section 3. In section 4 all the test results are presented. Further analysis is provided in section 5, and section 6 concludes the paper.

2. BACKGROUND

2.1 Basic least squares

We model images as frames of a sequence, so that a frame is denoted as a function of space and time, $I(\mathbf{x}, t)$ where $\mathbf{x} = (x, y)$ is the vector of spatial position and t is the current time index in the sequence⁵. The images are captured from a sensor moving through a scene which may contain independently moving objects. The assumption of grey-level constancy gives the following relation between frames²:

$$I(\mathbf{x}, t) = I(\mathbf{x} + \mathbf{u}(\mathbf{x}, t), t + 1) \quad (1)$$

where $\mathbf{u}(\mathbf{x}, t) = (p(x, y, t), q(x, y, t))$ is the vector of instantaneous velocity for a specific pixel. For the majority of pixels in the image, it is assumed that the vector \mathbf{u} is the apparent motion of the background, caused by the sensor motion. The values of the parametric functions p and q are determined by the sensor motion model, the complexity of which reflects the sensor motion. Using a projective sensor model, \mathbf{u} is given by:

$$\begin{aligned} p(x, y, t) &= a + bx + cy + gx^2 + hxy \\ q(x, y, t) &= d + ex + fy + gxy + hy^2 \end{aligned} \quad (2)$$

where a and d are the horizontal and vertical pixel translation respectively, $a-f$ define an affine transform and g and h are related to the variation in depth facilitated by the full projective transform.

A Taylor series expansion of the right hand expression in (1), discarding non-linear terms, yields:

$$I(\mathbf{x}, t) = I(\mathbf{x}, t) + \mathbf{u} \nabla I + \Delta I(\mathbf{x}) \quad (3)$$

where $\nabla I = \frac{\partial I}{\partial \mathbf{x}}$ and $\Delta I(\mathbf{x}) = \frac{\partial I}{\partial t}$. Therefore

$$\mathbf{u}\nabla I + \Delta I(\mathbf{x}) = 0 \quad (4)$$

The structure of this solution to the motion estimation problem is generic in terms of the motion model, \mathbf{u} . It is through a particular definition of p and q that the sophistication of the motion estimation is determined. For example, if p and q are defined as scalar valued functions, then \mathbf{u} is simply a translation estimate between two frames. For a given motion model, the estimate can be obtained and refined using an iterative approach. The estimate after the i th iteration at time t is denoted by $\mathbf{u}_i(\mathbf{x}, t)$. The following recursive equations determine how the final estimate is achieved:

$$\mathbf{u}_0(\mathbf{x}, t) = 0 \quad (5)$$

$$\mathbf{u}_i(\mathbf{x}, t) = \mathbf{u}_{i-1}(\mathbf{x}, t) + \delta\mathbf{u}_i(\mathbf{x}, t) \quad \forall i > 0 \quad (6)$$

From (4), the least squares estimation of the motion parameters, \mathbf{u} , assuming that the sensor motion between consecutive values of t is small, is obtained from the minimisation of the error function, at time t within the analysis region R :

$$E(\delta\mathbf{u}_i, t) = \sum_{\mathbf{x} \in R} (\delta\mathbf{u}_i \nabla I(\mathbf{x}, t) + \Delta I(\mathbf{x} - \mathbf{u}_{i-1}, t))^2 \quad (7)$$

For a frame pair, $I(\mathbf{x}, t)$ and $I(\mathbf{x}, t+1)$ the motion \mathbf{u} between the frames can be estimated by differentiating the least-squares error function, Eqn. (7), with respect to the parameters of \mathbf{u} . For translational motions p and q have one parameter each and are scalar valued. Hence

$$\left[\sum \nabla I(\mathbf{x} - \mathbf{u}_{i-1}, t) \nabla I(\mathbf{x} - \mathbf{u}_{i-1}, t)^T \right] \delta\mathbf{u}_i = - \sum \nabla I(\mathbf{x} - \mathbf{u}_{i-1}, t) \Delta I(\mathbf{x} - \mathbf{u}_{i-1}, t) \quad (8)$$

Summations are performed over the region R , which can initially be the entire image or smaller regions.

The restraint that the motion \mathbf{u} must be small can be overcome by using a hierarchical multiresolution approach^{1,4,5} in which a n -tier Gaussian image pyramid is created for each of the images, so they have representations at different resolution levels denoted by a subscript, $I_k(\mathbf{x}, t)$. The original, high-resolution, image is denoted by $k=0$ and the lowest resolution image, $1/2^{n-1}$ the size of $I(\mathbf{x}, t)$, is denoted by $k=n$. The maximum value of n is bounded only by the size of the input images. For our application, a value of $n=2$ is sufficient. Hence, a motion of 4 pixels at $k=0$ becomes a motion of 1 pixel at $k=2$ satisfying the small motion constraint. The motion estimation process uses the different resolutions as follows. Starting at $k=2$ the least squares calculation is applied iteratively to obtain an estimate of \mathbf{u} after i iterations. The process repeats for $k=1$ except that Eqn. (5) becomes

$$\mathbf{u}_0^k(\mathbf{x}, t) = 2\mathbf{u}_i^{k+1}(\mathbf{x}, t) \quad \forall k < n \quad (9)$$

Following measurement of the motion parameters, a relatively straight-forward approach to detecting independent motion is simply to threshold the difference image D defined by

$$D(\mathbf{x}, t) = |I(\mathbf{x}, t+1) - I(\mathbf{x} - \mathbf{u}, t)| \quad (10)$$

A potential problem with using the difference image is that if there are small errors in the measurement of the motion parameters, the registered images will be slightly misaligned. Consequently, the two will not cancel each other out when subtracted. This is not a problem where the spatial gradient is small, however, it does become a problem when it is large as the small misalignment could lead to a large magnitude in the difference image, causing misclassification. This

problem can be overcome¹ by using a more sophisticated approach in which the difference image is normalized by taking its inner product with the spatial gradient over a small local neighbourhood. A measure of the motion of each pixel is calculated as follows:

$$M(\mathbf{x}, t) \equiv \frac{\sum_{(\mathbf{x}_i) \in N(\mathbf{x})} |D(\mathbf{x}_i, t)| |\nabla I(\mathbf{x}_i, t)|}{\sum_{(\mathbf{x}_i) \in N(\mathbf{x})} |\nabla I(\mathbf{x}_i, t)|^2 + C} \quad (11)$$

where $N(\mathbf{x})$ is a 3×3 neighbourhood of \mathbf{x} and the constant C prevents numerical errors. The distribution of values M over the image indicate the magnitude of motion of a given pixel. Hence, M is known as the motion-measure map¹ and when combined with a threshold, T_M , can be used to classify regions of $I(\mathbf{x}, t)$ as moving or stationary.

2.2 Robust least squares

The ‘single pattern component’ motion model described in section 2.1 is inappropriate for many realistic scenes because of the assumptions involved. For our application the scenes produced consist of a moving dominant pattern, the background, and a much smaller pattern, the target, moving independently. A problem with this is that pixels of independent objects do not satisfy the constraint equation, Eqn. (1). Consequently, the target pixels will introduce errors in the motion parameter estimates. For example, in some circumstances, the estimates produced are a weighted average of the background and target motions. The errors may result in misregistration causing background pixels to be classified as moving independently. Ideally, to overcome this problem, we would like to remove the target pixels from the region of analysis R . There are many ways of achieving a more robust estimate. The most straight forward approach is simply to use the classification of moving/stationary regions obtained from the first frame pair to define a region of support (ROS) for the next frame pair. However, Irani *et al.*¹ developed a more sophisticated approach that employed a second classification scheme to determine background pixels. For this scheme they introduced the concept of reliability, Z , of the motion measure map value. This can be associated with a threshold, T_Z . The reliability is defined as the numerical stability of the optical flow equations. Due to the aperture problem, a neighbourhood may have a small motion measure map value, when in fact it may be moving independently. Therefore, Irani classified only those pixels as background that had a small motion measure map value and a high reliability. As a consequence some background may be removed from the ROS, however, this is not a problem provided the majority of background pixels are included. The important thing is to remove independently moving pixels from the ROS as it is these that cause errors. Furthermore, because the aperture problem is not as severe at lower resolutions, if a pixel has a low motion measure map value and a low reliability at the highest resolution, the motion measure map and the reliability at the lower resolution is used to classify it as belonging to the background or not. This further reduces the number of background pixels not included in the ROS. The ROS is therefore defined by

$$ROS_k(\mathbf{x}, t) = \begin{cases} 0 & \text{if } M(\mathbf{x}, t) > T_M \text{ and } Z(\mathbf{x}, t) > T_Z \\ ROS_{k+1}(\mathbf{x}/2, t) & \text{otherwise} \end{cases} \quad (12)$$

where for $k = n$ the ROS is given the value 1 if the thresholds are not exceeded.

To estimate the projective parameters, the hierarchical motion analysis is firstly performed assuming a translational model with $n = 2$. Following processing over the three resolution levels, a ROS is obtained. The affine analysis is then performed over this ROS but with $n = 1$. When the affine motion model has been used and the ROS updated, the projective motion model is applied at the highest resolution level ($n = 0$). Finally, rather than use $I(\mathbf{x}, t)$ and $I(\mathbf{x}, t + 1)$ to perform the motion parameter estimation, $I(\mathbf{x}, t)$ is replaced by a weighted average of registered frames as follows:

$$\begin{aligned} Av(\mathbf{x}, 1) &= I(\mathbf{x}, 1) \\ Av(\mathbf{x}, t + 1) &= (1 - w)I(\mathbf{x}, t + 1) + w.Av(\mathbf{x} - \mathbf{u}, t) \end{aligned} \quad (13)$$

where $Av(\mathbf{x},t)$ is the weighted average at time t and w is a constant such that $w \in (0,1)$. We set $w = 0.8$. The temporal integration reduces the effect of noise on the motion measure estimates. Also, because it is the dominant background component that is being registered, the background remains sharp. However, independently moving targets will blur, reducing the errors in the motion estimates.

3. METHODOLOGY

A series of high fidelity image sequences were artificially generated using a synthetic background image. The statistics used to generate the synthetic image were measured from a set of real background images obtained from an airborne sensor. An infra red model of a land rover was inserted into the sequences to simulate an independently moving object in the sensor field of view, Fig. 1. The target motion is horizontal across the image for all the tests.

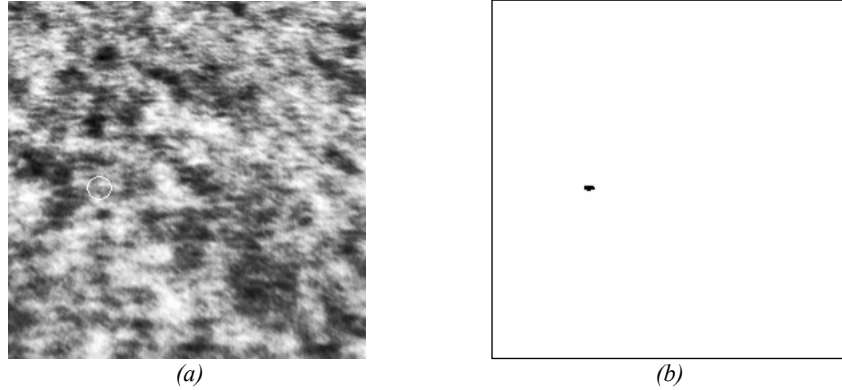


Figure 1: An example frame from a sequence simulating a forward looking sensor, (a), and an example segmentation (inverted), (b).

Generating sequences in this manner gives complete ground-truth and the ability to vary target size, contrast and add sensor noise. Experiments were conducted to measure the variation in system performance with respect to these variables. Contrast is a measure of the relative difference between an object and the background. It can be approximated by the ratio¹⁵

$$C = \frac{\bar{r} - \bar{b}}{\bar{r} + \bar{b}} \tag{14}$$

where \bar{r} and \bar{b} are the average object and background intensity respectively. The sensor noise, $n(\mathbf{x},t)$, is modelled as additive ‘white noise’, which has a Gaussian distribution with zero mean. The level of noise present is measured by the signal-to-noise ratio (SNR) which is based on the relative variance of the image and the noise and is defined by

$$SNR = 10 \log_{10} \frac{\sigma_{b+r}^2}{\sigma_n^2} \tag{15}$$

where σ_{b+r} and σ_n are the variances of the scene and noise respectively. Interpreting this value, if the variance of the noise is equal to the variance of the scene then the SNR is zero. A less corrupted image will have a high SNR, while a heavily corrupted image will have a low or negative SNR. From (15) the appropriate variance of the noise distribution can be determined for a desired SNR.

The test results are presented in terms of ROC curves, which show the probability of detection, p_d , against the probability of false alarm, p_{fa} . The definition of these probabilities is given by

$$p_d = \frac{\bar{n}_r}{N_r}, p_{fa} = \frac{\bar{n}_b}{N - N_r} \quad (16)$$

where \bar{n}_r is the average number of segmented pixels in the target region, \bar{n}_b is the average number of segmented pixels in the background region, N_r is the number of known target pixels and N is the total number of pixels in the image. The averages are calculated over the frames from different sequences for each test variable. To generate several points for each curve, the algorithm is applied using different thresholds on the motion measure map. Finally, the tests were performed with both the basic version of the least squares, described in section 2.1, and the robust version, section 2.2. All results presented are for the robust version.

4. RESULTS

4.1 Target contrast

In these tests, the contrast between the target and the background is varied by adding a constant to the target intensity. Analysis of the effect of contrast is presented in section 5.1. The error in the estimation of p and q is independent of target contrast relative to the background. Further to this, errors in the parameter estimation for the robust system were only marginally smaller than for the basic system. Figure 2 shows the ROC curves for the former for each of the different contrast values where it is clear that variation in target contrast has a negligible effect on performance. Visual inspection of an example segmented image, Fig. 1(b), confirms that most of the target area has been detected, whilst there are very few false alarms. Lastly, the performances, in terms of ROC curves, for the robust and basic least square versions were the same.

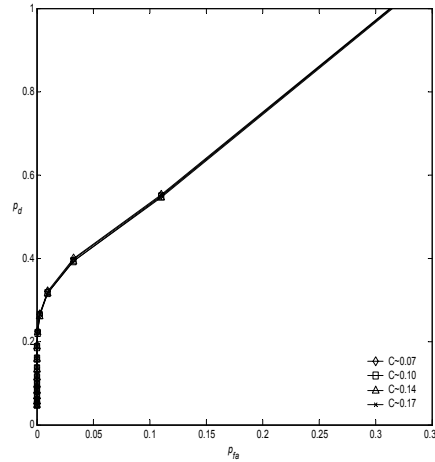


Figure 2: Variation in ROC curve with target contrast.

4.2 Target size

In this test the effect of target size on the algorithm performance is determined. Target sizes of 8×6 , 16×10 , 32×18 and 64×32 are used. As the target size increases it affects the accuracy of the least squares estimation, since the target moves independently of the background. The robust system very quickly removes the target region from the ROS of the background, and so its motion does not adversely affect the estimation of the motion parameters. However, when using the basic system, the target motion influences the measured background motion. Therefore, the use of the robust version is important as the target size increases. Figure 3(a) shows the ROC curves for the different target sizes. The reason for the improvement in ROC curve for smaller target sizes is due to the definition of p_d and p_{fa} in Eqn. (16). Since the target motion is constant, the ratio of the area of occlusion/disocclusion to that of target size, N_r , gets smaller for larger targets. Hence, the p_d drops with an increase in target size. However, this does not actually constitute a drop in

performance with size. Furthermore, there is no significant difference between the robust and basic versions for the sizes shown.

The influence of the target in the motion parameter estimates for larger targets can be seen in Fig. 3(b). A very large target, approx. 192×76 , was inserted into a synthetic sequence of images sized 256×256 with a vertical camera motion of 2.11 pixels. The true target motion was purely horizontal. This sequence was processed using both the basic and robust systems. The robust system quickly removes the target region from the least squares analysis and so the error in the horizontal parameter reduces to zero. For the basic system, the error is consistent over the entire sequence. The target also introduces a small error in the vertical motion parameter, which again is less for the robust system for the same reason.

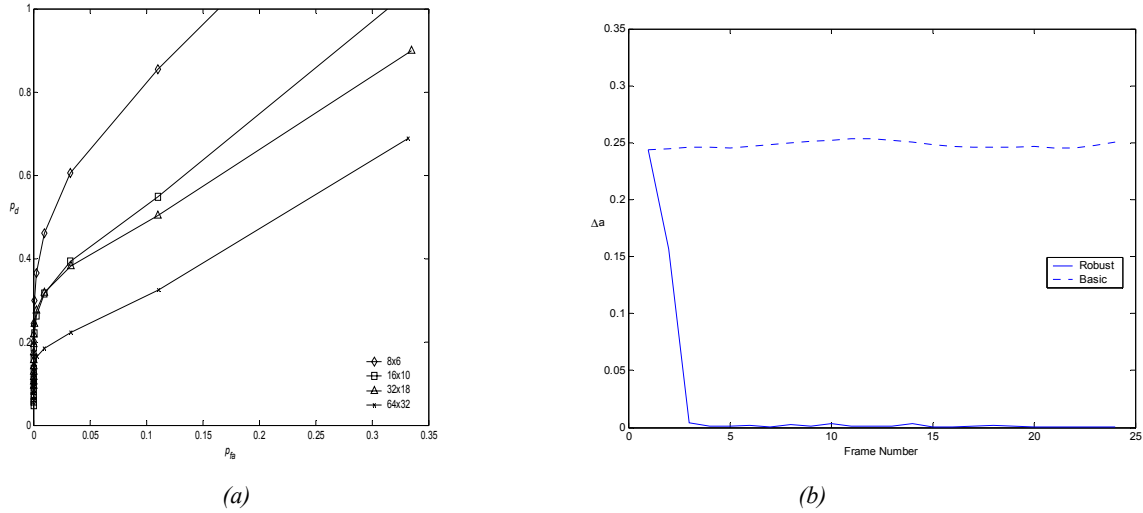


Figure 3: Variation in ROC curve with target size, (a). Horizontal motion parameter estimates, (b), in the presence of a large target for the basic and robust systems.

4.3 Sensor noise

To investigate the effect of noise on algorithm performance, the test image sequences were corrupted with white Gaussian noise. For each of the five tests, the variance was changed to give SNRs of 4, 8, 12, 24, and 48. With the exception of the two heaviest levels of noise, the motion parameter estimates obtained with the robust version were all accurate to two decimal places. In contrast, the errors for the basic system were much larger and increased with increasing levels of noise. To explain these results, consider that for the robust version the motion parameter estimation is helped by temporal integration. This uses the temporally averaged image A_V defined in (13). When high levels of noise are present in the image, large regions can be incorrectly classified as moving in the motion measure map. This in turn means that large regions are removed from the analysis region, R . After several frames, this region decreases in size until eventually there are insufficient points remaining to solve the system of equations, (8). Therefore the robust system needs to be modified to prevent such an event occurring.

Despite the reasonable parameter estimates for the robust version, the ROC curves in Fig. 4(a) show that noise strongly affects the segmentation. This is because the final motion measure maps are constructed using the original imagery in which the full noise is still present. Whilst the p_d values are high, attention should be paid to the scale of the p_{fa} axis which indicates nearly the entire image is being segmented for low values of T_M . If some pre-processing is performed, then the results can be improved. Figure 4(b) shows the ROC curves obtained for the same test, except here the images are pre-processed with a 5×5 Weiner filter to remove noise. At low thresholds, the segmentation is still poor. However at higher thresholds with a reasonable amount of noise the detection rate increases for a constant and low false alarm rate. Further analysis of the effects of noise is presented in section 5.2.

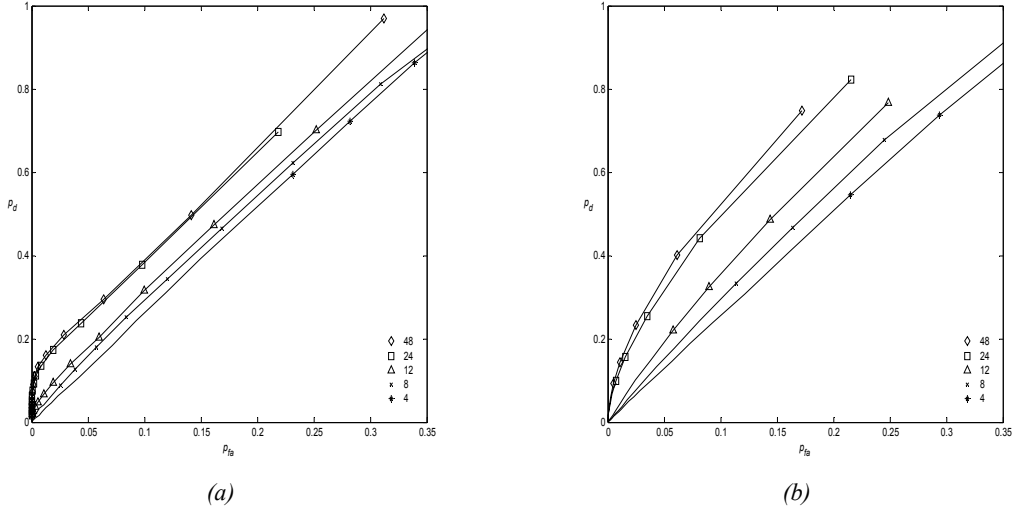


Figure 4: Variation in ROC curve with sensor noise without pre-processing, (a), and with pre-processing, (b).

4.4 Comparison of affine and projective motion models

In this section we investigate the effect of changing the system sophistication and computational complexity. On one hand, we can use the full projective motion model or, alternatively, we can limit the system to use at most the affine motion model. There are two aspects to the results here. The first is a comparison between the motion estimation accuracy and the degree of tilt in the forward looking sensor. A tilt of 0° is equivalent to a downward looking sensor, and a tilt of 90° is equivalent to a sensor looking directly ahead. The second aspect of the results is a direct analysis of the benefits of using the full projective motion model. A series of image pairs were created which simulate a moving sensor at varying angles as described above. The robust motion estimation algorithm was then applied to the image pairs in turn and the mean value of $D(\mathbf{x}, t)^2$ calculated. Figure 5 shows the results of the mean squared difference plotted against sensor tilt for both affine and projective motion models with a vertical zoom and parallel translation.

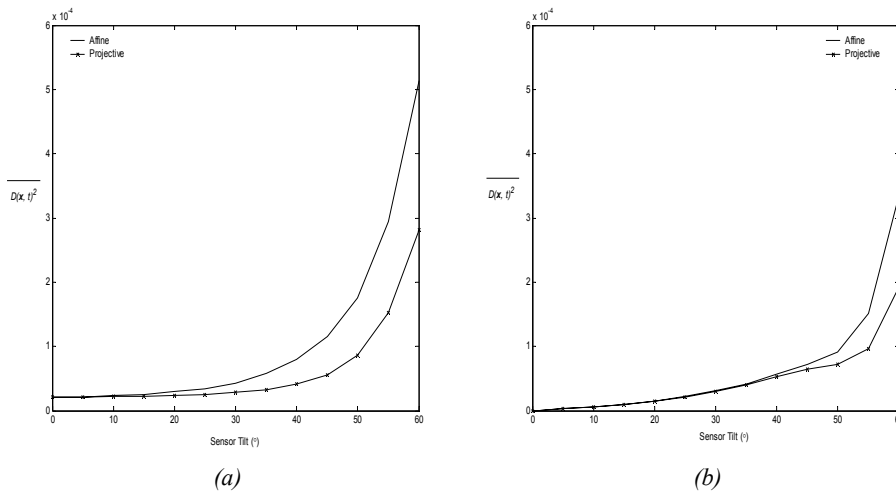


Figure 5: Variation with mean squared difference between registered frames for different sensor tilts using affine and projective motion estimates. The true motion between images is a zoom, (a), and a parallel translation, (b).

These results show that when the camera is translating, the affine and projective models perform equivalently up to a tilt of 45° . As the tilt increases further, the error increases for both motion models but more so for the affine model. This is expected, because the affine model does not incorporate depth differences in the 2D image.

4.5 Real imagery results

This section contains results from tests carried out with various real sequences, all obtained from a sensor on a moving platform, chosen to contain affine sensor motions with varying tilts. Since there is no ground-truth information available, success of the algorithm can only be determined by the segmentations obtained. Figure 6(a) shows a frame from a sequence where the sensor is panning to follow a moving vehicle and zooms in as it translates. The segmentation of the vehicle is shown in Fig. 6(d). The vehicle is partially occluded by vegetation, hence only the front and rear of the vehicle are segmented. Figure 6(b) shows another example, from a sequence where the sensor rotates as it translates parallel to the ground. The resulting segmentation, Fig. 6(e), show the vehicle clearly. Figure 6(c) shows a final example, this time from a sequence involving a variety of camera motions. In this scene, there are two vehicles, moving in the same direction, and both are successfully segmented, Fig. 6(f).

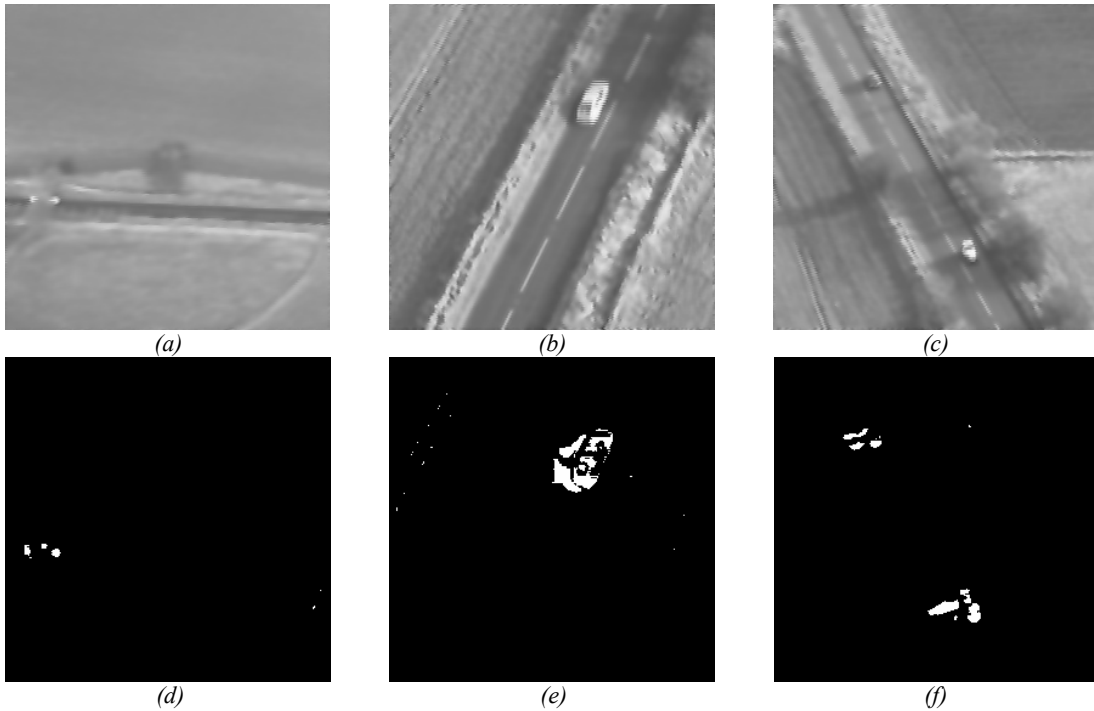


Figure 6: Frames from sequences involving a sensor translation and zoom, (a), a rotation and translation, (b) and a medley of motions, (c). The corresponding segmentations, (d-f) are also shown.

5. DISCUSSION

In this section we expand on the result analysis, in particular we show how target contrast variation does not affect performance and the effect of noise on the image.

5.1 Analysis of target contrast

As mentioned in section 2.2, the single motion component model is unrealistic. For a flat image obtained from a translating sensor with one target present in the FOV, the image composition, ignoring noise for the moment, can be explained by

$$I(\mathbf{x}, t) = b(\mathbf{x} - \mathbf{u}t) [1 - w_r(\mathbf{x} - \mathbf{v}t)] + w_r(\mathbf{x} - \mathbf{v}t) r(\mathbf{x} - \mathbf{v}t) \quad (17)$$

where $b(\mathbf{x})$ denotes the background clutter, $r(\mathbf{x})$ denotes the target, $w_r(\mathbf{x})$ denotes the target window and $\mathbf{v} = (p(\mathbf{x}, t) + m, q(\mathbf{x}, t) + n)^T$ is the motion of the target. The target window is defined as

$$w_r(\mathbf{x}) = \begin{cases} 1 & \text{when } r(\mathbf{x}) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

The equation to be minimised, from (7), is

$$\delta \mathbf{u}_i \frac{\partial I(\mathbf{x}, t)}{\partial \mathbf{x}} + \frac{\partial I(\mathbf{x}, t)}{\partial t} = 0 \quad (19)$$

and thus equation (8) can be written as

$$\left(\begin{array}{cc} \sum \left(\frac{\partial I(\mathbf{x} - \mathbf{u}_{i-1}, t)}{\partial x} \right)^2 & \sum \frac{\partial I(\mathbf{x} - \mathbf{u}_{i-1}, t)}{\partial x} \frac{\partial I(\mathbf{x} - \mathbf{u}_{i-1}, t)}{\partial y} \\ \sum \frac{\partial I(\mathbf{x} - \mathbf{u}_{i-1}, t)}{\partial x} \frac{\partial I(\mathbf{x} - \mathbf{u}_{i-1}, t)}{\partial y} & \sum \left(\frac{\partial I(\mathbf{x} - \mathbf{u}_{i-1}, t)}{\partial y} \right)^2 \end{array} \right) \delta \mathbf{u}_i = - \left(\begin{array}{c} \sum \frac{\partial I(\mathbf{x} - \mathbf{u}_{i-1}, t)}{\partial x} \frac{\partial I(\mathbf{x} - \mathbf{u}_{i-1}, t)}{\partial t} \\ \sum \frac{\partial I(\mathbf{x} - \mathbf{u}_{i-1}, t)}{\partial y} \frac{\partial I(\mathbf{x} - \mathbf{u}_{i-1}, t)}{\partial t} \end{array} \right) \quad (20)$$

If the motion parameter estimates are, indeed, independent of target contrast, it remains to demonstrate that, with $I(\mathbf{x}, t)$ as defined in (17), the partial derivatives in (20) are independent of target contrast. Dealing with the spatial gradients first, we obtain from (17)

$$\begin{aligned} \frac{\partial I(\mathbf{x}, t)}{\partial x} &= b_x - w_x b - w_r b_x + w_r r_x + w_{r_x} r \\ &= b_x - w_r b_x + w_r r_x + w_{r_x} (r - b) \end{aligned} \quad (21)$$

The presence of any terms involving the target can be seen as error terms in the estimate of the true motion of the background. The terms containing b_x and r_x are independent of target contrast. In fact, target contrast only affects the spatial gradients at positions where the background and target are in the local neighbourhood, i.e. pixels where $w_{r_x} > 0$.

The size of this neighbourhood is determined by the method used to calculate the spatial derivatives, but in essence the total number of pixels in this category is equivalent to the target perimeter. Since the target used for the contrast experiments is small, the perimeter is very small relative to the total area which the estimate is calculated over. Hence even large contrast variations do not affect the motion parameter estimates. The same argument can be used for the vertical spatial gradient and temporal gradient, since the equations involve similar terms. In the latter case, however, the pixels where $w_{r_x} > 0$ will be more numerous since the window w_r is moving.

It remains to explain why the ROC curve performance is constant with varying target contrast. When two images involving a moving object are registered there are four distinct regions of interest:

1. Regions that belong to the background in both images.
2. Regions that belong to the object in both images.
3. Regions that belong to the background in image one, but the object in image two.
4. Regions that belong to the object in image one, but the background in image two.

Region three is known as an occlusion region and region four a disocclusion region. In terms of the difference of the registered images, region one will be zero, or close to it, if the motion parameter estimates are accurate. The distribution of region two depends on the internal structure of the object. If it is of relatively uniform intensity, then region two will be close to zero. If there are large intensity variations, the region will have higher values. Regions one and two are independent of target contrast. Regions three and four will vary proportionally with contrast. If the threshold of the

difference image is between region one and the other regions, there will be no difference in performance for different contrast values. This is what has occurred in the tests presented in section 4.1.

5.2 Analysis of effects of noise

If we ignore the presence of a target, for simplicity, then (17) becomes

$$I(\mathbf{x}, t) = b(\mathbf{x} - \mathbf{u}t) + n(\mathbf{x}, t) \quad (22)$$

In a similar approach to that in section 5.1, we arrive at (20). It remains to demonstrate that the spatial and temporal partial derivatives are affected by the noise function, n . The spatial and temporal derivatives are given by

$$\frac{\partial I}{\partial x} = b_x + n_x, \frac{\partial I}{\partial y} = b_y + n_y, \frac{\partial I}{\partial t} = b_t + n_t \quad (23)$$

The terms involving n can be seen as error terms in the estimate of the true motion parameters. Unlike the presence of an object, which is localised, noise affects the entire image region. The magnitude of n is determined by the SNR. As the SNR decreases, the magnitude of n increases which increases the values of the derivatives of n . As these values increase, so their contribution to the image derivatives, and hence error in the sum, increases.

Since the images are corrupted, the grey-level constancy assumption no longer holds. Even with perfect alignment of images, the region of the difference image corresponding to background will be non-zero. The magnitude of the difference increases with decreasing SNR. With regard to the four regions defined in the previous section, noise affects all four but most noticeably the first. As the noise increases, so the difference between the first region and the other three decreases. With high levels of noise, region one can swamp the other regions. This makes it increasingly difficult to position a threshold to distinguish background pixels from object and occluded pixels. A result of this is that large numbers of background pixels are misclassified as independently moving objects which increases the p_{fa} rate. The p_d rate also increases in the ROC curves because some pixels that would normally be below the threshold in the region of the target are also affected by the noise.

6. CONCLUSION

A parametric model for estimating sensor motion between frames of image sequences was presented and developed to cater for an independently moving object in the scene. This model was incorporated into a system for estimating the motion in image sequences using an iterative multiresolution approach. The developments to the system, motion classification at different resolution levels and temporal integration, were introduced to increase the accuracy of the estimated motion parameters. A set of experiments were then carried out to assess the system performance with respect to the variables of target contrast, target size and sensor noise. The main conclusions to be drawn from this work are:

- Excellent-to-good segmentation was obtained for all tested real image sequences acquired by a sensor undergoing affine motions with varying angles between the ground plane and the sensor.
- For small targets considered it was found that the contrast did not affect performance.
- For a range of target sizes, corresponding to those acquired by a surveillance sensor in medium FOV, there is no significant affect on performance. For larger targets, the robust algorithm performs significantly better than the basic least squares.
- Small amounts of sensor noise will degrade performance significantly. Therefore, some form of noise reduction pre-processing is required for real image sequences. However such pre-processing impairs the ability to detect small objects.
- The robust algorithm gives slightly better performance with real image sequences than the basic system but not for the synthetic sequences.
- As the target size increases, or the sensor moves towards looking forwards, the robust least squares performs better than the basic version.

In summary, it would appear that for projective sensor motions with a large sensor tilt, i.e. forward looking, a robust least squares algorithm is capable of detecting small low contrast targets embedded in high degrees of clutter. However, a pre-processing step is required to reduce any sensor noise.

ACKNOWLEDGMENTS

Fabian Campbell-West gratefully acknowledges funding from EPSRC (Grant no. GR/R52138) and Octec Ltd.

REFERENCES

- 1 Irani, M., Rousso, B., and Peleg, S., "Computing Occluding and Transparent Motions", *Int. J. Comput. Vision*, **12**, pp. 5-16, 1994.
- 2 Horn, B.K.P., Schunck, B.G., "Determining Optical Flow", *Artificial Intelligence*, **17**, pp. 185-203, 1981
- 3 Burt, P.J., Hingorani, R., Kolczynski, R.J., "Mechanisms for Isolating Component Patterns in the Sequential Analysis of Multiple Motion", *IEEE Workshop on Visual Motion*, Princeton, NJ, pp. 187-193, 1991
- 4 Bergen, J.R., Burt, P.J., Hingorani, R., Peleg, S., "A Three-Frame Algorithm for Two-Component Motion", *IEEE Trans. PAMI*, **14**, pp. 886-896, 1992
- 5 Bergen, J.R., Anandan, P., Hanna, K.J., Hingorani, R., "Hierarchical Model-Based Motion Estimation", *Computer Vision ECCV*, pp. 237-252, 1992
- 6 Stewart, C.V., "Robust Parameter Estimation in Computer Vision", *SIAM Review*, **41**, pp. 513-537, 1999
- 7 Black, M. J., and Anadan, P., "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields", *Computer Vision and Understanding*, **63**, pp. 75-104, 1997.
- 8 Weisstein, E.W., "Least Squares Fitting", *Mathworld-A Wolfram Web Resource*, <http://mathworld.wolfram.com/LeastSquaresFitting.html>
- 9 Meer, P., Mintz, D., Rosenfeld, A., "Robust Regression methods for computer vision: a review", *Int. J. Comp. Vis.*, **6**, pp. 59-70, 1991
- 10 Irani, M., and Anandan, P., "A Unified Approach to Moving Object Detection in 2D and 3D Scenes", *IEEE Trans. PAMI*, **20**, pp. 577-589, 1998.
- 11 Patras, I., Worring, M., Boomgaard, R., "Dense Motion Estimation Using Regularization Constraints on Local Parametric Models", *IEEE Trans. Image Processing*, **13**, pp.1432-1443, 2004
- 12 Keller, Y., Averbuch, A., "Fast Motion Estimation Using Bidirectional Gradient Methods", *IEEE Trans. Image Proc.*, **13**, pp. 1042-1054, 2004
- 13 Lim, S., Gamal, A.E., "Optical Flow Estimation using High Frame Rate Sequences", *Proc. Int. Conf. Image Processing*, **2**, pp. 925-928, 2001
- 14 B. D. Lucas, T. Kanade, "An iterative image registration technique with an application to stereo vision", *Proceedings of DARPA Image Understanding*, pp. 121-130, 1981
- 15 Gordon, R., Rangayyan, R.M., "Feature enhancement of film mammograms using fixed and adaptive neighbourhoods", *Applied Optics*, **23**, pp.560-564, 1984