Using Mutual Information to identify coupled motion

Neil M. Robertson

September 10, 2002

1 Introduction

We are interested in using mutual information to find out to what degree the motion of objects is connected. There are other metrics we have experimented with such as minimum description length and correlation however there are limitations with correlation such as the assumption that the data is linear, Gaussian and stationary. It is known that mutual information works well with non-linear and non-Gaussian data [4]. The work of Viola [1] and a subsequent implementation of this work by Gilles [2] provides a basis for the calculation of mutual information from samples of random variables, which is defined as $I(X,Y) = H(X) + H(Y) - H(X,Y)$[1] , where $H(n)$ is the entropy of variable n. An important aspect of [1] and [2] is the use of the derivative if information, $\frac{dI}{dT}$. Where they were concerned T is the transformation matrix for fitting models to images or registering images. In our case, T is the *translation in time* applied to one trajectory in order to maximise mutual information. Once the derivative is calculated a basic gradient descent algorithm can be formulated as follows.

$$T \leftarrow T + \lambda \frac{\overline{dI}}{dT} \tag{1}$$

λ is known as the *learning rate* and is a scalar quantity which determines how far along the curve of information vs. translation we move for a given iteration of the algorithm. Intuitively if the gradient of information for a certain translation is negative we make the time shift such that we move closer to a zero crossing and hence, ultimately find the maximum of mutual information. The overall goal is to ensure we are not ignoring situations where the motion may be out of phase and where merely calculating mutual information for the $T = 0$ situation will be misleading.

[1] For a derivation of this equation see appendix A.

2 Derived equations for trajectories

Given that mutual information is closely linked to entropy, we must find a way to calculate the entropy of the vector samples of the random variables. In our case the assumption is that, given we can't say with any high degree of certainty where the object will move by the time we next sample it, we can take it as a random variable which we then sample to find a *probability density*, allowing us to calculate entropy. A method for doing this uses the *Parzen Window* to model the probability density [3]. We approximate the underlying distribution as a superposition of Gaussian densities centred on the elements of the sample A taken from the elements of random variable z i.e.

$$p(z) \approx \frac{1}{N_A} \sum_{z_j} G_\psi(z - z_j) \tag{2}$$

Where,

$$G_\psi \equiv (2\pi)^{\frac{-n}{2}} |\psi|^{\frac{-1}{2}} exp(\frac{-1}{2} z^T \psi^{-1} z) \tag{3}$$

In the vector case, where the samples are $(x, y)^T$ from a trajectory, this equation still holds.

The derivative of information is given by

$$\frac{d}{dT} h(z(T)) \approx \frac{1}{N_B} \sum_{x_i \in A} \sum_{x_j \in B} (v_i - v_j)^T [W_v(v_i, v_j) \psi_v^{-1} - W_{uv}(w_i, w_j) \psi_{vv}^{-1}] \frac{d}{dT} (v_i - v_j) \tag{4}$$

With

$$W_v(v_i, v_j) = \frac{G_{\psi_v}(v_i - v_j)}{\sum_{x_k} G_{\psi_v}(v_i - v_k)} \tag{5}$$

and

$$W_{uv}(w_i, w_j) = \frac{G_{\psi_{uv}}(w_i - w_j)}{\sum_{x_k} G_{\psi_{uv}}(w_i - w_k)} \tag{6}$$

Where the following definitions hold: $u_i = u(x_i)$, which is the sample of the untranslated, or model, trajectory; $v_i = v(T(x_i))$, which is the trajectory translated in time by T; and $w_i = [u_i, v_i]^T$. x_i is sampled from one trajectory, x_j from the other. A and B are two random sample sets drawn from the set of data taken from the tracked trajectories.

In (4) the term $\frac{d}{dT}(v_i - v_j)$ is calculated by taking a simple gradient measure of the two terms in the simplification $\frac{d}{dT}(v_i) - \frac{d}{dT}(v_j)$ that is, measuring how the point v_i or v_j changes with translation in time of one or other of the coordinate sample sets, A or B, of the trajectories.

It is hence possible to calculate this derivative of information from the samples taken from the trajectory coordinates of objects we have tracked.

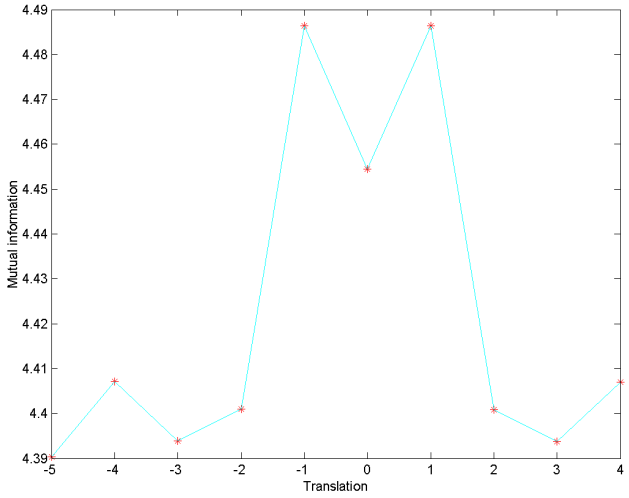


Figure 1: **Mutual Information versus Translation:** *The translation is in time with a coordinate point of a trajectory being given for each point in unit time.*

3 Experiments

3.1 Testing the algorithm

We can check that the above theory works by plotting the mutual information and the derivative of mutual information for increasing translations in time in each direction, positive and negative, for a given trajectory. Here we are comparing a dataset to itself in order to verify that, on average, the derivative of mutual information has a zero crossing at the point where the trajectories being compared are identical. In addition, the mutual information should in theory be at a maximum at this point, that is where $T = 0$.

The results in figures 1 and 2 are taken from a trajectory of 20 coordinate points where all, or as many as are available, of the data points are used to calculate the stochastic approximation to the real mutual information. The learning rate, λ in (1), is of no significance here as we are not trying to find a maximum yet, simply illustrate where it is expected to lie.

3.2 Aligning trajectories by maximising mutual information

3.2.1 Simple experiment on real data

A simple experiment using the trajectories of two cars, the tracked coordinates of which are shown in figure 3 being tracked illustrates how this method can reveal the phase differences in the motion of objects by locating the maximum of

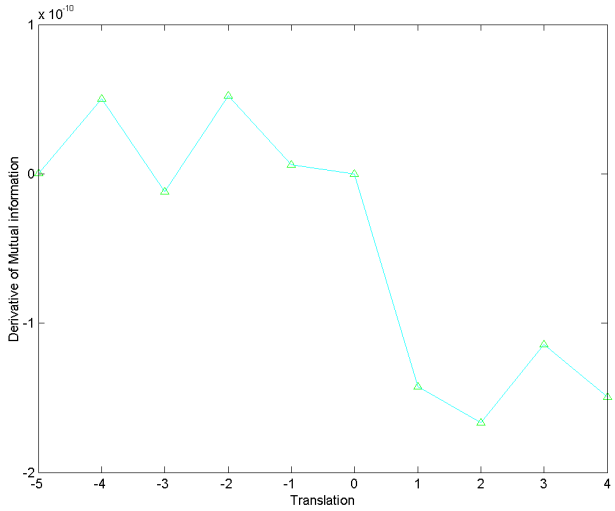


Figure 2: **Derivative of Mutual Information versus Translation.**

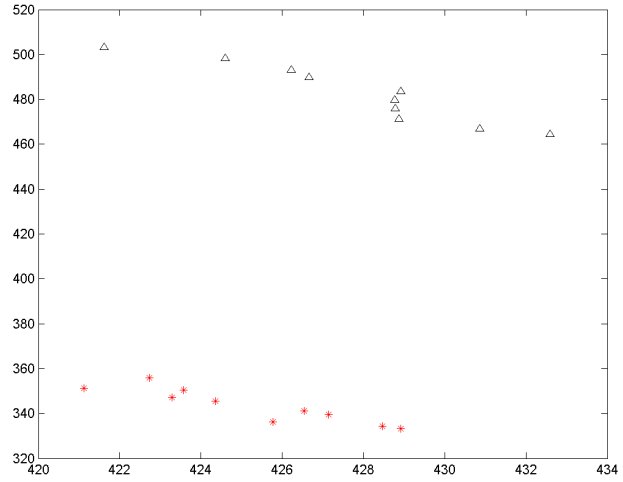


Figure 3: **Tracked objects relative trajectories**

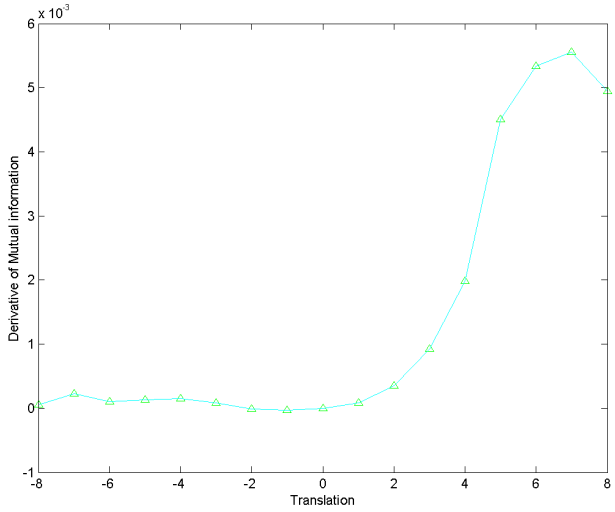


Figure 4: **Derivative of Mutual Information versus Translation for tracked trajectories. (Zero crossings correspond to maxima and minima of mutual information.)**

mutual information. Here, we take 10 points measured from the trajectories of the objects and translate in time these point sets, for each translation calculating the derivative of mutual information and the stochastic approximation to the mutual information.

It can be seen from figures 5 and 4 that the maximum of mutual information is located at a shift in the first trajectory by 7 units of time. This is corroborated by the fact there is a zero crossing of the derivative at that point also. Note the same is true for the minimum of mutual information occurring at a translation of 1 in trajectory 1.

3.2.2 Aligning Trajectories which have a phase lag

The above experiments show that the theory outlined in the previous sections can detect the maxima of mutual information. In reality the situation may arise where the mutual information approximation from the coordinates without any translation in time does not reflect the fact that there is a definite linked motion taking place e.g. balls bouncing out of phase etc. To this end it is required to test the algorithm to ensure it is possible to identify such a scenario.

A set of data is analysed where a distinctly similar relationship between the tracked coordinate sets occurs but with a phase change. The trajectory is shown in figure 6 and the change in mutual information with translation is shown in figure 7.

The gradient descent method for finding the optimum mutual information

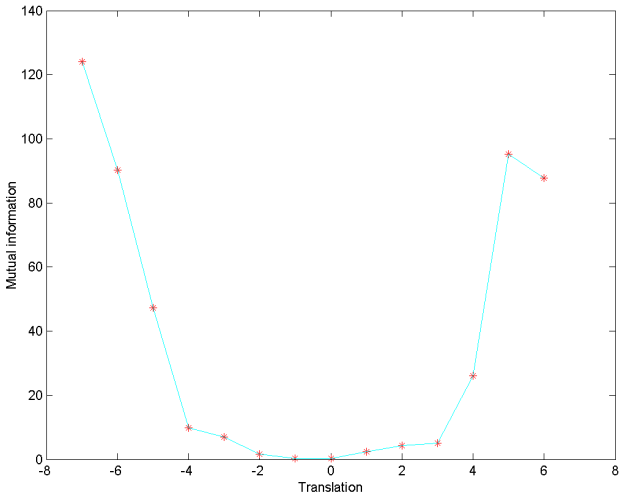


Figure 5: **Mutual Information versus Translation for tracked trajectories**

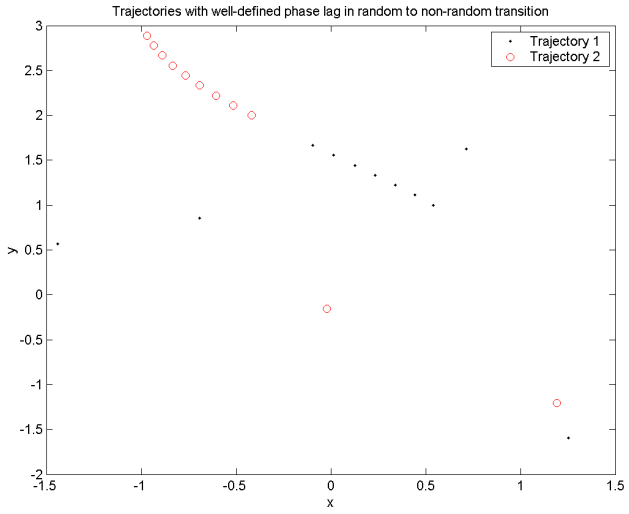


Figure 6: **Trajectories which show a distinct phase lag between the random and non-random elements**

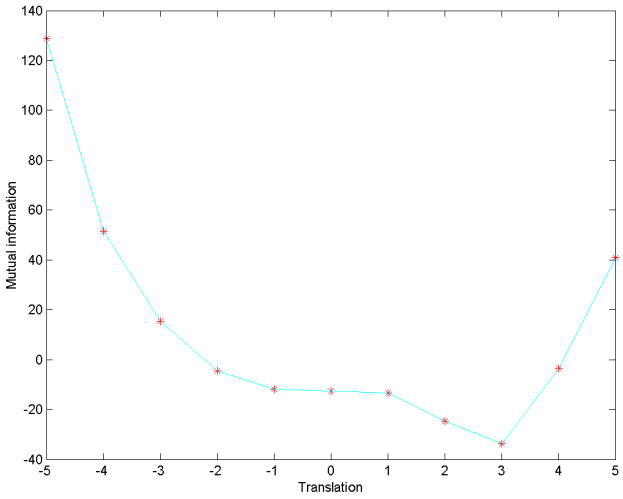


Figure 7: **Trajectories which show a distinct phase lag between the random and non-random elements**

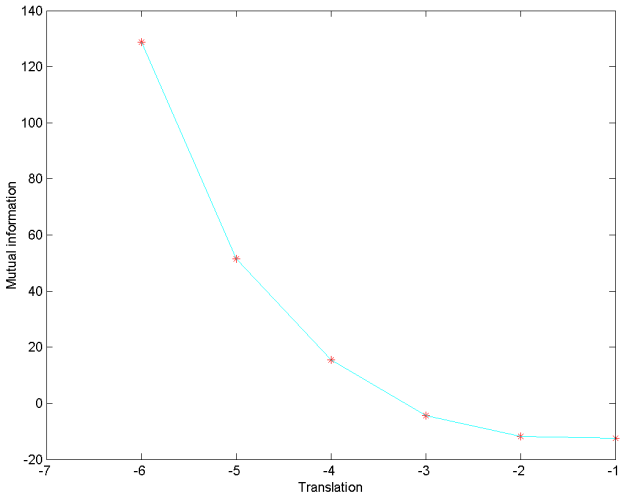


Figure 8: **Gradient descent algorithm convergence**

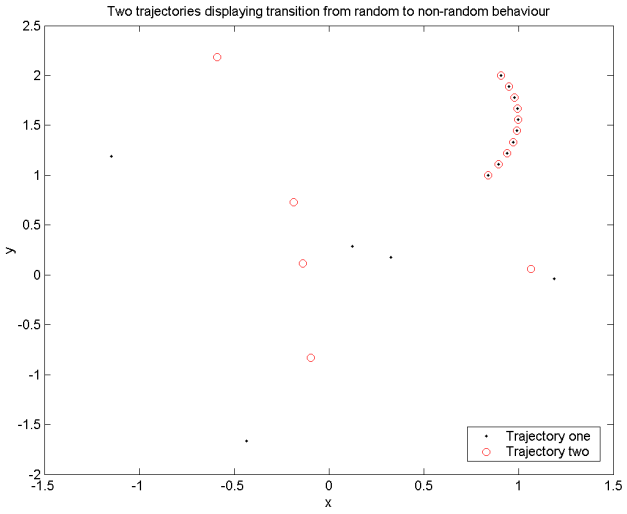


Figure 9: **Trajectories illustrating the concept of the change in behaviour from random (scattered points) to highly correlated (curved section).**

using the derivative of mutual information (1) is also utilised to find the corresponding maximum of mutual information as shown in figure 7. The result is that with a learning rate $\lambda = 1$, for a trajectory set consisting of 15 sampled coordinates, the algorithm takes 5 iterations to converge on the correct solution starting at $T = 0$ and finding the maximum of mutual information at $T = -5$, as shown in figure 8.

3.2.3 Detecting changes in randomness of motion

For surveillance purposes, for example if we were monitoring cars chasing one another on a motorway, it is necessary to determine whether mutual information can reveal when the motion of two objects under observation becomes nonrandom. This moment will probably correspond to the point in time where the motion becomes interesting from a surveillance perspective.

Figure 9 shows clearly the kind of motion in mind for this experiment. Of course, it is unlikely that exactly the same motion will take place at the point where the change from random to non-random takes place as we have synthesised here. However this does not affect the calculation of mutual information as it is based on a probability density estimation and not a pure measure of relative coordinates.

Figure 10 show the results of this experiment. Conceptually what is happening is that, as a translation in time takes place, different sections of the trajectories are being compared such that, where the mutual information increases markedly i.e. around $T = 5$ and around $T = -5$ the random element of

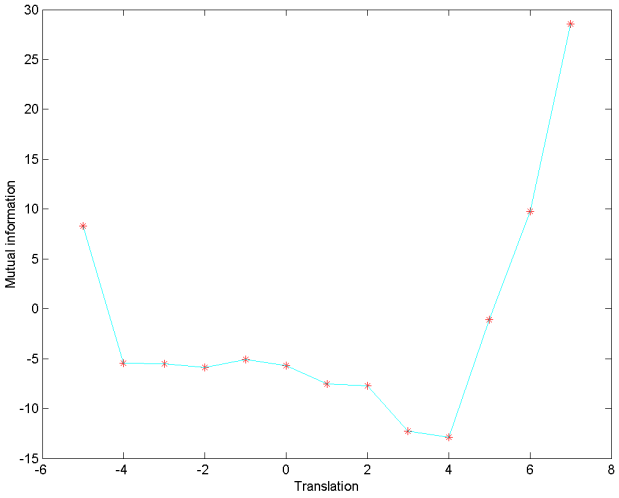


Figure 10: **Mutual Information versus Translation for tracked trajectories where the relative behaviour of the trajectories changes markedly from random to non-random.**

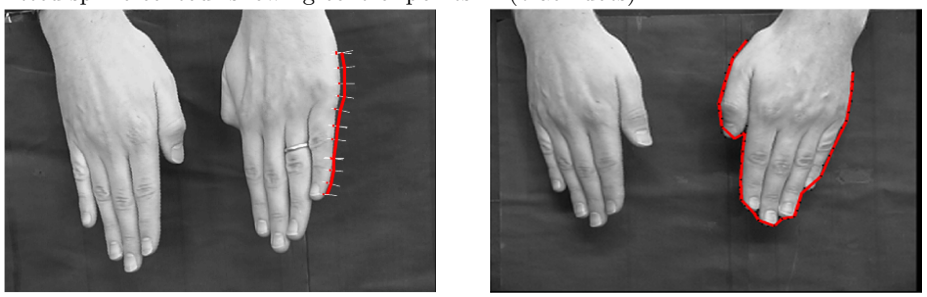
one trajectory set is being eliminated with respect to the non-random section which is now dominating the calculation and hence the mutual information is increasing in recognition of this fact. Hence it is possible to say that between $T = 4$ and $T = 6$ there is a significant change in the mutual information of the two trajectories. If the trajectory points are counted in figure 9, it is clear that at time $t = 5$ in sampling terms, the motion becomes coupled.

It is worth noting that this method may not work if the translation T is such that it causes the calculation to involve a larger component of randomness than non-randomness. There may be an instance where, because the non-random element of the two trajectories is significantly smaller in proportion to the random section or less strongly correlated section, the mutual information calculated may not extend beyond a threshold which has been set.

4 Car tracking sequence

The data used so far to verify the effectiveness of the algorithms presented has, in the main, been tested on synthesised but nonetheless realistic data. In reality, due to the less-than-perfect nature of the tracking technology we have at our disposal, there will be noise in the signals and perhaps this could affect the identification of joint structure in the respective datasets. In any case a system is built for real use and it is important to be tested.

Figure 11: (*left to right)* searching along normals for image features and complete fitted spline contour showing control points in (black dots)



4.1 Contour tracking

Kass et al. introduced Kalman snakes which uses numerical integration of the Lagrangian differential elastodynamics equations [6]. Dynamic contours evolved from the snake [7]. The shape-space method [8] is particularly effective at limiting the degrees of freedom of a contour such that arbitrary manipulations produce shapes which have at least a reasonable likelihood of fitting the image object. The use of splines is desirable as they provide a compact representation of complex curves where a knot vector and set of control points $\mathbf{Q}$ is all that is required to reconstruct the curve. A shape space is constructed allowing for affine deformations of a template contour (represented by control points, $\mathbf{Q_o}$), with a *shape-matrix*, $\mathbf{W}$, such that $\mathbf{Q} = W\mathbf{X} + \mathbf{Q_o}$, where $\mathbf{X}$ is the *shape-space vector*. It can be seen from (2) that the shape-space vector directly influences the shape of the resulting contour e.g. $\mathbf{X} = (0, 0, 1, 1, 0, 0)^T$ is the template doubled in size etc.

To track using this construction we sample along the normal vectors at intervals along the spline seeking image features, in this case edges as shown in figure 11. Edges are not the only image feature we could use, but it is more efficient to limit ourselves to a 1D search. A set of samples $\mathbf{r}_f(s)$ is found on the new image curve by searching in normal directions from points $\mathbf{r}(s)$ on the spline which represented the tracked image contour in the previous image. This measurement is effectively invariant to re-parameterisation and s spans the entire spline.

A recursive method is used to find the best fitting curve in shape space i.e. a new $\mathbf{X}$ is found for each image in the sequence. This curve is then used to initialise normal searches for the next frame and so, provided edges are found successfully, the contour is tracked throughout the sequence.

4.2 Experiment

We track the cars in the sequence using the contour tracker and condense the contour to the centroid and this is the data we sample throughout the sequence (see figure 12).

Figure 12: Tracking the centroid of a contour using a Kalman estimate, first and last images in sequence

The results of this experiment are as follows: a simple scan of the data with $\pm T$ ranging from zero to half the size of the dataset gives $\frac{dI}{dT} \approx 0$ meaning no convergence is possible using the gradient descent algorithm as the values of mutual information for each and every T is almost identical. For such a highly correlated motion this is what is expected. This data provides a good example of objects which are clearly coupled and given the evidence of the previous experiments on synthetic data we fully expect this fact to be readily identifiable in the results.

5 Experiments to determine changes from random to non-random motion

When using the metric of mutual information and the ability of the algorithm outlined above to detect changes in the randomnesss of the relative motion of objects in a decision making system, further applications are numerous. A clear application is the area of counter-terrorism where detection of suspicious behaviour in an airport lounge, for example, would be beneficial. Another interesting example application would be the classification of sports players positions based on their movement or even assessing their respective influences on the match. It is with this latter application in mind, although this experiment is of relevance to all areas of surveillance, that we use data taken from footballers training to test the theory.

The sequences have a number of interesting features. In particular, featured is examples of the following motions:

- totally unconnected relative motions e.g. many players training on separate parts of the pitch with no interaction between them
- motion which is linked although there is not necessarily any direct interaction taking place e.g. running together side-by-side but no passing of a ball between players

- distinct changes in relative motion from apparently random to linked e.g. when a ball is passed between two players who were previously moving independently
- change in motion from connected to unconnected, which is effectively a reversal of the situation in the previous point

Taken with the above considerations, it is of particular interest to discover whether the subtleties of such an intricate real-world situation as a sports match. However this setting is also ideal given that there is a finite set of rules which the agents in the scene are required to obey. Moreover these rules are generally enforced by the presence and actions of a referee. Of course, they can be broken but large deviations will not go unpunished e.g. punching an opponent. This is unlike a traffic scene where, with the exception of speeding, the rules will not be enforced except by the agents instinct for self-preservation, and so cannot be relied upon to the same degree. A warfighting manouevre will generally involve strict formation adherence but these rules are merely guides, albeit strong ones, and rules of engagement are flexible on instructions from command. To make the problem tractable it seems to be important to choose scenes where there are rules well-defined and generally well kept certainly at the concept-proving stage. This is not to say that the other scenarios are not ripe for automatic dynamic scene understanding application but the rules may well involve a high degree of effort in order to be learned. That is the purpose of a learning and decision making system.

5.1 Tracking by thresholding

There are some scenes for which contour tracking will not be appropriate. These include scenes where the target is in the far-field and so is very small in relation to the background or size of image as a whole. Such a situation means that the normal searches are unlikely to latch on to an image feature which relates to the target. As Blake et al. have shown it is possible to track succesfully in clutter by using appropriate filtering techniques and learned motion models. However, for a static camera with relatively stationary background as in the video data we consider for this application i.e. the training footballers data it is simpler to track using a learned background model. An additional feature which makes this approach feasible is the fact that the objects have specific features which do not change, for example, participants in a sports match generally wear similar, distinctive colours.

Firstly using a selection of images from the sequence we learn statistics for each pixel in the image and choose a threshold which will allow us to segment the image on the basis of the anticipated features of the targets, in this case dark clothes which stand out well against the background.

The average value is somewhere around the value for the background which in this case is grass and fairly bright. Occasionally a target will move across this region in the image and the intensity will drop rapidly. Hence, a strategic threshold is selected of around $(mean(pixelintensities) - 50)$. This provides a

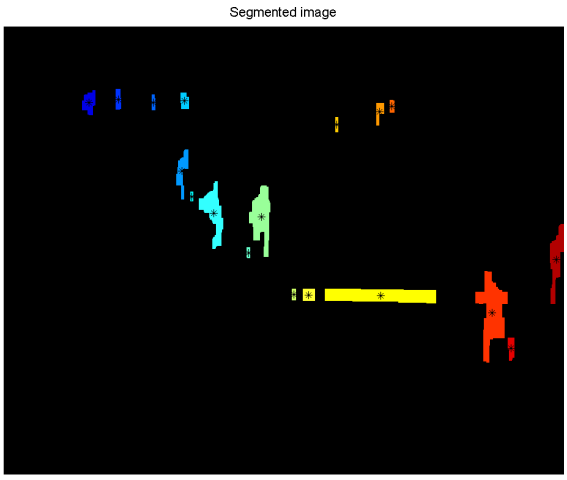


Figure 13: Classified foreground and background image

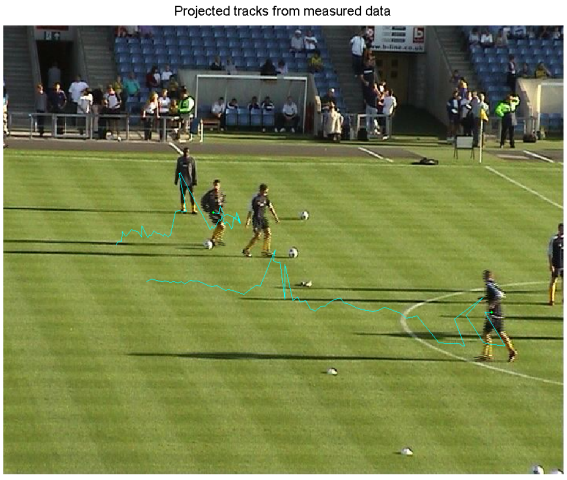


Figure 14: Object tracks superimposed on first image in the tracking sequence

basis for segmenting the image into regions corresponding to foreground (with a bias towards darker objects) and background: pixels with an intensity value below the threshold are chosen and given a value of 1, the others are given a value zero. In this way we get a black and white image showing regions of foreground and background. The results of this process are shown in fig. **??**.

After classification it is necessary to identify specific objects in the scene which we can segment as targets. This is done by applying a structuring element to the image to find target shaped objects and scanning the resulting image to identify the extent of the targets. The centroid of each area can then be found and identified as the target location at that point. The motion is small between consecutive frames and so the continuity of tracks is retained by looking for the target nearest to the previous recorded location.

Since we are interested in the interactions between two objects, we track the targets over the period of suspected interaction. Sample tracks are shown in fig 15

6 Discussion of results

Experiments have been performed which show how the mutual information method performs in the following situations

- where both trajectories are the same for test purposes
- where there is real data with no apparent structure
- when the trajectories have distinct coupling and with clear phase lag in one with respect to the other
- when the gradient descent algorithm is required to determine where the phase lag occurs
- where there is evidence of a change in behaviour from random to coupled motion, detecting that this occurs and where

Another scenario which has not been explicitly set out but would be of interest is where there is one object following the other, say cars on a highway chasing one another. In this case, the sampled track of the tailing car would not bear an exact resemblance instantly to the leading as there is drivers reaction times to be taken into account. There will be a combination of two of the situations outlined above: a computer vision surveillance system will be interested in identifying not only a change from random to coupled motion but also the fact there is phase between the trajectories could be of interest in classifying the motion as 'chasing' and not 'following' for example. This may be too subtle a distinction to be able to make using this method alone, however given that it is overlapping regions which allow the coupled motion sections to negate the effect of randomness we can see that this particular scenario identifies a certain degree of error if a system is trying to pinpoint exactly *when* the change from random

to coupled motion took place. This because where there is a phase lag, it is likely the mutual information will increase more rapidly with fewer iterations of the gradient descent algorithm since the algorithm will identify the overlapping regions of coupled motion sooner.

7 Conclusions

From the straightforward experiments we have performed in this report, it is clear there is a basis for using mutual information to determine when the motion of trajectories becomes coupled or changes to random. A decision process will require a threshold based on a value of mutual information to be valuable since a pure measure of untranslated mutual information would be misleading. There will be situations where the relative motion is out of phase and the gradient descent algorithm presented here using the derivative of mutual information will then be of use. This will enable a system, on the basis of incoming trajectory data, to determine if there is any phase lag which is causing the initial, untranslated in time signal to register a value of mutual information below an indicator threshold or if there is in fact no high degree of correlation between the objects' motion.

8 References

[1] Sébastien Gilles, Description and experiment of image matching using mutual information, Technical Report, Robotics Research Group, Oxford University, 1996.
[2] Paul Viola and William M. Wells, Alignment by Maximisation of Mutual Information, Int. Journal Computer Vision, 24(2) pp137-154, 1997.
[3] Duda and Hart, Pattern classification and scene analysis, J.Wiley pub.
[4] I.Rezek, P.Sykacek, S.J.Roberts, A comparison of Bayesian and maximum likelihood learning of coupled hidden Markov models, IEE proc. Science Technology and Measurement, vol.147, issue 6, p345-350.
[5] M.Li, P.Vitanyi, An introduction to Kolmogorov complexity and its applications,Springer-Verlag, New York, 1996.
[6] M. Kass et al., Snakes: active contour models, Proc. 1st Int. Conf. on Computer Vision, 256-268, London.
[7] R. Curwen and A. Blake, Dynamic contours: real-time active splines, Active Vision, ed. Blake, Yuille, MIT, 1993.
[8] A. Blake and M. Isard, Active Contours, 2nd-ed.,Springer-Verlag, London, 2000.

A Derivation of mutual information

The question that is being addressed in calculating mutual information is *how much information can a random variable X give about another random variable*

Y? Using the combinatorial approach of Li and Vitanyi [5], if X ranges over S_X and Y ranges over S_Y, and if the set of possible joint occurrences of events X and Y is not the Cartesian product $S_X \times S_Y$ then there is some dependence of X on Y or vice-versa.
The *conditional entropy* of Y given $X = a$ we define as $H(Y|a) = \log d(U_a)$. This suggests that the information in X about Y is

$$I(X : Y) = H(Y) - H(Y|X) \tag{7}$$

It is clear from this approach that $H(X|X) = 0$ and that $I(X : X) = H(X)$, hence this formulation is an assumption of uniform distribution of the probabilities.
Now if we let the probability of the joint occurrence of event X and Y be the *joint probability*, $p(X, Y)$, we get the following formulae for joint variables X and Y:

$$H(X, Y) = -\sum_{X,Y} p(X, Y) \log p(X, Y) \tag{8}$$

$$H(X) = -\sum_{X,Y} p(X, Y) \log \sum_{Y} p(X, Y) \tag{9}$$

$$H(Y) = -\sum_{X,Y} p(X, Y) \log \sum_{X} p(X, Y) \tag{10}$$

and therefore it can be shown,

$$H(X, Y) \leq H(X) + H(Y) \tag{11}$$

where there is equality if X and Y are independent.
In addition the conditional probability $p(Y|X)$ is defined by

$$p(Y|X) = \frac{p(X|Y)}{\sum_Y p(X|Y)} \tag{12}$$

We consider the *conditional entropy* of Y given X as the average of the entropy for Y for each value of X weighted by the probability of getting that particular value:

$$H(Y|X) = -\sum_{Y} p(Y|X) \log p(Y|X) \tag{13}$$

$H(Y|X)$ is a measure of how uncertain we are of Y on average when we know X, with

$$H(Y) = -\sum_{X,Y} p(X, Y) \log \sum_{X} p(X, Y) \tag{14}$$

substituting the formula for $p(Y|X)$,

$$H(Y|X) = H(X, Y) - H(X) \tag{15}$$

which can be written as

$$H(X, Y) = H(X) + H(Y|X) \tag{16}$$

This can be interpreted as, *the uncertainty of the joint event* (X, Y) *is the uncertainty of* X *plus the uncertainty of* Y *given* X.
Combining the above equations gives $H(Y) \geq H(Y|X)$ and so

$$I(X : Y) = H(Y) - H(Y|X) \tag{17}$$

is the information in X about Y.
The *mutual information* is then

$$I(X : Y) = H(X) + H(Y) - H(X, Y) \tag{18}$$