

# Multi-resolution template kernels

C. J. Needham and R. D. Boyle  
School of Computing  
The University of Leeds  
Leeds, LS2 9JT, UK  
{chrisn,roger}@comp.leeds.ac.uk

## Abstract

*Domains in which shapes of objects change rapidly and significantly are a challenge for existing representation techniques: sport is a good example of this. We present a texture-based approach that copes with these problems in addition to resolution variation. A set of exemplar poses are learned from subsampled example images of the target object, creating a set of multi-resolution template kernels which when convolved with the image respond suitably. This technique may then be used in established tracking algorithms (e.g. CONDENSATION [4]). We demonstrate the technique in two domains, and suggest a Markov approach using it to model behaviour.*

## 1. Introduction

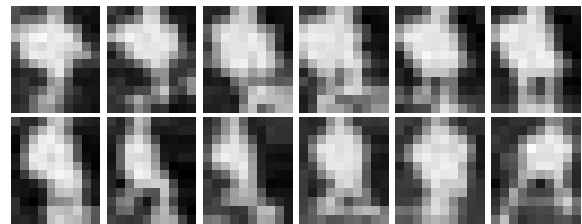
Feature description methods in computer vision are used for a variety of tasks. Being able to describe objects' features or characteristics allows them to be categorised or recognised. There exist many objects for which a shape model in the form of a Point Distribution Model [3], Active Appearance Model [2], Spline model [1] or 3D skeletal model [6] either fails, or is over complicated or computationally expensive. This paper presents a simple texture-based shape descriptor for object localisation.

Describing the features of a sports player is not a trivial task. Sports players' poses and shapes vary enormously. Creating a description to capture the range of configurations of a player's pose is the motivation behind this research.

## 2. Multi-resolution kernels

Details of implementation are discussed in relation to the first example application: localisation of soccer players. A set of (120) hand annotated bounding boxes containing players is obtained as a training set. A colour foreground/background model is used to extract the players, which assigns to each pixel a probability of being fore-

ground (belonging to a player). The monochrome 'foreground probability image' within the bounding box of a player, of size  $w \times h$ , is scaled down so that the resolution becomes  $a \times b$ , by sub-sampling using a Gaussian window function (Figure 1). This removes the obstacle of the differences in scale between players close to the camera, and those further away. This image can now be represented by  $\mathbf{g}$ , an  $a \times b$  vector, such that  $\mathbf{g}_j$  equals the value of the foreground probability at pixel  $j$  (for  $j = 1, \dots, ab$ ). A normalised feature vector,  $\mathbf{f}$ , of length  $ab$  can be created by taking  $\mathbf{f}_j = \frac{\mathbf{g}_j}{\sum_{i=1}^{ab} \mathbf{g}_i}$ .



**Figure 1. Sub-sampled images (10x12 pixels) of soccer players.**

This normalised feature vector  $\mathbf{f}$  may be represented as a kernel  $\mathbf{k}$  with  $\mathbf{k}_j = \mathbf{f}_j - \frac{1}{ab}$ , and clipped such that  $\mathbf{k}_j < \frac{1}{ab}$ . Details of results of unclipped kernels are presented in [5]. The image region within the bounding box may now be evaluated by convolving with the kernel  $\mathbf{k}$ . The test vector  $\mathbf{g}$  represents the sub-sampled (scaled down) image information from the test image within the bounding box. Calculating  $\sum_{j=1}^{ab} \mathbf{k}_j \mathbf{g}_j$  gives the response of kernel  $\mathbf{k}$  to the image region represented by  $\mathbf{g}$ .

Three methods for incorporating the convolution kernel into a tracking scheme are now discussed and evaluated:

- A **PCA model.** Generate an example from the model.
- B **An example.** Randomly choose an example from the training set.
- C **Exemplars.** Cluster training set into exemplars and apply each one.

For Method A, an example  $\mathbf{k}$  is generated from the model, and convolved with the test vector  $\mathbf{g}$  ( $\sum_{j=1}^{ab} \mathbf{k}_j \mathbf{g}_j$ ). For Method B, an example  $\mathbf{k}$  is chosen from the training set, and convolved with the test vector  $\mathbf{g}$  ( $\sum_{j=1}^{ab} \mathbf{k}_j \mathbf{g}_j$ ). For Method C, the training set has been clustered using the k-means algorithm. Figure 2 shows the k-means cost for different numbers of clusters. Five clusters are chosen, as the steep reduction in costs begins to slow at this number of clusters. The five template kernels ( $\mathbf{t}^0 \dots \mathbf{t}^4$ ) representing the cluster centres are shown in Figure 3. Each of these is convolved with the test vector  $\mathbf{g}$ , and the largest chosen. ( $\max_i (\sum_{j=1}^{ab} \mathbf{t}_j^i \mathbf{g}_j)$ ).

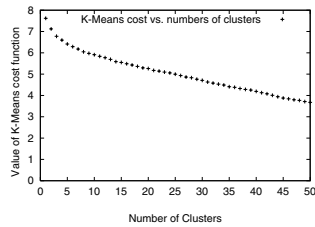


Figure 2. The k-means cost function for increasing number of clusters.

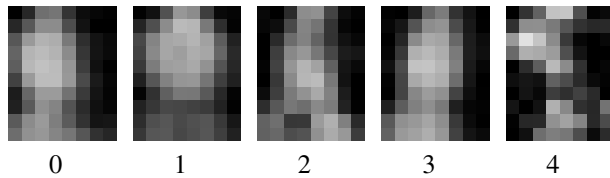


Figure 3. The five footballer template kernels.

## 2.1. Evaluation of the three methods

Each of the methods (A, B & C) described above are evaluated on a set of (124) positive examples of bounding boxes well-centred on a player (Figure 4(a)), and also on a set of (98) negative examples of bounding boxes not well-centred on a player (Figure 4(b)). For each approach the

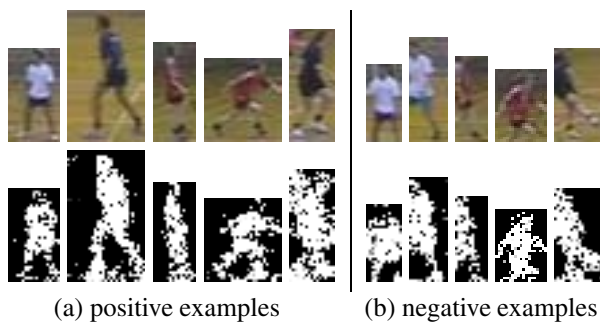


Figure 4. Positive and negative examples of BBs, and corresponding FG/BG images.

foreground extracted image within the bounding box is sub-sampled to an  $8 \times 10$  image, which is then convolved with the kernels.

The negative examples all feature a player, but have chopped off the player's legs, the right-hand side of the body, the player's head, or are judged to be not well fitting to the player's image. Ideally, comparing the difference between the means for the positive and negative examples for each method will identify the method for which the separation between gaining a high response (for a positive example) from the kernel, and a poor response (for a negative example) is greatest. Histograms of the results of each method on the two sets are shown in Figure 5. A simple analysis of the means of the three methods on both sets of test images (Table 1) shows that the difference between the means is small for method B. For both methods A and C a similar separation between the means is observed, 0.100 and 0.089 respectively. Method C will be employed in the remainder of this work. The advantage of this method over method B is clear. Over method A it has the advantage that an example (which may not be representative of any of the training examples) from a PCA model does not need to be generated for each comparison. An added advantage is that pose identification/categorisation may be possible, as is demonstrated in Section 3.

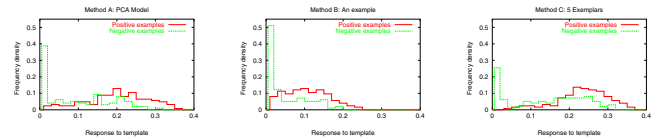


Figure 5. Graphs of the response of each of the methods A, B, and C on a set of positive examples of bounding boxes well-centred on a player, and a set of negative examples of bounding boxes not well-centred on a player.

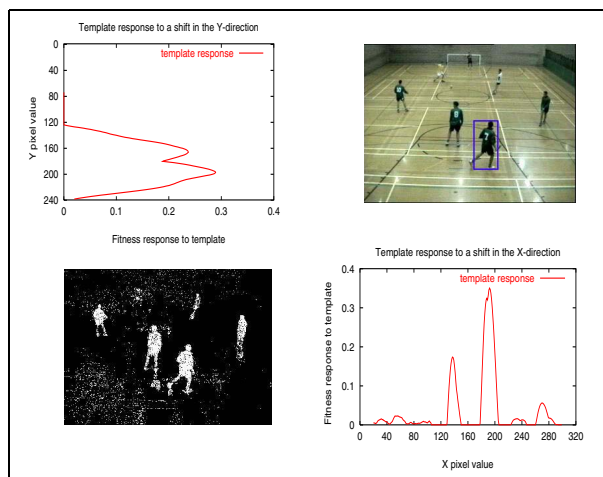
| Method                               | A                | B                | C                |
|--------------------------------------|------------------|------------------|------------------|
| mean (and s.d.) of positive examples | 0.190<br>(0.081) | 0.096<br>(0.056) | 0.222<br>(0.068) |
| mean (and s.d.) of negative examples | 0.090<br>(0.091) | 0.043<br>(0.056) | 0.133<br>(0.101) |
| difference in means                  | 0.100            | 0.053            | 0.089            |

Table 1. Comparison of methods A, B, and C as a template fitness function.

## 2.2. Multi-resolution template kernel evaluation

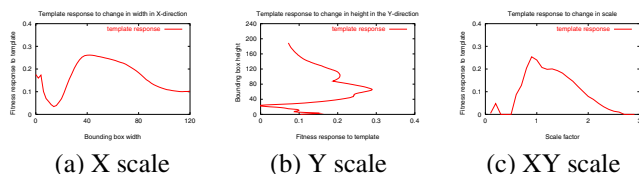
A scene from a soccer game is used to demonstrate the robustness of locating a suitable bounding box around a player. In particular, the sensitivity of the bounding box

is evaluated with respect to vertical positioning, horizontal positioning, changes in width, changes in height, and changes in scale. Figure 6 shows the image used, and also the foreground segmented from the background. It is this monochrome image to which the templates are applied. The five template kernels shown in Figure 3 are each convolved with the image to gain a response which can be used as a *fitness function* in the CONDENSATION tracking scheme.



**Figure 6. Templates' fitness functions illustrating the sensitivity/robustness to vertical and horizontal positioning of the bounding box around 'Player 7'.**

Inspection of the image reveals that 'Player 7' appears at position  $(x, y) = (184, 190)$  with a bounding box of width 38 pixels and height 75 pixels. This bounding box is marked on Figure 6. The position  $(x, y)$  is taken as the mid-point of the base of the bounding box. Each bounding box is subsampled to form an  $8 \times 10$  image (arbitrarily chosen), to which the template kernels are applied. Ideally the fitness function should peak at the value which most agrees with the player's position, should be piecewise continuous, and give a small response away from the player's position. The degree of sharpness of the function, in the area surrounding the peak will affect the performance when employed in a tracking scheme.



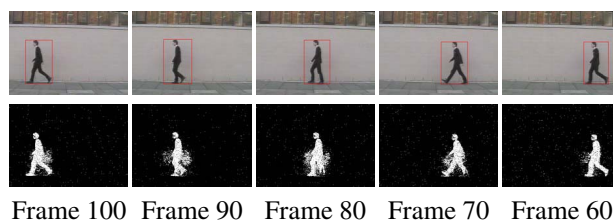
**Figure 7. Analysis of templates' fitness with different scale bounding boxes.**

Figures 7(a), 7(b), and 7(c) demonstrate the template kernels' sensitivity and robustness to changes in width, height

and scale (fixed aspect ratio) respectively. Further discussion about the properties of the template kernels may be found in [5]. This brief evaluation has demonstrated the performance of the multi-resolution template kernels, and the shape of the distribution of the fitness function has been shown on an example soccer player in an example scene, when the bounding box size or shape varies.

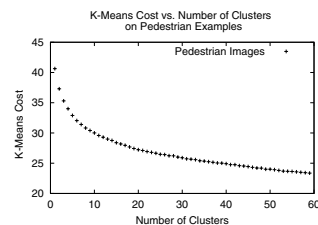
### 3. Pedestrian pose analysis

This section examines the use of multi-resolution template kernels for analysing the patterns of movement through different poses of pedestrians walking. Figure 8 shows example footage of a pedestrian walking across the scene from right to left, along with a foreground extracted images. From 18 sequences of pedestrians walking across



**Figure 8. Pedestrians: example footage, tracking and foreground extraction.**

the scene, a set of 1206 training examples are obtained. It is straightforward to construct a bounding box around each pedestrian, and to create a fixed length feature vector representing the pedestrian by subsampling the image. In this case a 180 dimension vector is formed ( $12 \times 15$  pixels). These 1206 training examples are clustered using the k-means clustering algorithm. The results of the k-means cost (RMS distances from each vector to closest cluster centre) versus number of clusters is plotted in Figure 9. This graph



**Figure 9. K-means cost of different numbers of clusters of pedestrian kernels.**

is used to decide that it is sensible to use six clusters to represent the data, as the curve becomes less steep. This is supported by the fact that including more templates was judged not to produce additional templates which were noticeably different. The six clusters (each of which contain between 75 and 370 of the original 1206 training examples)

revealing the six templates which can be used to identify the different poses. Images representing the feature vectors at the centre of the clusters are shown in Figure 10. Templates 1 and 5 represent pedestrians when striding out with legs apart, templates 0 and 4 represent when the legs are reasonably close together and some shadow is cast, template 2 represents the pedestrian with legs close together, and template 3 looks least like a person and is formed from the start of image sequences as the pedestrians enter the scene from the right.

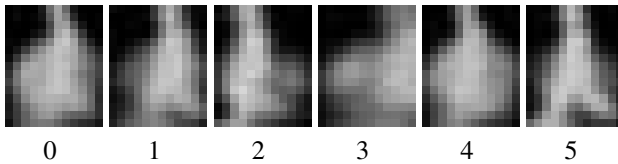


Figure 10. Six pedestrian template kernels.

### 3.1. Pose transitions

Given the six template kernels, each of the 1206 examples can now be convolved with each of these kernels, and assigned as closest to the one which returns the highest score. Since these example images are taken as part of a sequence, it is possible to create a transition matrix of the movements from one pose/template to the next. From the 18 sequences 1188 pose transitions are observed ( $1206 - 18 = 1188$ ). Figure 11 illustrates graphically the frequency of transitions between poses observed in the training sequences.

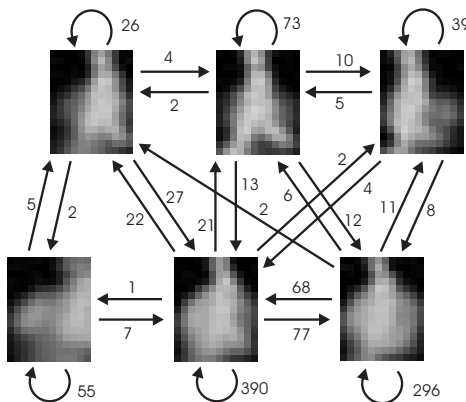


Figure 11. Pedestrian pose transition graph.

Markov models (first order) allow an outcome of an independent process to depend *only* on the state at the generative time step. Thus, by analysing the changes between poses (states) in the training set (18 sequences in total; 3 sequences of 6 pedestrians), a Markov model can be created. A stochastic matrix,  $P = p(i, j)$ , of transition probabilities is formed based on the frequency of transitions from one pose/state to another observed in the training data. If the

frequency of transitions from state  $i$  to state  $j$  is denoted  $freq(i, j)$ , then:

$$p(i, j) = \frac{freq(i, j) + 1}{\sum_k freq(i, k) + N} \quad (1)$$

where  $N$  = the number of states, here  $N = 6$ . This preserves the need for  $\sum_k p(i, k) = 1$  to create a stochastic matrix, and allows for all possible transitions to occur. It is now possible to:

- generate a sequence, by taking a probabilistic walk through the model
- evaluate how likely a test sequence of a pedestrian is (in comparison to the training sequences)

This allows the state or pose of the pedestrian at the next time step to be predicted. Within the stochastic sampling of the CONDENSATION tracking, or indeed any ‘predict and sample’ tracking scheme, this would allow the samples to be partitioned, so that a corresponding percentage of samples of each pose type could be used for tracking. i.e. tracking a pedestrian at time  $t$ , who is in pose 1, it would be expected at time  $t + 1$  to be in pose 0 or 1, where most of our samples can be directed, with a few for the remaining poses/states.

## 4. Summary

A novel fitness function for use in a CONDENSATION based tracking framework has been presented. A set of exemplar poses are learned from subsampled example images of soccer players, creating a set of multi-resolution template kernels which when convolved with the image respond suitably. This assists with the localisation of target players in the tracking application. The same technique has also been applied to pedestrians, identifying their poses as they walk.

## References

- [1] A. M. Baumberg and D. C. Hogg. An efficient method for contour tracking using active shape models. In *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 194–199, Austin, Texas, 1994.
- [2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. European Conf. on Computer Vision*, volume 2, pages 484–498, 1998.
- [3] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conference*, pages 9–18, 1992.
- [4] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. Computer Vision*, pages 343–356, Cambridge, UK, 1996.
- [5] C. J. Needham. *Tracking and modelling team game interactions*. PhD thesis, School of Computing, The University of Leeds, UK, 2003.
- [6] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conf. on Computer Vision*, pages 702–718, 2000.