Extraction of Motion Data from Image Sequences to Assist Animators

David P. Gibson, Neill W. Campbell, Colin J. Dalton and Barry T. Thomas
Department of Computer Science
University of Bristol
Bristol, BS8 1UB, UK
gibson@cs.bris.ac.uk

Abstract

We describe a system which is designed to assist animators in extracting high-level information from sequences of images. The system is not meant to replace animators, but to be a tool to assist them in creating the first 'rough-cut' of a sequence quickly and easily. Using the system, short animations have been created in a very short space of time. We show that the method of principal components analysis followed by a neural network learning phase is capable of motion tracking (even through occlusion), feature-extraction and gait classification.

We quantify the results, and demonstrate the system tracking horses, birds and actors in a film. We demonstrate a system that is powerful, flexible and, above all, easy for non-specialists to use.

1 Introduction

Many attempts have been made to extract data from video and film in a form suitable for use by animators and modellers. Such an approach is attractive, since motions and movements for people and animals may be obtained in this way that would be difficult using mechanical or magnetic motion capture systems. Visual extraction is also appealing since it is non-intrusive and has the potential to capture, from film, the motion and characteristics of people or animals long dead or extinct.

Almost all attempts to perform visual extraction have been based around bespoke computer vision applications which are difficult for non-experts to use or adapt to their own needs. This paper presents a generic approach to extracting data from video. Whilst our approach allows low-level information to be extracted (information more usually obtained by motion tracking using template tracking or classification using template matching) we show that higher-level functionality is available also. This functionality can be utilised in a manner that requires little knowledge of the underlying techniques and principles. Our approach is to approximate an image using principal component analysis, and then to train a multi-layer perceptron to predict the feature required by the user. This requires the user to hand-label the features of interest in some of the frames of the image sequence. One of the aims of this work is to keep to a minimum the number of frames that need to need labelled by the user. The trained multi-layer perceptron is then used to predict features for images that have never been labelled by the user.

BMVC 2000 doi:10.5244/C.14.31

Other attempts to extract useful information from video sequences include the use of edge-detection and contour or edge tracking [7, 2, 4], template matching [10] and template tracking. All such systems work well in some circumstances, but fail or require adaptation to meet the requirements of new users. For instance, in the case of template tracking, the user needs to be aware of the kinds of features that can be tracked well in an image and also choose a suitable template size. This is not a trivial task for non-specialists.

2 Method

The main steps in using our system are detailed below:

- The user selects the sequence (or set) of images for which they wish data to be extracted from. This may well comprise of several shorter clips taken from different parts of a film.
- These images have some pre-processing performed on them (principal components analysis) to reduce each image to a small set of numbers.
- The user decides what feature(s) they wish to extract and labels this feature by hand in a fraction of the images chosen at random. The labelling process may involve clicking on a point to be tracked, labelling a distance or ratio of distances, measuring an angle, making a binary decision (yes/no, near/far etc.) or classifying the feature of interest into one of several classes.
- Once this ground-truth data is available, a neural network is trained to predict the feature values in images that have not been labelled by the user.

The two main steps of principal components analysis and neural network training are discussed next.

2.1 Feature Extraction

Principal components analysis (also known as eigenvector analysis) has been used extensively in computer vision for image reconstruction, pattern matching [8] and classification [1].

Given the i^{th} image in a sequence of images, each of which consists of M pixels, we form the vector x_i by concatenating the pixels of the image in raster scan order and removing the mean image of the sequence. The matrix X is created using the x_i's as column vectors. Traditionally, the principal modes, q_i, are extracted by computing

$$XX^T q_i = \lambda_i q_i \tag{1}$$

where λ_i's are the eigenvalues, a measure of the amount of variance each of the eigenvectors accounts for. Unfortunately, the matrix XX^T is typically too large to manipulate since it is of size M by M. Such computation is wasteful anyway since only N principal modes are meaningful, where N is the number of example images. In all our work $N << M$. Therefore we compute:

$$X^T X u_i = \lambda_i u_i \tag{2}$$

Figure 1: The first eigenvector of a sequence of frames from a film. The images show a large negative value of the eigenvector (Left) and a large positive response (Right) applied back onto the sequence mean image. The eigenvector appears to be mainly encoding whether the piano-player's body is facing towards the camera, or to the side.

and we can obtain the q_i's that we actually require using:

$$q_i = X u_i \tag{3}$$

In practice only the first P modes are used, $P < 30 << N$.

The principal mode extracted from a short film clip is shown in Figure 1 and is used later to help an animator to construct a cartoon version of the clip.

It is tempting to think that such modes could be used directly to predict, say, the rotation of the man's shoulders. However, the second mode also encodes information about shoulder movement and it is only by combining information from many modes that rotation can be reliably predicted.

2.2 Training the System

Once the response of an image to the principal modes are extracted, these are fed into a feed-forward multi-layer perceptron. The neural network used here has three layers; an input layer containing P nodes, a hidden layer, and an output layer typically containing one node. The multi-layer perceptron allows non-linear associations to be made between the input and output. Once the animator has a feature that they wish to track or predict, they label a few dozen frames of the sequence, chosen at random. The response of these images is used as an input to the network and the hand-labelled ground-truth used to train the output. Backpropagation [9] and conjugate gradient descent is used for training. Once training is complete then the rest of the frames in the image are automatically labelled using this network. The animator views the result and either accepts it, or takes steps to improve the results. This attempted improvement can come about from several different options:

- Random assignment of weights in the network means that, even with the same information, simply re-training may lead to an improvement.

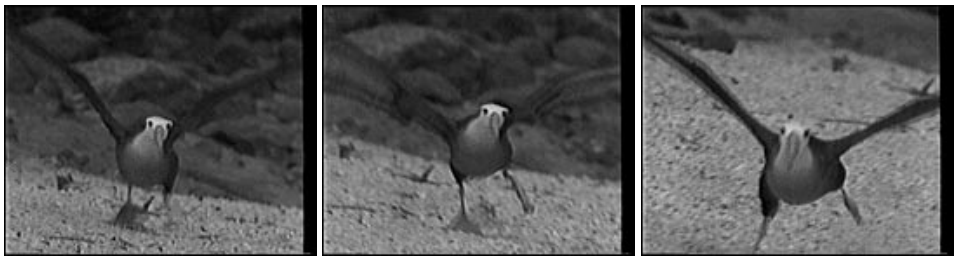

Figure 2: Three images in the takeoff sequence. Image 0, 50 and 200. The images also show the rock used to perform motion tracking (seen between the legs of the bird in the first image).

- The addition of more examples to the training set.
- The use of more principal components as input to the network. By default 5 are used.
- Changing the number of hidden units in the network. By default, the number of hidden units is set to two-fifths of the number of input units.

The training of such small networks takes only a few seconds and experience has shown us that the animator quickly learns how to improve results without an in-depth understanding of neural networks.

3 Results

In this section we look at three practical examples of our system being used for animation tasks. In the first example we show the creation of a short cartoon of a bird, based on a video of an albatross taking off. Our system is used to provide the animation data required for, amongst other controls, the motion of the wings. In our second example we show the production of a cartoon sequence, once again based on a film clip. This time lip-synching information is extracted from the two characters, as well as information used for rotating and translating their heads. The third sequence involves tracking the gait of a horse. At the end of the section we make some quantitative observations on the performance of the system. The reader is also referred to [3] for a more detailed description of this work.

3.1 Animating an Albatross

In this example we show how our technique was used to aid an animator in producing a short sequence of a bird attempting to take off. Since this particular species of bird finds it very difficult to become airborne, the animator was especially interested in the movement of the birds wings. The angle of the wings doesn't follow a simple sine-wave sequence, and it is these differences that give realism and 'life' to the animation. Three frames of the sequence are shown in Figure 2.

The animator labelled 19 frames of a 200 image sequence with two numbers; one each for the angle of the left and right wings respectively. Notice that the actual *position*

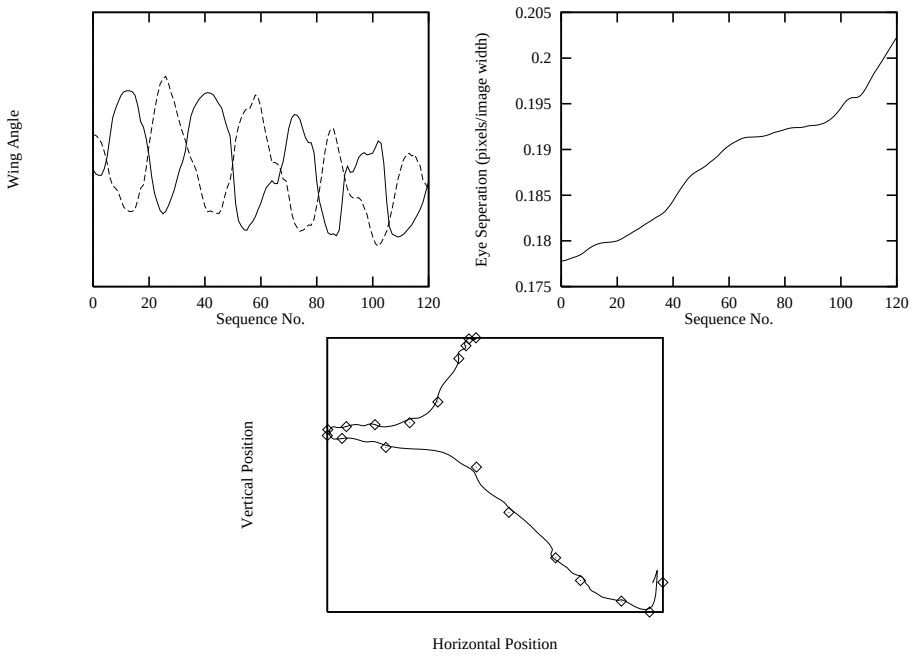

Figure 3: (Top Left) Tracking the angle of a bird's wings: left wing (dashed line), right wing (solid line). Since the bird is attempting to take off, the right and left wings are not moving in a symmetric manner. (Top Right) Graph showing the distance between the bird's eyes. Due to perspective, as it gets nearer to the camera the apparent distance increase. The bird is seen to accelerate, slow down, and then accelerate again. (Bottom) Rock tracking results (solid line) and training data (points). Due to camera movement, a rock lying on the beach behind the bird has an apparent motion. Tracking occurs successfully despite the large amount of occlusion present in the sequence.

of the wings does not need to be labelled, unlike a contour or edge tracking system. Two neural networks were then trained to predict the angle of the wings for the rest of the images. The results of this process can be seen in Figure 3. It is clear from the graph that the wings are moving not in a simple, symmetric manner but in a way that would be difficult to keyframe manually. The amplitude of the flapping decreases slightly over time and the bird stumbles at around frame 70, changing the pattern as it fights to regain its balance. This stumble by the bird is also clearly shown in Figure 3. This shows the distance between the birds left and right eyes. As the bird approaches the camera this distance increases. The animator extracted this information in a similar manner to the wing angle information, and used it to set the depth of the bird in the animation. The slope of this graph gives an indication of the speed the bird which accelerates slightly, stumbles and slows down, and then accelerates again.

The bird image sequence was taken with a hand held camera, and the animation team were interested in emulating this effect. Since both the camera and bird are moving, a static object was required in the background to use as a reference point. The only object of

Figure 4: Tracking the piano player's mouth (white dot) using principal components analysis computed using the uncropped images.

this type available was a small rock lying on the beach behind the bird. A major problem, however, is that the rock is occluded by the bird in around half the frames in the sequence. The animator used our method to extract the horizontal and vertical position of the rock, using the same set of 19 training frames used above. The tracking results are shown in Figure 3, along with the training data used. The success of the tracking is remarkable as shown in Figure 3. Due to the amount of occlusion present, a more traditional template tracking approach fails.

3.2 Singing Heads

In this, the second practical demonstration of our system, we show how an animator created a cartoon version of a short section of a movie. As in the case of the albatross animation, it is a few distinctive movements that the animator is interested in, rather than tracking the absolute positions of the heads. In the clip, the piano-player is talking and singing. He turns his head left (to play the piano), right (to talk to a woman) and lifts his head to sing. The animator chose to track the piano-player's mouth to obtain the horizontal rotation of his head, his nose-tip to obtain the vertical movement of the head, his bowtie to obtain the rotation of the shoulders and his lip separation for lip-synching purposes.

The first three of these features turned out to be easy to track, good performance resulting after labelling only 25 frames of the entire 424 frame sequence. The extent to which the piano-player's lips are open, however, was more problematic, caused by three problems.

The first problem is that the first principal components of the sequence encode rather coarse features. Reconstruction of an image from these modes results in a low-pass (blurred) version of the image and using a large number of modes is unappealing since the size of the neural network would be increased accordingly. Therefore, it was necessary to use the approach taken in [6] which is to recalculate the principal components for a smaller area cropped out of the original image. This is done automatically by the system, which tracks the mouth using the original set of principal components, and extracts a sub-region around the tracked point. Tracking of these points is demonstrated for three frames shown in Figure 4. These sub-regions are then used, and principal component analysis re-applied. The eigenvectors obtained are now solely concerned with mouth opening/closing and other effects (such as mouth translation) have been removed.

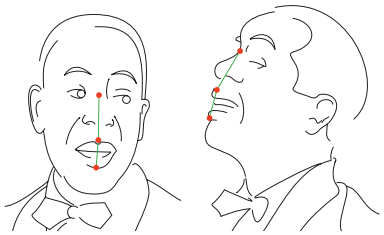

Figure 5: Labelling lip separation in such a way as to make it invariant to affine motions of the head.

When the animator first attempted to label the extent to which the piano-player's lips are separated, they labelled the distance between the lips in pixels. This is inappropriate since, as he turns to the right the distance between his lips (in pixels) decreases due to perspective effects, even if no lip movement has occurred. What is required is a way in which to label the images so that a fully open mouth gives the same numeric value, regardless of any scaling or rotation of the head in the scene. The solution adopted here is to label the lip separation as a *ratio* of the height of the face, as shown in Figure 5.

The third issue is that the feature to be extracted is in fact very small (only a few pixels in length) and difficult to accurately label by hand. This introduces a slight but noticeable 'jitter' into the resulting output. We take advantage of the temporal nature of the data by allowing the user to apply an extended Kalman filter to smooth the results.

Re-training the system using these three techniques now gives excellent lip-synching results and some stills extracted from the resulting cartoon are shown in Figure 6.

3.3 Horse Gait

Animators find it difficult to model the complex motion of quadrupeds, such as horses. Such motions are made more complex by the fact that the motions themselves completely change when the animal increases speed. Simply speeding up a walking horse does not make a convincing running horse. The image sequences used in this work were of a horse travelling on a treadmill with different gaits. Three sequences were captured, each consisting of 136 images (about 5.5 seconds of video) representing walking, trotting and cantering gaits. The sequences were analysed individually and in combination with the aim of extracting information about the way the horse or parts of the horse were moving. Figure 7 shows three of the features extracted from each sequence.

In Figure 8 the results of tracking the angle of the horses back are shown. Upon testing, a standard deviation of 2.0 degrees was obtained.

Another interesting experiment was to try and classify the gait of the horse, given a random still image from the entire sequence. An accuracy of 100% is obtained for this task (using 100 training images), perhaps not surprisingly since the first principle component by itself is almost capable of doing this, as shown in Figure 9.

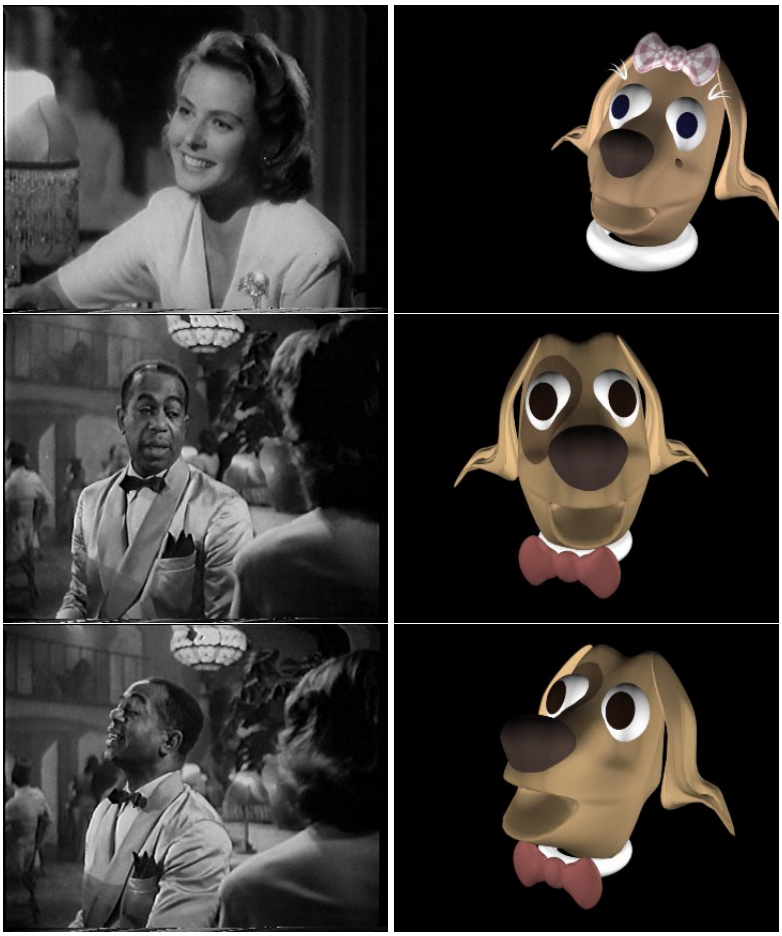

Figure 6: Cartoon of film clip. Both head models are animated using features extracted from the original sequence.

3.3.1 Ground-Truth

In a system such as ours it is often difficult to quantify the performance of the system, since a non-expert is in charge of creating the ground-truth (which is itself difficult and prone to error) and deciding when performance is "good enough". Generally, adding more and more training examples will improve the system, but clearly contravenes the aim of using as small an amount of user input as possible.

Labelling images by hand is a difficult and laborious task. In order to quantify the possible error in the labelling of this feature, the images were relabelled by the same person, using the same labelling tool, one month later. Figure 9 shows a plot of the two sets of hand labelled front leg angles for the 135 frame walking horse image sequence. The standard deviation of the error between these two labelling results was found to be 7.35 degrees. From Figure 9 it can be seen that errors are largest at the maximum and

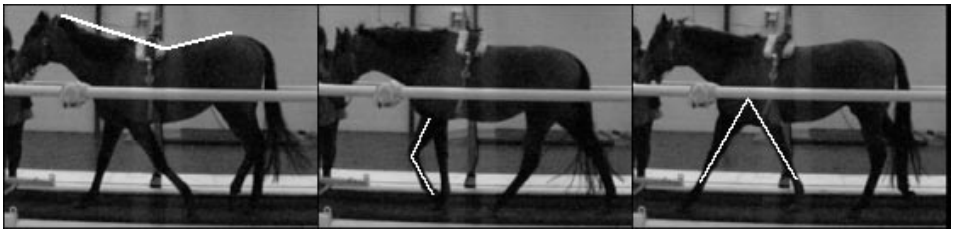

Figure 7: The features extracted from the horse images. The flexing of the back (Left), the angle of the left front leg (Middle) and the angle between the two front legs (Right).

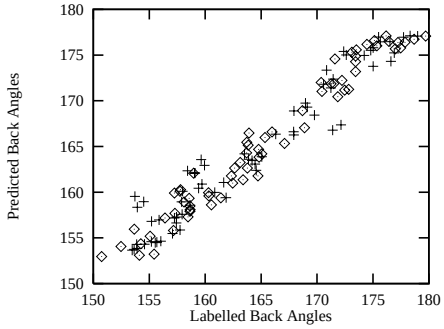

Figure 8: Labelled vs. predicted angles for the back of the horse. The figure shows both training set (diamonds) and test set (crosses).

minimum of the angle values. These cases correspond to the occurrence of occlusion at the minimum values where it becomes hard to tell which leg is which and the difficulty in determining where the pivotal point should be at the maximum values. When a neural network is trained using a subset of these images the standard deviation of the error between the hand labelled ground truth and the automatically labelled test images is 7.57 degrees. This shows that for some cases the hand labelling process is itself difficult and prone to almost as much error as our system after training on a subset of images.

4 Conclusions

We have described a system which is designed to assist animators in extracting high-level information from sets or sequences of images. The system is not meant to replace animators, but to be a tool to assist them in creating the first 'rough-cut' of a sequence quickly and easily. Using the system, short animations and cartoons have been created in a very short space of time. We have shown that the method of principal components analysis followed by a neural network learning phase is capable of motion tracking and also making high-level decisions such as predicting the angle of a birds wing, the type of gait of a horse and the ratio of two lengths to give mouth-tracking information. Tracking performance, even in the presence of occlusion, is impressive with the system being capable of tracking the angles between horses legs which frequently obscure each other. We do not claim that our system is better than other vision systems written with one particular, specialised

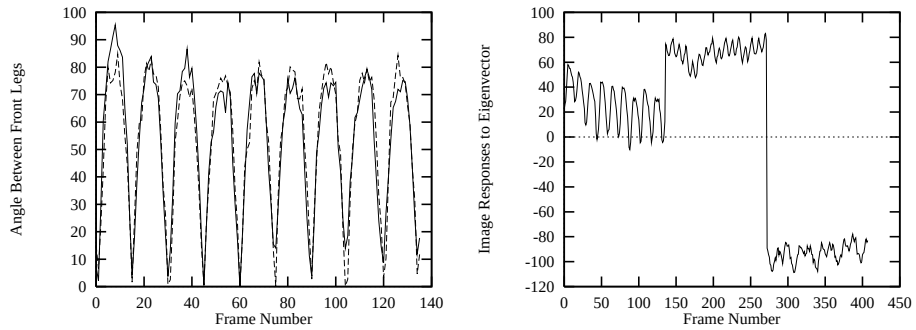

Figure 9: (Left) Two separate hand labellings for the angle between the front legs of the walking horse sequence. (Right) Image response to the first eigenvector of the entire horse sequence.

application in mind. We do, however, believe that we have demonstrated a system that is powerful, flexible and, above all, easy for non-specialists to use. Future work will look at utilising potentially more powerful non-linear basis functions [5] in place of principal components analysis.

References

[1] R. Brunelli and T. Poggio. Face Recognition: Features versus Templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[2] Richard E. Chandler. A Tracking Algorithm for Implicitly Defined Curves. *IEEE Computer Graphics and Applications*, 8(2):83–89, March 1988.

[3] D.P. Gibson. *The Application of Computer Vision to Very Low Bit-Rate Communications.* PhD thesis, University of Bristol., 1999.

[4] Michael Hoch and Peter C. Litwinowicz. A Semi-Automatic System for Edge Tracking with Snakes. *The Visual Computer*, 12(2):75–83, 1996. ISSN 0178-2789.

[5] J. Karhunen and J. Joutsensalo. Representation and Separation of Signals using Nonlinear PCA type Learning. *Neural Networks*, 7(1):113–127, 1994.

[6] B. Moghaddam and A. Pentland. Probabilistic Visual Learning for Object Detection. In *ICCV*, pages 786–793, 1995.

[7] A. Pentland and B. Horowitz. Recovery of Non-Rigid Motion and Structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):730–742, 1991.

[8] R.W. Picard and T. Kabir. Finding Similar Patterns in Large Image Databases. In *IEEE ICASSP*, volume V, pages 161–164, Minneapolis, MN, USA, April 1993.

[9] D.E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin. *Bakpropogation: The Basic Theory*, pages 1–34. 1995.

[10] Rung-Huei Liang and Ming Ouhyoung. A Real-time Continuous Alphabetic Sign Language to Speech Conversion VR System". *Computer Graphics Forum*, 14(3):67–76, August 1995.