Practical Image Processing and Computer Vision, Chapter on Human Activity Recognition in Video

Neil Robertson

Contents

1

Human activity recognition in video

1.1 Introduction

In contrast to many computer vision applications, such as HCI where the person is interacting with a computer via gestures, in surveillance or sports footage the zoom level often results in the imaged person being low/medium resolution e.g. 150 pixels high. This presents a unique challenge to any computer vision system due to the lack of fine detail with which to operate in order to interpret quite sophisticated behaviour. Due to the increasing interest in the automatic interpretation of human activity in video this challenge has recently begun to be addressed. The scientific state-of-the-art in video interpretation has started to move beyond the analysis of simple trajectories of tracked people. This chapter introduces computer vision techniques to help researchers progress towards the construction of a system for articulating higher-level descriptions of human activity in video. In this chapter a number of practical techniques are used including: mean-shift applied to video data, optic flow, skin detection, projective geometry and data fusion.

1.1.1 Chapter overview

The chapter opens by describing an efficient method for tracking objects in video which can be implemented to operate at frame-rate on modest hardware and also made robust to occlusion. Section 1.3 then describes a method for interpreting the type of activity that a person may be undertaking at any one instant, for example: walking, running etc. This is based on the computation of optic flow vectors between successive person-centred images of the target derived from the video tracker. Recognising that the perceived focus-of-attention of an individual is a significant clue to interpreting and predicting the actions of that person, section 1.4 describes a method for estimating the gazing-direction of the person from low-resolution face images. Illustrative examples from the surveillance domain are given throughout.

Practical Image Processing and Computer Vision **N.M.Robertson/M.Varga (ed.)**

1.2 Colour-based tracking in video

1.2.1 Introduction

(a) (b) (c) (d)

Figure 1.1 Automatic initiation of targets using background subtraction is possible when the camera is static.

By tracking an object repeated measurement of the location of a moving target throughout the frames of a video is achieved. Tracking is often a challenging task since the target may change in shape or appearance as the target orientation varies in relation to the camera. Additionally, there may be some small unwanted per-frame camera motion e.g. camera-shake caused by wind, for example. There may also be intentional motion due to smooth panning by an experienced operator to centre a moving target. Using colour alone to define the target provides invariance to shape changes so long as the appearance of the true target remains sufficiently different from background clutter.

When the images are acquired from a static camera, the target of interest can be initiated using background subtraction. Figure 1.1(a-d) illustrate the sequence of processing steps required to achieve this:

1. Obtain a reference "background" frame with no moving foreground objects (Figure 1.1(a)),
2. Subtract the foreground pixels (Figure 1.1(b)) directly to obtain the raw difference image (not shown),
3. Threshold the difference image and compute the connected components (Figure 1.1(c)),
4. Compute the blob orientation and location (Figure 1.1(d)),
5. Initiate the target model i.e. extract the histogram from the bounding region (Figure 1.1(d)).

1.2.2 The mean-shift algorithm

The target model is thus defined as a set of pixels extracted from one image. When using the mean-shift algorithm the target is represented by a histogram only. The *distribution* of colour pixels is significant, but the original *spatial arrangement* of the target pixels to another is not. The mean-shift algorithm uses the Battacharyya coefficient as the similarity measure between

two distributions which are discretised into u bins: $p(y)$ at the current image window centred at y and q, the target model histogram. This is given by:

$$\rho(p,q) = \sum_u \sqrt{p_u q_u} \tag{1.1}$$

which is maximised using an efficient iterative algorithm introduced by Comaniciu in (Comaniciu 1999). Each pixel, x, in a window (centred on the current target location y_0) is assigned a weight:

$$w_x = \sum_u \delta[I(x) - n]\sqrt{q_u / p_u(y_0)} \tag{1.2}$$

The new estimate of the target position is computed as:

$$y_1 = \frac{\sum_x x w_x k(x,y)}{\sum_x w_x k(x,y)} \tag{1.3}$$

where k is a kernel which weights pixels close to the centre of the current window higher than those at the edge. A Gaussian kernel is therefore appropriate. The iteration stops when $|y_1 - y_0| < \epsilon$, where ϵ is a predefined threshold, typically 1 pixel.

1.2.3 Mean-shift in scale-space

A search in scale-space is interleaved between each step of the gradient-descent in position (which is described above). In order to achieve this, a set of Gaussian kernels are defined with:

$$\{\sigma_s = \sigma_0 * b^s, -n \leq s \leq n\} \tag{1.4}$$

where $b > 1$ is the base of the logarithmic scale and n defines range of the search in scale around the current scale σ_0. Collins suggest choosing $b = 1.1$ and $n = 2$ (Collins 2003). The effect of tracking in scale, as well as image space, can be significant as the tracker is less likely to be seduced by passing objects. This is illustrated in Figure 1.2where, when a person is tracked without adjusting the scale parameter, the passing vehicle attracts the tracker due to the similarity in colour. By tracking in scale-space as well as position the tracking is more robust because the true position of the tracked object is located more accurately. It is desirable that as much of the background be eliminated from the target as possible when the target-centred image will be used for further processing, such as for recognising actions - the subject of section 1.3.

1.2.4 A strategy for dealing with occlusion

While the basic mean-shift algorithm offers a degree of robustness to occlusion, as shown in Figure 1.3, it will, as with most simple appearance-based tracking algorithms, fail where the target is completely occluded. In order to provide robustness to occlusion Bibby and Reid (Bibby and Reid 2005) proposed a simple improvement to the standard mean-shift approach: when the Battacharyya coefficient drops below a certain value, the search window is expanded and the Battacharyya coefficient for a grid of windows around the current location is computed. Provided the target has not disappeared altogether or moved outwith this wider search region, the location can often be recovered. An example of the utility of

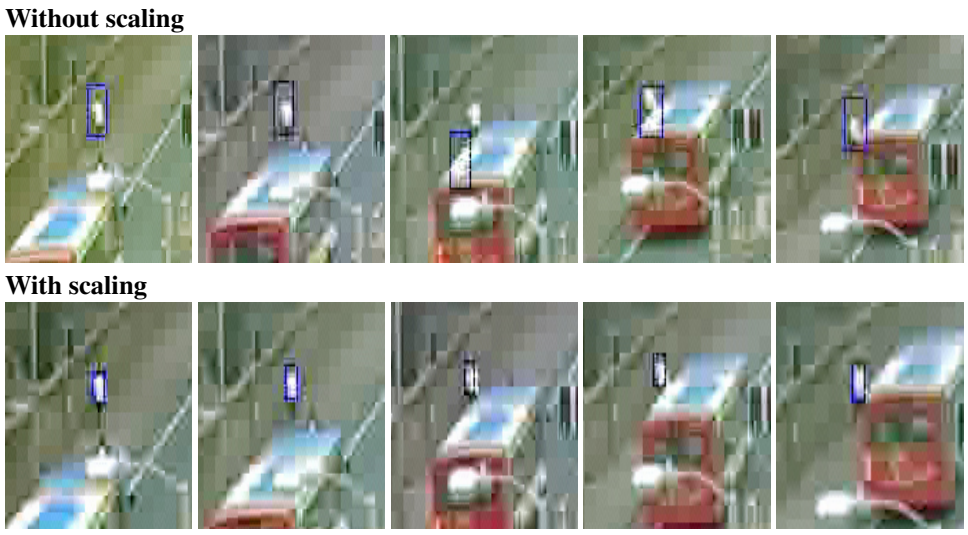


Figure 1.2 Tracking in scale as well as position can prevent the track becoming seduced by passing objects.

this method in a surveillance context is shown in Figure 1.4 where the tree in the scene has the potential to completely occlude the target which in this case is a person. The standard mean-shift algorithm fails when the target is completely occluded because the search window does not extend to the point where the target reappears. The position update has no better estimate than the current location since the true model has disappeared and, in general, all of the local background represents an equally dissimilar colour histogram compared to the target histogram. By expanding the search window when the histogram similarity measure - the value of the Battacharyya coefficient - falls below a specified threshold it is possible to recover the true target location.

1.3 Action recognition

Using the mean-shift tracking algorithm, the following information for each target in every frame can be extracted: position, velocity and the bounding-box of the target. In addition to the target's place and speed it is very much of interest to classify the *action* of the person being tracked e.g. *walking* or *running*.

One of the most promising methods for doing this is based on identifying the dominant components of motion between frames for a certain activity class. For example, one expects that when comparing *walking* and *running* the legs or arms have a sufficiently different motion between such that it is possible to disambiguate the two automatically. Efros *et al.* (Efros 2003) developed a simple, yet effective, local motion descriptor based on coarse optic flow which is extracted from a stabilised target window. This pre-processing step is achieved by segmenting the target from the background and centering the "mass" of the target in the window using the centroid of the blob computed using a connected-components

Figure 1.3 The mean-shift target tracking algorithm has a degree of robustness to partial occlusion provided the targets are suitably distinct in appearance.

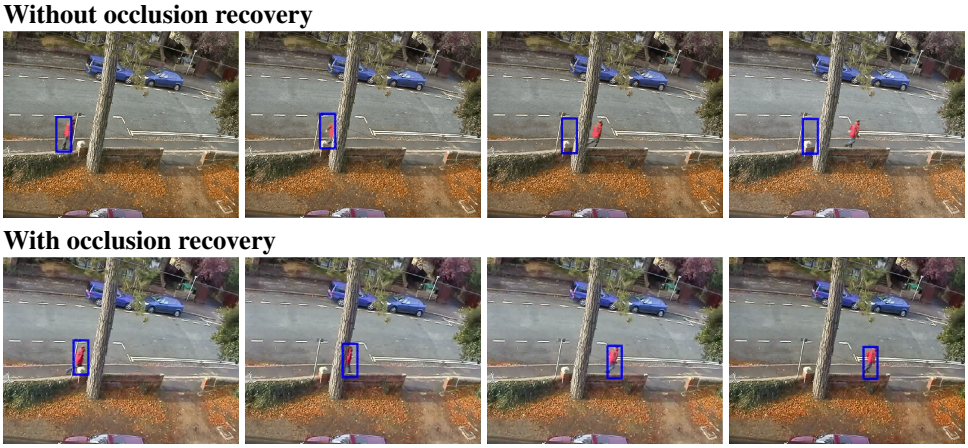


Figure 1.4 The addition of an occlusion-reasoning step enables tracking recovery when the target becomes completely obscured.

Figure 1.5 Fixating on a target using a colour-based tracker. The extracted target image is shown in the expanded images along various points of the target centroid trajectory, showing tracking successfully in scale and image-space.

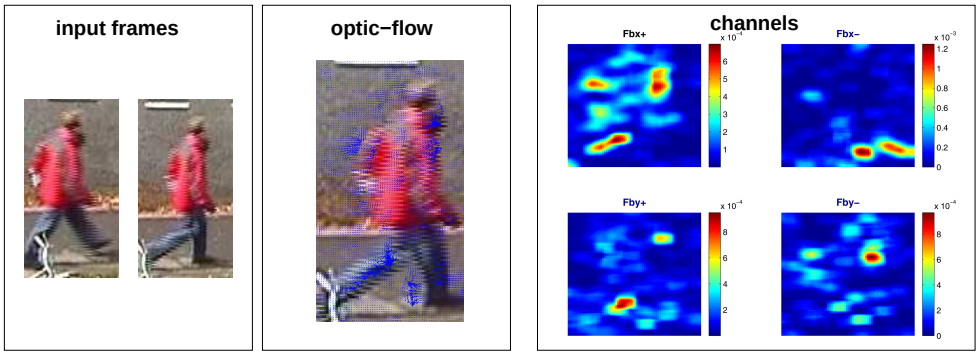
algorithm.

This local motion descriptor is then compared to the entries in a database of previously-seen motion descriptors that have been hand-labelled with the corresponding actions. The nearest-neighbour match therefore provides an action label for the current data. Note that more efficient search techniques may be employed but are not described here (for more detail see Robertson and Reid (2006)).

In order to obtain this action descriptor, the optic flow between consecutive frames of a sequence is computed. Optic flow is ideal for this purpose because it is photometrically invariant and invariant to clothing or appearance (Lucas 1981). Invariance is essential because a general description of the motion of a person is required to match the action between different people even though they may vary in size and appearance.

1.3.1 Optic flow

Optic flow is a measure of image-velocity. In estimating optic flow the aim is to compute an approximation to the 2-D motion-field which is a projection of the 3-D velocities of surface points onto the image plane (Horn 1986; Verri and Poggio 1987). There exist a number of methods for estimating the optic flow field. Barron *et al.* have reported on a comprehensive study of the most common methods in each of these categories (Barron 1994). While they do not conclude that one method is consistently superior than all others, it is apparent from the experiments performed that the Lucas and Kanade technique (Lucas 1981) is among the best for the quantitive experiments performed by Barron *et al.* (Barron 1994). The average error across synthetic and real sequences was reported as 1.06^o. The Lucas-Kanade algorithm gives the best performance i.e. the angular error is proved to be the smallest of all common



(a) (b) (c)

Figure 1.6 The action-recognition descriptor is computed from optic flow vectors between successive images of a tracked person.

optic flow measures, according to Barron *et al.*

The results of the Lucas-Kanade method applied to images of a person walking are shown in Figure 1.6. As can be seen from this example, optic flow reveals how pixel information is translated in an image between successive frames. For a given input pair of images, which are shown in Figure 1.6(a), the flow vectors are computed. These are superimposed on one image and shown in Figure 1.6(b). The Gaussian blurred optic flow in the x and y direction is further split into the four (blurred) non-negative channels which are shown in Figure 1.6(c). When combined these blurry motion channels comprise a descriptor of instantaneous action defined which can be used as the basis for human action-recognition, which is described in more detail in the next section.

1.3.2 From flow-vectors to action descriptors

The optic flow vector-field $\mathbf{F}$ is split into two scalar fields which are the horizontal and vertical components of the optic flow field, F_x and F_y. These are then half-wave rectified into positive channels F_x^-, F_x^+, F_y^- and F_y^+ such that:

$$F_x = F_x^+ - F_x^- \tag{1.5}$$

$$F_y = F_y^+ - F_y^- \tag{1.6}$$

Each of the channels is blurred with a Gaussian kernel and normalised, producing the four motion descriptors for every frame of the sequence $\hat{F}b_x^+$, $\hat{F}b_x^-$, $\hat{F}b_y^+$ and $\hat{F}b_y^-$.

A version of normalised cross-correlation is further employed such that, if the four motion-channels for frame i of a sequence A are defined to be a_1^i, a_2^i, a_3^i and a_4^i (similarly for frame j of the sequence B), then the similarity between motion descriptors centred at frames i and j is given by:

$$S(i,j) = \sum_{t \in T} \sum_{c=1}^{4} \sum_{x,y \in I} a_c^{i+t}(x,y) b_c^{j+t}(x,y) \tag{1.7}$$

and, when the matrix A_1 is defined as the concatenation of all a_1 vectors (similarly for the other channels, and for sequence B), the frame-to-frame similarity matrix between the two sequences is:

$$S = A_1^T B_1 + A_2^T B_2 + A_3^T B_3 + A_4^T B_4 \tag{1.8}$$

T is defined in the work of Efros *et al.* as "the temporal extent of the descriptor". Although equation 1.7 implies that, by varying T, temporal context can be achieved, in practice, T is defined when the descriptor is computed, initially.

For frame-to-frame optic flow, therefore $T = 1$, or at most $T = 2$. It is not explained that, unless encoded in the descriptor itself, that Efros *et al.* intended this term to allow for temporal context in the descriptor, as this is not discussed in the work of (Efros 2003).

Further, Efros *et al.* recognise that if the sequences A and B are similar but occur at different rates the similarity matrix will have strong responses along the off-diagonal elements and so S is convolved with a kernel which is a weighted-sum of near-diagonal lines:

$$K(i,j) = \sum_{r \in R} w(r) \chi(i, rj) \tag{1.9}$$

where R is a range of rates.

1.3.3 Observations and results

It should be borne in mind that this action-recognition method works well when a newly-observed sequence for which one wishes to find a best match is represented in the example set which has been previously compiled. If the database of exemplars is small, then there is an increased risk of a mismatch. For every example sequence in the exemplar set (which may be regarded as a "database"), the best match can be found at any time step by using the similarity matrices of equation 1.8 as a lookup table.

For illustration, these matrices are shown in Figure 1.7 for a new sequence compared to four exemplar sequences. In this example, the database comprises 4 sequences and the frame-to-frame similarity matrices are shown for each of these models given the new input sequence (shown in middle row). Note that for the best-matching *sequence* there is evidence of periodic structure in the similarity matrix (*left* "walk-LR"). The best matching frame in the database at each new input frame is chosen from the similarity matrices. The input frames are at the top and, directly below, the best matching frame is shown. The different backgrounds clearly indicate these are different frames from separate sequences. For completeness, in the *third row* it is shown that matching is effective despite the fact that the appearance of each person is quite different.

A final illustration drawn from a second scene which has a considerably richer set of actions is shown in figure 1.8. The example set is comprised of 27 different types of *spatio-temporal* activity with a range of person-centred actions from walking in a variety of directions relative to the camera to running and standing still, loitering etc. A new example of the action "walking" is matched into the exemplar database by taking the ML match from

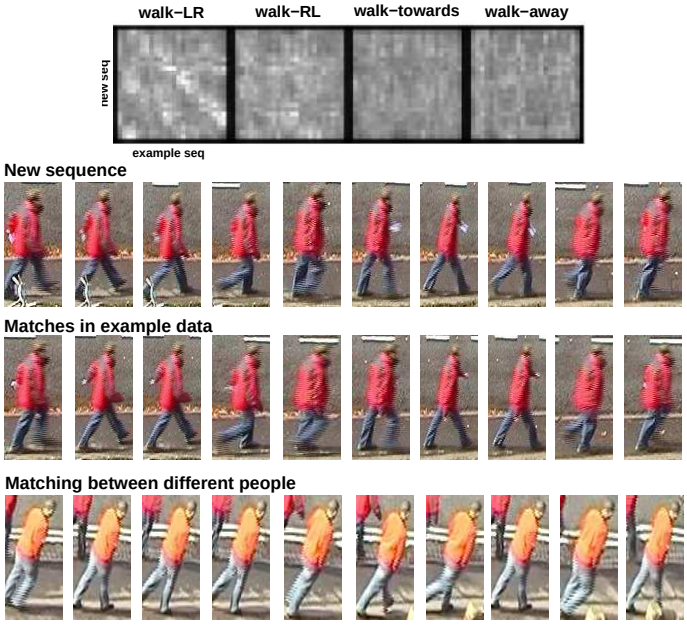


Figure 1.7 Similarity matrices which compare each motion descriptor to each example in the database, can be used to pick the most similar match from those actions already observed.

Figure 1.8 Action recogntion using data gathered from a CCTV camera.

the previously-seen examples at each frame. The input is on the top row of the sequence of person-centred frames, with the nearest-matching exemplar frame directly beneath.

1.4 Gaze-direction estimation

In applications where human activity is under observation, be that CCTV surveillance or sports footage, for example, knowledge about where a person is looking (i.e. their gaze) provides observers with important clues which enable accurate explanation of the scene activity. It is possible, for example, for a human readily to distinguish between two people walking side-by-side but who are not "together" and those who are acting as a pair. Such a distinction is possible when there is regular eye-contact or head-turning in the direction of the other person. In soccer or rugby, for example, head position is often a guide to where the ball will be passed next i.e. it is an indicator of *intention*. In all but the most highly-skilled teams, where awareness of a team-mate's position appears to be intuitive, a player at least glances in the direction of the intended pass. This is essential for higher-level processing and video understanding.

This section describes a method for automatically inferring gaze direction in images where any one person represents only a small proportion of the frame. Typically in surveillance images, the head ranges from 20 to 40 pixels high.

1.4.1 Previous work in gaze-direction estimation

Determining the instantaneous focus of a person's attention in surveillance images is a challenging problem that has received little attention despite the necessary components existing for some time in the image processing literature. This problem was first addressed by the author (Robertson *et al.* 2005; Robertson and Reid 2006). Although there has been some interesting related work reported in the literature.

Everingham and Zisserman (Everingham and Zisserman 2005) developed a technique for overlaying 3-D head models on faces, with a resolution in the range 15 to 200 pixels high as a means to identifying people in broadcast video sequences. This could have potentially been extended to determine where the person is looking but the crucial drawback with Everingham and Zisserman's work in relation to surveillance video is the fact that they search for faces of a *specific* character whose appearance is known *a priori* and for whom a 3-D face model has been constructed in advance. This would clearly be impossible in a surveillance application where nothing is known about the appearance of the person under observation before they appear in the video.

Closely related in technical approach to the work of this chapter is that of Efros *et al.* (Efros 2003) for recognition of human action at a distance which was described in section 1.3. Head pose is not discussed by Efros (Efros 2003) but the use of a descriptor invariant to lighting and clothing is of direct relevance to head pose estimation and has inspired aspects the algorithm described in this section.

Dee and Hogg (Dee and Hogg 2004) developed a system for detecting unusual activity which involves inferring which regions of the scene are visible to an agent within the scene. A Markov Chain with penalties associated with non-hidden state transitions is used to return a score for observed trajectories. The state transition penalties essentially encode how directly a person made his/her way towards predefined goals, typically scene exits. In their work, gaze inference is vital, but gaze is inferred from trajectory information alone which can lead to significant interactions being overlooked, as shown later in this chapter, because the assumption that the head is always aligned with body-direction is not robust.

In contrast, there has been considerable effort to extract gaze direction from relatively high-resolution faces, motivated by the drive toward ever better Human/Computer Interfaces (Gee and Cipolla 1994; Matsumoto 2000; Perez 2003).

1.4.2 Skin detection

The lowest level of this approach is based on skin detection. Because of the significant interest in detecting and tracking people in images and video, skin detection has naturally received much attention in the image processing and computer vision community (Chai and Ngan 1998; Hidai 2000; Jebara 1997). However skin detection alone is error-prone when the skin region is very small as a proportion of the image. That said, contextual cues such as direction can help to disambiguate gaze using even a very coarse head-pose estimation. By combining this information in a principled i.e. probabilistic, Bayesian fashion, gaze estimation at a distance becomes a distinct possibility as discussed later in the chapter (see section 1.4.4).

The aim is to determine which pixels in an image correspond to skin and non-skin. Perhaps the most straightforward method is to construct a look-up table by deciding in advance in which regions of a given colour-space skin colour is found. This method was used by Chai and Ngan (Chai and Ngan 1998). This technique is unreliable in very low resolution images, however. Hidai et al. (Hidai 2000) defined an ideal skin colour using an average of exemplar face images from which they defined skin and non-skin pixels via non-parametric matching. Parameterised techniques usually involve multi-variate Gaussians, the parameters of which are learned using the Expectation-Maximisation algorithm (Jebara 1997).

Although people differ in colour and length of hair and some people may be wearing hats, beards etc. it is reasonable to assume that the amount of skin that can be seen, the position of the skin pixels within the frame and the proportion of skin to non-skin pixels is a relatively invariant, if coarse, cue for a person's gaze direction in a static image. This descriptor is obtained in a robust and automatic fashion as follows. First, a mean-shift tracker (Comaniciu 1999) is automatically initialised on the head by using naive background subtraction to locate people and subsequently modelling the person as distinct "blocks", the head and torso. Second, the head is centred within the tracker window at each time step which stabilises the descriptor ensuring consistent position within the frame for similar descriptors. That is, the head images are scaled to the same size and, since the mean-shift tracker tracks in scale-space a stable, invariant, descriptor is obtained. Third, there is no specific region of colour-space which represents skin across all sequences and therefore it is necessary to define a skin histogram for each scenario by hand-selecting a region of one frame in the current sequence to compute a normalised skin-colour histogram in RGB-space. (It has been demonstrated that there is no difference in the performance of skin detectors based on colour-regions when RGB or YCbCr, HSV etc. colour-spaces are used (Phung 2003).) The weights (skin/non-skin probabilities) are then computed for every pixel in the stabilised head images which the tracker automatically produces to indicate how likely it is that it was drawn from this predefined skin histogram. This will be recognised as a similar approximation to the Battacharyya coefficient as implemented in the mean-shift algorithm above. Using the knowledge of the background the foreground is segmented out of the tracked images. Every pixel in the segmented head image is drawn from a specific RGB bin and so is assigned the relevant weight which can be interpreted as a probability that the pixel is drawn from the skin model histograms.

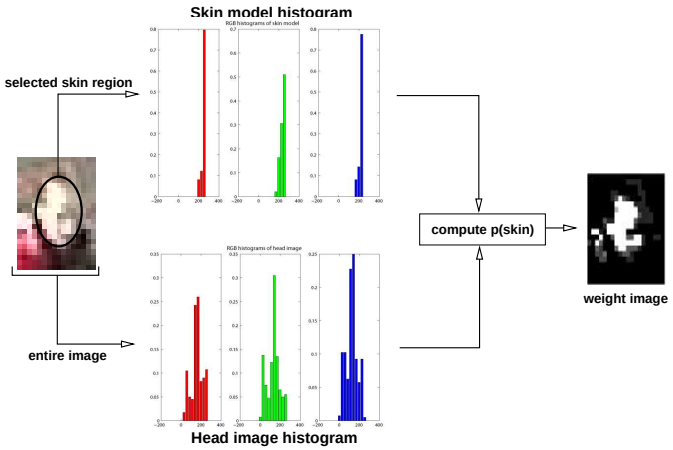


Figure 1.9 Computing the probability of skin pixels using a pre-defined reference histogram.

Some mean-shift implementations suggest a histogram discretised into 20 bins for each dimension of colour space. So if a 3-D histogram is computed with axes along the R, G and B dimensions of the colour-space then the histogram is an 8000-element volume. The actual skin-colour occupies a very small region of this volume. A significant amount of computational effort is therefore expended computing this large histogram for each step of the tracker since the weights are computed at each frame.

It is therefore expedient to split the RGB space into three independent histograms, compute the likelihood that each pixels R, G and B value was drawn from that histogram and multiply together to obtain a likelihood that each pixel was drawn from the overall (RGB) skin histogram. For every bin i (typically there are 10 bins) in the predefined, hand-selected skin-colour histograms q_R, q_G and q_B the histograms of the tracked image p_R, p_G and p_B a weight, w_i, is computed:

$$w_i = \sqrt{\frac{q_{R,i}}{p_{R,i}}} \cdot \sqrt{\frac{q_{G,i}}{p_{G,i}}} \cdot \sqrt{\frac{q_{B,i}}{p_{B,i}}} \tag{1.10}$$

Every foreground pixel in the tracked frame falls into one of the bins according to its RGB value and the normalised weight associated with that pixel is assigned to compute the overall weight image, as shown in Figure 1.9. The non-skin pixels are assigned a weight that the pixel is *not* drawn from the skin histogram. This non-skin descriptor is necessary because it encodes the "proportion" of the head which is skin, which is essential as people vary in size and scale. Each descriptor is scaled to a standard 20×20 pixel window to achieve robust comparison when the head sizes vary.

Figure 1.10 shows the images which result from the mean-shift image patch tracker (*col. 1*). Background subtraction is applied (*col. 2*) and the descriptor is normalised by centering the head in the window and resizing. The weight image which represents the probability

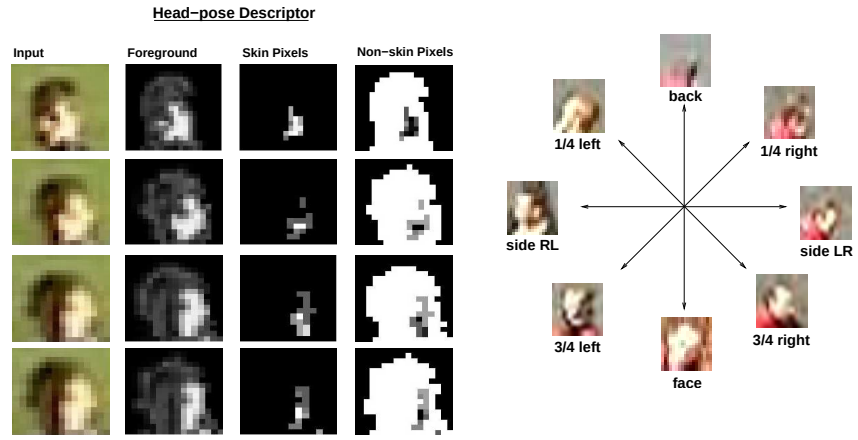


Figure 1.10 (Left) The head-pose descriptor is defined using the skin and non-skin which defines the proportion of skin to the size of the head. (Right) Head-pose is discretised into 8 distinct poses.

that each pixel in the head is skin is computed (*col. 3*). From this, non-skin probability can be easily computed (*col. 4*). Non-skin is significant as it captures proportion without the need for scaling. The concatenation of skin and non-skin weight vectors is the feature vector which is used to determine eight distinct head poses which are shown and labelled on the in Figure 1.10.

A set of training exemplars are identified and labelled. Algorithm 1 describes the process for the extraction of training data. Varying lighting conditions should be accounted for by representing the same head-pose under light from different directions in the training set. The same points on the "compass" are used as the discretisation of direction i.e. N, NE, E, etc.

Algorithm 1 To obtain head-pose training data

1: Track head in a video sequence
2: Centre head within tracker window at each frame
3: Define skin histogram for sequence (by hand, if necessary)
4: Segment the foreground in every image
5: For every pixel belonging to the foreground compute p(skin) and p(non-skin)

Once more, the nearest-neighbour match in the exemplar dataset can be found for any given input. Examples of successful matching using this descriptor are shown in figure 1.11 with the estimated gaze-angle superimposed on the images. Note that the same set of training examples is used across a variety of test datasets.

Figure 1.11 Detecting head pose in different scenes using a standard set of training examples.

1.4.3 Rectification to the ground-plane

Gaze inference is only of use if it can be estimated from the gazing direction what it is that a person can see or, even better, what he/she is *looking at*. The human visual system has a field-of-view of 105^o (Prothero and Hoffman 1995). Picking an arbitrary visual range therefore allows the 2-D visual field to be drawn on the images. Note that there is no occlusion reasoning about the field-of-view so this is an idealised indication of what can be seen. What can truly be seen by the person is in the world and not the image plane. Therefore it is important that some effort is invested to correct for various perspective effects, if the gaze area in pixels is to be used for further processing/reasoning.

Computing a planar homography

The homography computation allows the image to be "ortho-rectified". That is, to warp the original image in such a way that the view is as though the image was capture by a camera whose image plane is parallel to the ground-plane. This is done by computing the planar projective transformation which is a linear transformation on homogeneous 3-vectors represented by a non-singular 3×3 matrix. For details see (Hartley and Zisserman 2003).

The easiest way to compute the projective transform required to rectify an image is to select, in the image, a set of points corresponding to a planar section of the world. Image coordinates and world coordinates are selected as shown in Figure 1.12. The control points shown in Figure 1.12 are on a plane which has been warped by perspective effects in the imaging process. By computing the inverse transform it is possible to undo the effect of perspective.

It is important to note that the rectification achieved in this way does not require any knowledge of the camera's parameters or the pose of the plane. The effect of this on a full

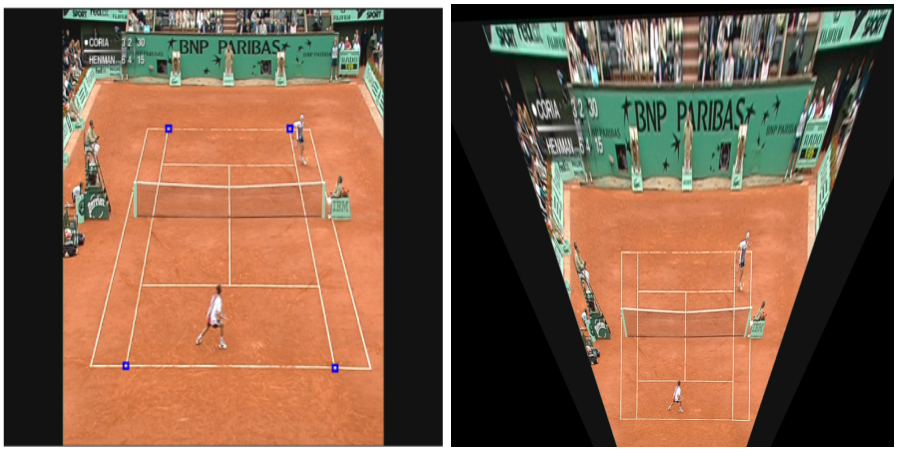


Figure 1.12 The rectification of an image to the ground plane is achieved by computing the projective transform between point correspondences.

frame in shown in Figure 1.12. However one does not want to compute the entire frame's projection, just the gaze so that one can determine what can really be seen in the world by the person. This is demonstrated in Figure 1.13. Figure 1.13(a) illustrates the gaze with no projection of the onto the ground-plane and no compensation of the gaze angle (relative to the camera-centred frame). In Figure 1.13(b) gaze is projected onto ground-plane but perspective alterations in the assigned angle are not computed. In Figure 1.13(c) the gaze angle is computed using the projection from camera-frame to world-frame to create the final estimate of what the person can see.

Correcting gaze angles under perspective imaging

In order to display where the person is looking in the images angles are assigned to the discretised head-poses shown in Figure 1.10 according to the "compass" e.g. $N : 0^o$ etc. However, when the field-of-view is superimposed on the image (and, more importantly, when visibility of other objects in the scene is determined using this field-of-view) it is important to correct for the fact that the camera is not fronto-parallel to the scene as for the acquisition of training data. The assigned angles must then be corrected for the projection of the camera at each time step depending on the location of the person on the ground-plane in the image.

In order to choose the correct frame of reference there is no need to perform full camera calibration but rather to compute the projective transform ($\mathbf{H}$: image$\rightarrow$ground-plane) by hand-selecting 4 points in the image as described above and shown in Figure 1.14. The vertical vanishing point, ($\mathbf{v}$), is computed from the manual selection in the image of 2 lines which are known to be normal to the ground plane and parallel in the world. (See (Hartley and Zisserman 2003) §8.6 for details on how this relates to the "footprint" of the camera on the reference ground-plane). The angle θ between the projection of the optic-rays through the camera centre, $\mathbf{Hv}$, and the image centre, $\mathbf{Hc}$, and the point at the feet of the tracked person,

(a) (b) (c)

Figure 1.13 Progression of improvements for visualising the gaze estimate.

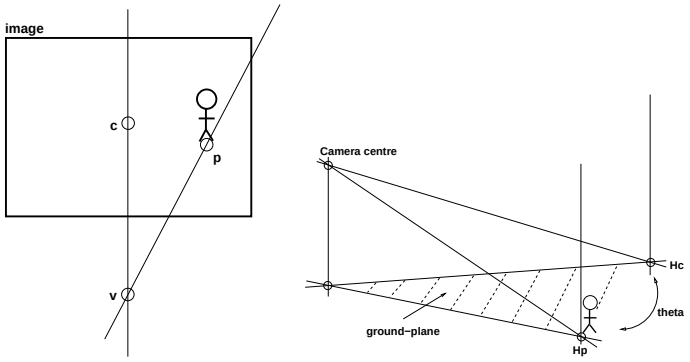


Figure 1.14 When assigning angles to the matched discretised head-poses one must compensate for the camera projection since the angles assigned to head-poses do not in general correspond to vertical in the image plane.

$\mathbf{Hp}$, is the angle which adjusts vertical in the image to "North" in the ground plane reference frame i.e.

$$\theta = \cos^{-1}[(\mathbf{Hc} \times \mathbf{Hv}).(\mathbf{Hv} \times \mathbf{Hp})] \tag{1.11}$$

1.4.4 Bayesian fusion of head-pose and direction

The naive assumption that direction of motion information is a good guide to what a person can see has been used to generate the estimated focus-of-attention in the first row of Figure 1.15. However, it is clear the crucial interaction between the two people is missed. To address this issue one may compute the joint posterior distribution over direction of motion *and* head pose. The priors on these are initially uniform for direction of motion, reflecting the fact that for these purposes there is no preference for any particular direction in the scene, and for head pose a centred, weighted function that models a strong preference for looking forwards rather than sideways. The prior on gaze is defined using a table which lists expected (i.e. physically possible) gazes and unexpected (i.e. non-physical) gazes.

Define g as the measurement of head-pose, d is the measurement of body motion direction. G is the true gaze direction and B is the true body direction, with all quantities referred to the ground centre. The joint probability of true body pose and true gaze is then given by:

$$P(B, G|d, g) \propto P(d, g|B, G)P(B, G) \tag{1.12}$$

Now given that the measurement of direction d is independent of both true and measured gaze G, g once true body B pose is known,

$$P(d|B, G, g) = P(d|B) \tag{1.13}$$

Similarly the measurement of gaze g is independent of true body pose B given true gaze G, i.e.

$$P(g|B, G) = p(g|G) \tag{1.14}$$

Then:

$$P(B, G|d, g) \propto P(g|G)P(d|B)P(G|B)P(B) \tag{1.15}$$

It is assumed that the measurement errors in gaze and direction are unbiased and normally distributed around the respective true values

$$P(g|G) = \mathcal{N}(G, \sigma_G^2), P(d|B) = \mathcal{N}(B, \sigma_B^2) \tag{1.16}$$

(actually, since these are discrete variables a discrete approximation is used).

The joint prior, $P(B, G)$ is factored as above into

$$P(B, G) = P(G|B)P(B) \tag{1.17}$$

where the first term encodes the knowledge that people tend to look straight ahead (so the distribution $P(G|B)$ is peaked around B, while $P(B)$ is taken to be uniform, encoding the belief that all directions of body pose are equally likely, although this is easily changed: for example in tennis one player is expected to be predominantly facing the camera).

While for single frame estimation this formulation fuses the measurements with prior beliefs, when analysing video data one can further impose smoothness constraints to encode temporal coherence: the joint prior at time t is in this case taken to be

$$P(G_t, B_t|G_{t-1}, B_{t-1}) = P(G_t|B_t, B_{t-1}, G_{t-1})P(B_t|B_{t-1}) \tag{1.18}$$

where the assumption that the current direction is independent of previous gaze is used. Although it is recognised that this may in fact be a poor assumption in some cases since people may change their motion or pose in response to observing something interesting while gazing around. It is also assumed that the current gaze depends only on current pose and previous gaze. The former term, $P(G_t|B_t, B_{t-1}, G_{t-1})$, strikes a balance between between the belief that people tend to look where they are going, and temporal consistency of gaze via a mixture i.e.

$$G_t \sim \alpha\mathcal{N}(G_{t-1}, \sigma_G^2) + (1 - \alpha)\mathcal{N}(B_t, \sigma_B^2) \tag{1.19}$$

The joint distribution for all 64 possible gazes resulting from possible combinations of 8 head poses and 8 directions is now computed using this result. This posterior distribution allows the probabilistic estimates to be maintained without committing to a defined gaze which will be advantageous for further reasoning about overall scene behaviour. Immediately though it can be seen that gazes which are considered very unlikely given the prior knowledge of human biomechanics (since the head cannot turn beyond 90^o relative to the torso (Pang 2004)) can be rejected, in addition to the obvious benefit that the quality of lower-level match can be incorporated in a mathematically sound way. An illustrative example is shown in Figure 1.15. In this video sequence there is an interaction between two people in the frames. The fact that they look at each other is the prime indicator that they are "together". On the first row the gaze from body direction alone is estimated. On the second row gaze is estimated using head-pose alone, which gives an improved result, as far as detecting the interaction is concerned, but this is still prone to some errors. In the third row of Figure 1.15 it can be seen that fusing the head-pose and body-direction estimates gives a significantly improved result when it is the interaction that is required to be identified. That is, the "head angles" graph clearly shows two main head-turning events, the first short, the second longer. The angle-error is computed by comparing the estimated head-angles to hand-labelled ground-truth.

1.5 Conclusion

This chapter has described the scientific state-of-the-art for human activity recognition in surveillance video sequences. The emphasis has been on achieving canonical descriptions of "instantaneous" motion-type and focus-of-attention. A common theme in the technical approach is the development of a descriptor which can be readily computed from low/medium resolution images of people.

Further avenues for research based on the work of this chapter include:

- Use of contextual information, such as position and velocity, to articulate text "commentary" on activity. This could be particularly effective in generating automatic sports commentary, for example.

- For reasoning about human activity, the action/gaze recognition algorithms could be viewed as providing data at the sensors of the system which could then be used as input to a higher level video understanding component, perhaps based on Bayesian networks.

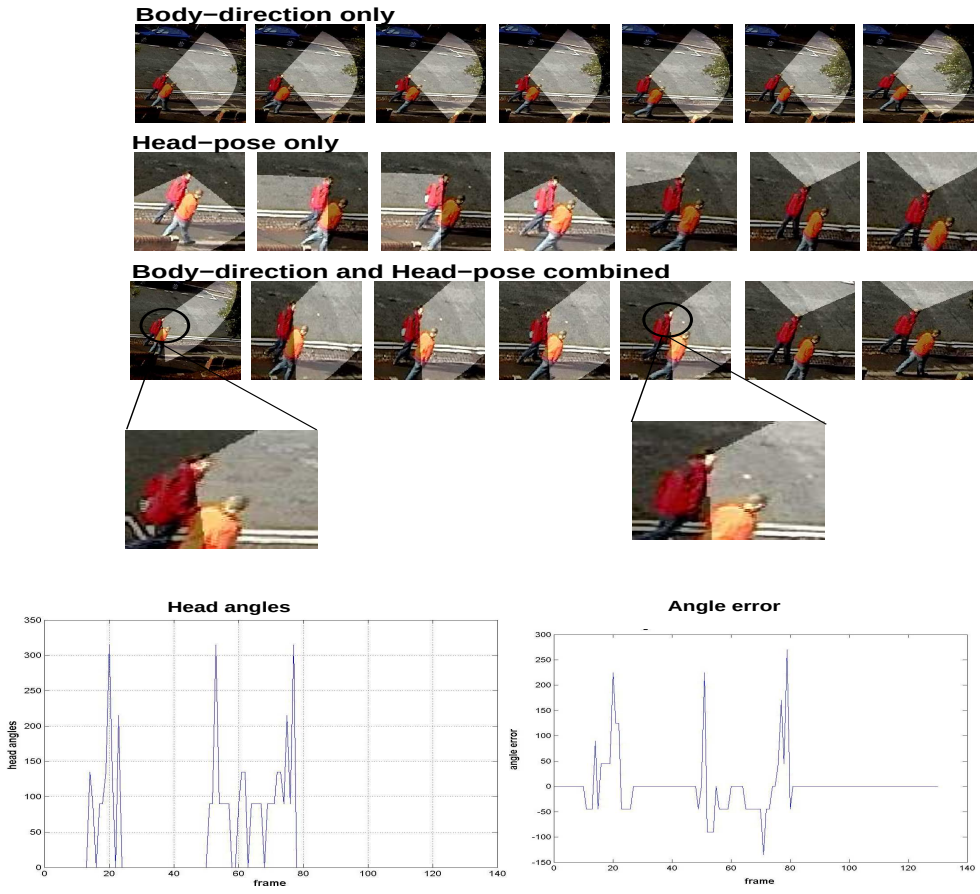


Figure 1.15 Fusion of body-direction with head-pose improves the gaze-estimate.

Bibliography

J.L. Barron, D.J. Fleet, S.S. Beauchemin *Performance of Optical Flow Techniques* International Journal of Computer Vision 12:1 pp43-77, 1994.

C. Bibby and I. Reid *Visual Tracking at Sea* International Conference on Robotics and Applications, Barcelona, 2005

D. Chai and K. N. Ngan *Locating facial region of a head-and-shoulders color image* Third IEEE International Conference on Automatic Face and Gesture Recognitions, Nara, Japan, pp. 124-129, April 1998.

R.T. Collins *Mean-shift Blob Tracking through Scale Space* IEEE Computer Vision and Pattern Recognition, Madison, WI, June 2003.

D. Comaniciu and P. Meer *Mean-shift Analysis and Applications* Proceedings of the International Conference on Computer Vision-Volume 2, p.1197, September 20-25, 1999.

H. Dee and D. Hogg *Detecting Inexplicable Behaviour* Proceedings of the British Machine Vision Conference, 2004.

A.A. Efros, A. Berg, G. Mori and J. Malik *Recognising Action at a Distance* Proceedings of the International Conference on Computer Vision, Nice, France, July 2003.

M. Everingham and A. Zisserman *Identifying individuals in video by combining generative and discriminative head models* Proceedings of the International Conference on Computer Vision, Beijing, China, October 17-20, 2005.

A.H. Gee and R. Cipolla. *Determining the gaze of faces in images.* Image and Vision Computing, 12(10):639-647, December 1994.

R. Hartley and A. Zisserman *Mulitple view geometry in computer vision* Cambridge University Press, 2nd Ed., 2003, ISBN 0521 54051 8.

K. Hidai et al. *Robust Face Detection against Brightness Fluctuation and Size Variation* International Conference on Intelligent Robots and Systems, vol. 2 pp. 1379-1384, Japan, October 2000.

B.K.P Horn *Robot Vision* MIT Press, Cambridge, MA, USA, 1986.

T.S. Jebara and A. Pentland *Parametrized Structure from Motion for 3-D Adaptive Feedback Tracking of Faces* Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pp. 144-150.

B.D. Lucas and T. Kanade *An Iterative Image Registration Technique with Application to Stereo Vision* DARPA Image Understanding Workshop, 1981.

Y. Matsumoto and A. Zelinsky *An Algorithm for Real-time Stereo Vision Implementation of Head Pose and Gaze Direction Measurement* Proceedings of IEEE Fourth International Conference on Face and Gesture Recognition, pp. 499-505, 2000.

D. Pang, M.D. and V. Li, M.D. *Atlantoaxial Rotatory Fixation: Part 1-Biomechanics OF Normal Rotation at the Atlantoaxial Joint in Children.* Neurosurgery. 55(3):614-626, September 2004.

A.Perez, M.L. Cordoba, A. Garcia, R. Mendez, M.L. Munoz, J.L. Pedraza, F. Sanchez *A Precise Eye-Gaze Detection and Tracking System* Proceedings of the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2003.

S.L. Phung, A. Bouzerdoum, and D. Chai *Skin segmentation using color and edge information* Proc. Int. Symposium on Signal Processing and its Applications, 1-4 July 2003, Paris, France.

J.D. Prothero and H.G. Hoffman *Widening the Field-of-View Increases the Sense of Presence in Immersive Virtual Environments* Technical Report TR-95-2, Human Interface Technology Laboratory, University of Washington.

N.M. Robertson, I.D. Reid and J.M. Brady *What are you looking at? Gaze recognition in medium-scale images* Human Activity Modelling and Recognition (HAREM), British Machine Vision Conference (BMVC), Oxford, UK, September 2005.

N.M. Robertson and I.D. Reid *Behaviour understanding in video: a combined method* Proceedings of the International Conference on Computer Vision (ICCV), October 2005, Beijing, China.

N.M. Robertson and I.D. Reid *Estimating Gaze Direction from Low-Resolution Faces in Video* Proceedings of the 9th European Conference on Computer Vision (ECCV), Graz, Austria, May 2006

N.M. Robertson and I.D. Reid *Human activity recognition in video using a combination of parametric and non-parametric techniques* Journ. Computer Vision and Image Understanding (CVIU), Special Issue on Modeling People: Shape, Appearance, Movement and Behaviour, 2006.

A. Verri and T. Poggio *Against quantitative optical flow* Proc. IEEE International Conference on Computer Vision, London, 1987, pp. 171-180.

Index