Contents

CHAPTER 1

Looking at People

Understanding what people are doing by analysing video is one of the great unsolved problems of computer vision. A fair solution opens tremendous application possibilities, including: improved surveillance systems; a better understanding of what people do in public; better architectural design; and better human computer interfaces. The problem is difficult, because each of the component problems seem to be difficult, and because we're not yet sure how the components fit together. Detecting people and determining their configuration in an image is hard, most likely because of appearance variations (section **??**). Human motion appears to have a compositional property that complicates building representations (we review some background material from the animation community in section **??**). Tracking people in video remains difficult, probably because motion models don't help resolve appearance variations as much as one could hope for (section **??**). The relations between a 2D track and a 3D representation of the body are clear in outline, but important details remain obscure (section **??**); however, multicamera reconstruction is now quite well understood (section **??**). And finally, it isn't clear how one should represent activity. Simple discriminative methods work very well on simple problems (for example, telling "running" from "walking"), but more complex distinctions about composite activities ("watching tv while preparing food", say) remain difficult to draw (section **??**).

1.1 DETECTING AND PARSING PEOPLE

Notes: *Big issue here is what model and what features to use: should we encode kinematic knowledge implicitly or explicitly? and should we make features clothing, lighting invariant, or estimate appearance*

Notes: *(1) Why it is important. (2) What makes it difficult.(3) What cues could we use. (4) Strategies: Clothing independent implicit kinematics. (5) Strategies: Clothing independent explicit kinematics (6) Strategies: Clothing dependent explicit kinematics. (7) Notes: motion can help.*

A **human detector** needs to tell whether an image window contains a person at about the scale of that window or not. Human detection has several valuable applications. Pedestrian detection is worth doing well, because cars that can automatically detect and avoid pedestrians might save many lives (1997 figures give approximately half-a-million pedestrians killed by cars each year). Detecting humans is an important component in many surveillance applications because there are many places where detecting a human should immediately lead to an alarm (on an active runway, for example). People are the main subject of news pictures, home photo collections, broadcast video, commercial movies and home videos, so a human detector would be useful for building search and collection management tools for multimedia collections.

A **human parser** must produce some report of the configuration of the body in an image window. A human parse offers cues to what the person is doing, by reporting where the arms and legs and so on are. Applications could include building a user interface that can respond to someone's gestures or building a medical support system that can tell, by watching video, whether a physically frail person is safe at home, or has sustained an injury and needs care. Tracking people is a particularly useful technology (we'll discuss its applications below), and the currently most reliable technologies for human tracking involve a combination of detection and parsing. Detection could occur before parsing, in which case one would know that a window contains a person and must then determine how the body is laid out in that video. Alternatively, the two can be integrated, and detection would occur by searching a window for bits of the body and evaluating whether their layout suggests a person is present. It is still not certain which view is more useful.

Both detection and parsing are difficult problems. Many effects cause people to look different from window to window. There is a range of shapes and sizes of body. Changes in body configuration and in viewpoint can produce dramatic changes in appearance. The appearance of clothing can vary widely. As of writing, no published method can find clothed people wearing unknown clothing in arbitrary configurations in complex scenes reliably. The main cues to help overcome these difficulties are the fairly strong constraints on the layout of the body, and the relatively restricted appearance of a range of human body parts and configurations.

There are numerous possible kinematic cues. Many important problems involve walking or standing people, and these activities involve a relatively small range of body configurations, which are quite characteristic. Standing people have a "lollipop" (wide torso, narrow legs) appearance. Walking people tend to be in either this configuration, or in a "scissors" configuration with their legs apart. Some curves in the outline of the body, particularly the curve around the head and shoulders, can be quite distinctive and are fairly stable across viewing directions. This suggests using a model based on **implicit kinematics** encoded by a classifier; because the range of variation is relatively small and orderly, the number of training examples required to represent it is tractable. Building an implicit kinematic model that would accept every view of every configuration would involve a major engineering effort. Among other difficulties, the relative frequency of items within the training data would most likely misrepresent the relative frequency of configurations in real life. People are made of body segments which individually have a quite simple structure, and these segments are connected into a kinematic structure which is quite well understood. This suggests building a model based on **explicit kinematics** to represent the different configurations and views available.

A model that does not represent the layout of the body explicitly will have difficulty comparing appearance cues across the body. But if one uses a model that represents the layout of the body explicitly, there is a wide range of possible appearance cues. Body segments tend to be cylindrical, and so their edges will be roughly straight and roughly parallel. This means that **clothing independent methods**, which use features that are largely unaffected by the color and texture of clothing, can be successful. The advantage of a such a method is that we don't need to estimate the appearance of the clothing. The disadvantage is that there are useful cues that the method doesn't exploit. Segments on the left half of the body tend to have the same color and texture as segments on the right half of the body. Generally, the appearance within a segment is quite coherent (meaning that the color and texture at one place on, say, a lower arm will be quite like that at another). Quite often, most of the background does not look like the person. These observations mean that **clothing dependent methods**, which try to estimate the appearance of the clothing and then identify the body, can be successful, too. These methods must pay the cost of estimating the appearance, and are not obviously superior as a result.

1.1.1 Clothing-independent Implicit Kinematic Human Detection

Pedestrians can be detected quite successfully by a moving window technique. The main issue is building an appropriate feature to feed into a classifier. Pedestrians wear too many different kinds of clothing, and appear in too many different configurations, for just testing pixels to be successful. We expect to see some quite distinctive edge patterns, however, from, among other sources, the “scissor” and “lollipop” configurations of walking or standing pedestrians, and the curve around the head and shoulders. A pure edge map will not work as a feature because changes in illumination intensity or background will lead to contrast effects that tend to knock out useful edge points. Instead, orientations would be a better basic feature. The large range of spatial variations suggests using some form of histogram to smooth, but simply histogramming orientation over a whole image window will work poorly, too, because too many image windows will produce the same orientation histogram. A better approach is to break up the window into cells, which could overlap, and build an orientation histogram in each cell. The feature is now a vector made by stacking cell orientation histograms (each of which is a vector). This gives a feature that can tell whether the head and shoulders curve is at the top of the window or at the bottom, but will not change if the head moves slightly.

One further trick is required to make a good feature. Orientation is not affected by illumination brightness. This property has been useful, but prevents us from treating high contrast edges specially. For example, we are counting a light grey stripe on a slightly darker grey background in the same way as we are counting a black stripe on a white background. By doing this, we will make the distinctive curves on the boundary of a pedestrian count with the same weight as fine texture detail in clothing or in the background, and so the signal will be submerged in noise. We can recover contrast information by counting gradient orientations with weights. Gradients will add counts proportional to their magnitude to each cell histogram in which they appear, at the appropriate orientation. High contrast edges will dominate the histogram. However, matching will still be affected by illumination (if we double the light intensity, the histogram counts will all double). Correcting illumination across the whole window will not resolve this problem, either, because one side of the pedestrian may be shadowed and the other bright. Instead, we will normalize illumination locally in each cell. Doing so boils down to ensuring that each of the cell orientation histograms stacked into the feature vector is of the same total weight. If the cells overlap, which is the usual case, then our feature will implicitly represent several normalizations for each point.

The resulting feature is a variant of SIFT, known as a **HOG** feature (for

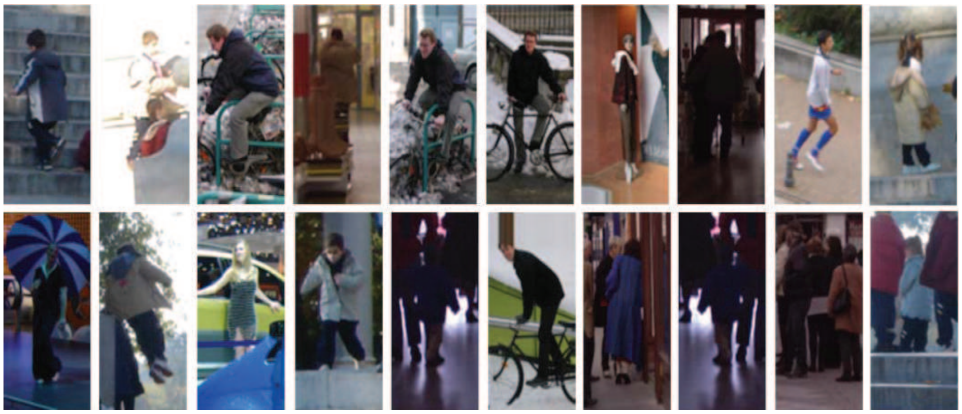


FIGURE 1.1: *Test examples that contain people but are misclassified by a reimplementation of Dalal and Triggs' [21] pedestrian detector, from [**?**]. Notice that unusual body configurations are quite common in this set (more so than in the test set). This suggests some errors made by that method are caused because the training set misrepresents the different configurations people can get into, as it must. Figure from "Configuration estimates improve pedestrian finding", D. Tran and D.A. Forsyth, NIPS, 2007*Shown in draft in the fervent hope of receiving permission for final version

histogram of oriented gradients), due to Dalal and Triggs [21]. This feature, coupled with a linear SVM, yields a well-behaved pedestrian detector. For example, Dalal and Triggs show this method produces no errors on the 709 image MIT dataset of [74]; they describe an expanded dataset of 1805 images.

Notes: *Figure from Deva + Pedro's paper*

1.1.2 Clothing-independent Explicit Kinematic Human Detection

Notes: *Work in Deva+Pedro here, too*

Pedestrian finders seem to fail preferentially on windows with unusual body configurations (Figure **??**), and this (if it is the case — it's still difficult to be sure) it is a major problem. We cannot run someone over because they get on a bicycle. An explicit kinematic model could help, by allowing us to encode all possible kinematic configurations rather than just all those observed in training data. Generally, we would exploit an explicit kinematic model by finding parts and then reasoning about their layout. The core idea is very old (early examples of this line of reasoning include [3, 4, 8, 42, 58, 71]) but the details are hard to get right and important novel formulations are a regular feature of the current research literature.

Most methods of this type represent a view of a person as a set of 2D body segments linked by rotary (and perhaps translational) joints, placed at the body joints. The body segments are usually torso, head, upper and lower arms and legs, and are usually represented by image rectangles of fixed size. Section **??** describes the pictorial structure model, which uses a spatial layout model to parse a person given an appearance model of that person. This model can be extended to detect people with unknown clothing in two steps. To produce a detector, we could either look at the cost of the best parse under the pictorial structure model, or apply a discriminative detector to the image domains identified by the best parse. To deal with the unknown appearance of the parts, we can build discriminative part finders.

The simplest way to build part finders is to build a labelled dataset containing image windows aligned to body segments and background windows, then train classifiers to detect these parts. The original approach uses filter outputs [], but one might use HOG features instead. We can then regard the strength of the part detector responses as a segment cost function (compare section **??**), and apply a pictorial structure model as before. Because some parts might be missing or occluded, it makes sense to build a model around a mixture of trees, rather than a single tree (again, compare section **??**). One problem with this approach is that the tree model and the body segment model are trained separately. This means we have missed the opportunity to bias the errors that the detectors make in a direction that the tree model is capable of fixing. It is possible to train both detectors and tree model simultaneously, using a procedure known as **structure learning**. Doing so requires a dataset of people where the full layout of the body has been labelled. To train the method, one chooses the parameters of the tree model and of the detectors so that, on the training data, the combination of tree model and detector selects a configuration close to that labelled. This means that, at each step of learning, we must run a pictorial structure model on every training data item. The technical details are beyond the scope of this chapter, but the procedure improves the performance of human detectors somewhat [**?**].

Body segments seem to be the “natural” parts from which to build a kinematic model, but the models that result are not entirely satisfactory. Parses tend to be poor, most likely because the kinematic model isn’t particularly accurate and neither are the part detectors. One source of inaccuracy is the common assumption that the segments are of fixed size. If we are building a detector, the model’s complexity might be unnecessary and might impede good performance. An important alternative, due to Felzenszwalb *et al.* [], is to allow the model to discover parts, which would be image patches that tend to: (a) be associated with the object;(b) be of coherent appearance and (c) turn up in similar locations relative to the object and one another. Doing this requires learning both part appearance models and part kinematics within the context of a small family of explicit kinematic models.

We can build a model at two scales. The first is the root scale, and some pattern should be detected at that scale to get a good detection. The root scale pattern is like a conventional window based detector. The second scale is the part scale, and we expect that several window detectors should respond strongly to smaller windows, close to particular locations relative to the root scale. What is important here is that these part detectors may have nothing to do with, say, arms or legs. Instead, they represent detail patterns that tend to appear close to particular locations. The model scores a window and a set of part locations by adding a score for the root scale, a score for how well each part matches the part scale window at its location, and a score for the locations of the part matches. The best value over all locations then gives the overall score for a window.

If we knew where each part detector should respond for every positive training image, learning such a model would be straightforward: we have positive and negative example image windows for the root scale and each part, and so can train those parts of the model using an SVM; and we have the location of each part, so we can use (for example) a maximum likelihood approach to build a location score. This suggests training by iteratively reestimating parts given a location model, then locations given a part model. As of time of writing, this class of model had the best scores in difficult human detection challenge problems [].

Notes: *Felzenszwalb figure*

1.1.3 Clothing-dependent Explicit Kinematic Human Detection

Appearance independent body part finders can work poorly, because the only cue available is the tendency of image edges around a part to look somewhat like the edges of a cylinder. Ramanan points out that we can deal with this is by estimating the appearance of the person []. Assume we know that a person is present in the window, and we have built a (necessarily somewhat unreliable) appearance independent body segment finder. We could use a pictorial structure model together with this body segment finder to obtain an estimate of the person's configuration. The result may not be right, but is unlikely to be completely wrong. Better, we could generate multiple estimates of configuration, using the procedure for sampling described in section **??**. These estimates appear with frequency proportional to the posterior. We can build a map of the posterior a pixel is, for example, a head pixel by rendering the head segment for each of these estimates of configuration and then summing the images. In turn, this means we have a set of weighted head/non-head pixels, which can be used to build a discriminative appearance model for the head. From this, we can build a map of the posterior that a pixel belongs to a segment, which is called a **parse**; such maps are conveniently encoded as an image with higher intensity for larger posterior values, and a different color for each segment. Some smoothing will help manage possible errors in the configuration estimate; for example, we might require that the appearance model has a mirror symmetry property. We now use a pictorial structure method with this appearance model to reestimate the configuration (Figure **??**). We then reestimate the appearance model, then the configuration, and so on. Again, the technical details of this procedure are beyond the scope of this chapter, but the procedure we have described can produce simultaneous estimates of parses and appearance models for complex images.

Reestimating appearance and configuration can be fooled if the human figure covers a relatively small percentage of the image area. In this case, there is the prospect that the initial estimate of configuration is wholly wrong, and there is little chance that reestimation will help here. This suggests that we should use other information to reduce the search domain, and doing so has been shown to produce very good upper body parses automatically (Figure **??**). The first thing to do is find an approximate search domain. We detect the figure's upper body, and then use the scale and orientation information for that detect to derive a box from the constrained length of the arms and from the fact that the torso is below the upper body. Everything outside this box is certainly not on the person. Because the torso is below the upper body and the detector is oriented, some pixels inside the box are

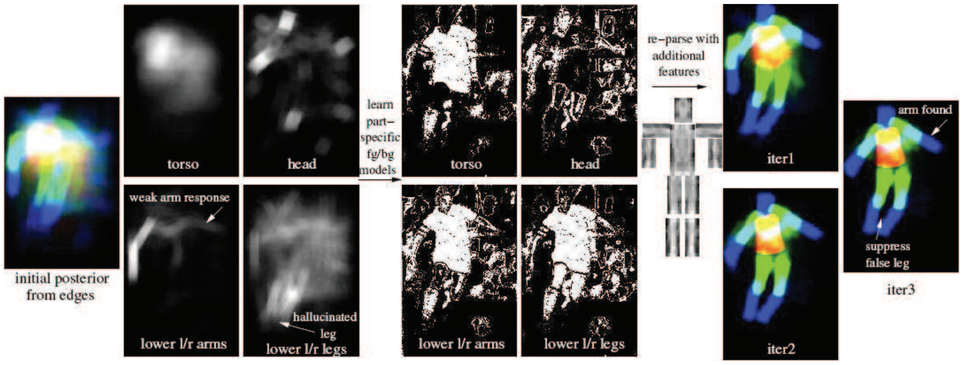


FIGURE 1.2: *Appearance independent body segment finders tend not to be particularly effective, because they can use relatively little image evidence. This suggests it is a good idea to estimate segment appearance models. Ramanan iterates between estimating configuration using an appearance model, and estimating appearance using the current set of configurations [***?***]. In the first step, one uses edges to produce an appearance independent segment finder, then produces multiple sample configurations from a pictorial structure model. Overlaying these configurations yields a posterior that a pixel belongs to a segment —* a parse *— which can be used to train a discriminative appearance model for the segment. The parses on the* **left** *come from an edge model, and in the* **center***, the pixels predicted to belong to each segment by the discriminative appearance model. Now we generate new configurations using this appearance model (labelled* **iter 2***, on the* **right***), and reiterate (***iter 3***, and so on, on the* **right***). Notice that in a few iterations we have a crisp parse — we know which pixels belong to arm, leg, and so on. Figure from "Learning to parse images of articulated bodies," D. Ramanan, NIPS 2006*Shown in draft in the fervent hope of receiving permission for final version

definitely on the person. We can now use an interactive segmentation method like Grabcut [] to segment an estimate of the person from the background. Grabcut uses a color model for foreground and background, built from various possible sources, to segment out a foreground. In this case, the background color model can be estimated from pixels outside the box, and some inside the box; the foreground color model can be estimated from some of the pixels inside the box; and we can constrain some pixels to be foreground in the final segmentation. Because the segmentation might not be precise, we can dilate it to get a somewhat larger domain. We now have a relatively small search domain and a very rough initial estimate of configuration to start the iterative reestimation process. Further constraints are available if we are working with a motion sequence; these are explored in section **??**.

1.1.4 Motion Features for Human Detection

Features for building human detectors are usually derived from arguments about kinematic behaviour or about appearance. Motion is another source of features, at least if one is working with video. Walking people in particular tend to move in

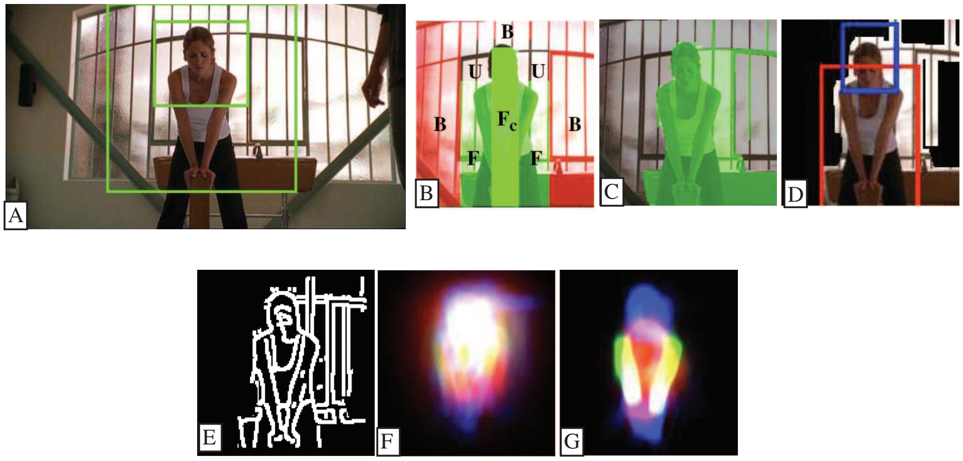


FIGURE 1.3: *The human parser of figure 1.2 is a search of all spatial layouts in the image to find one that is consistent with the constraints we know on appearance. Ferrari* et al. *show that reducing the search space improves the results. First, one finds upper bodies, and builds a box around those detections using constraints on the body size (***A***). Outside this box is background, and some pixels inside this box are, too. In* **B***, body constraints mean that pixels labelled F_c and F are very likely foreground, U are unknown, and B are very likely background. One then builds color models for foreground and background using this information, then use Grabcut to segment, requiring that F_c pixels be foreground, to get* **C***. The result is a much reduced search domain for the human parser, which starts using an edge map* **D***, to get an initial parse* **E***, and, after iterating, produces* **F***. Figure from "Progressive search space reduction for human pose estimation," V. Ferrari, M. Marín-Jiménez and A. Zisserman, CVPR 2008*Shown in draft in the fervent hope of receiving permission for final version

quite restricted ways, and, as Figure **??** shows, their movements leave distinctive structures in an **XYT image** (a stack of frames, registered as to camera motion, originally due to Baker [40]). These structures could be used to identify motions [68] or recover some gait parameters [67].

There are generally two strategies to exploit these characteristic spatio-temporal patterns to build human detectors. One could compute spatial features for each of a set of frames, stack these into a feature vector, and present the feature vector to a classifier. Doing so encodes dynamics implicitly, and can produce a significant improvement in detection rate for a given false positive rate [75]. Viola *et al.* use explicit motion features — obtained by computing spatial averages of differences between a frame and a previous frame, possibly shifted spatially — and obtain dramatic improvements in detection rates over static features ([104, 105]; see also the explicit use of spatial features in [20, 72, 73], which prunes detect hypotheses by looking for walking cues). This work uses a cascade architecture, as in section **??**.

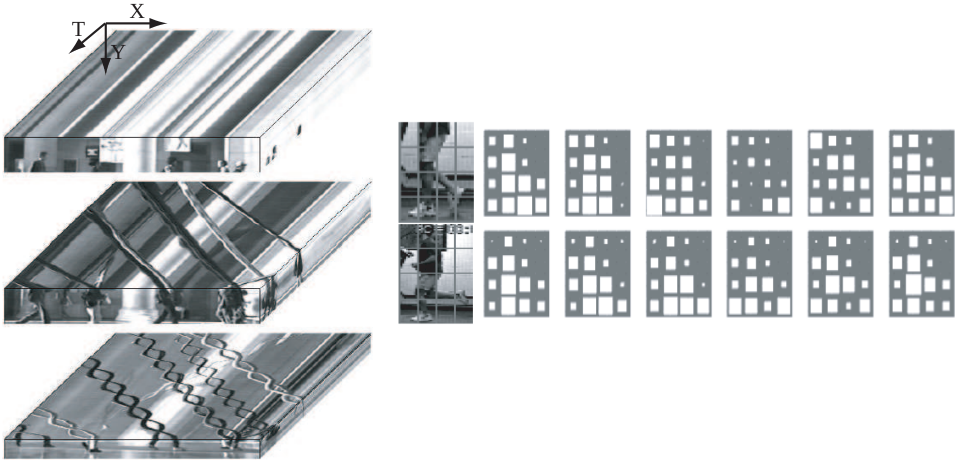


FIGURE 1.4: **Attribution:** *figure 2 of polana nelson recognizing activities , figure 2 of Niyogi Adelson recognizing gait On the* **left***, an XYT image of a human walker. The axes are as shown; the stack has been sliced at values of Y, to show the pattern that appears in the cross section. Notice that, at the torso there is a straight line (whose slope gives an estimate of velocity) and at the lower legs there is a characteristic "braid" pattern, first pointed out by Niyogi and Adelson [68]. On the* **right***, a series of estimates of the spatial distribution of motion energy (larger white blocks are more energy) for different frames of a walk (***top***) and a run (***bottom***); the frame is rectified to the human figure by translation, and one image frame from each sequence is shown. Notice that, as Polana and Nelson point out, this spatial distribution is quite characteristic [77].* Figure from "Recognizing Activities", Polana and Nelson, Proc. Int. Conf. Pattern Recognition, 1994, © 1994 IEEE. Figure from "Analyzing Gait with Spatiotemporal Surfaces", Niyogi and Adelson, Proc. IEEE Workshop on Nonrigid and Articulated Motion, 1994, © 1994 IEEE.

1.2 HUMAN MOTION AND COMPOSITION

Notes: *Main points: motion capture data is useful, easy to get and important; a primitive representation is available and seems about right but details are confused; composition is a big deal, because it affects how we think about motion*

1.2.1 Motion Capture Data

Motion capture refers to special arrangements made to measure the configuration of a human body with (relatively) non-invasive processes. Early systems involved instrumented exoskeletons (the method is now usually seen as too invasive to be useful except in special cases) or magnetic transducers in a calibrated magnetic field (the method is now usually seen as unreliable in large spaces). More recent systems involve optical markers. One can use either **passive markers** (for example, make people wear tight-fitting black clothing with small white spots on them) or **active markers** (for example, flashing infrared lights attached to the body). A collection of cameras views some open space within which people wearing markers move around. The 3D configuration of the markers is reconstructed for each individual; this is then cleaned up (to remove bad matches, etc.; see below) and mapped to an appropriate skeleton. Motion capture is a complex and sophisticated technology; typical modern motion capture setups require a substantial quantity of skilled input to produce data.

Typical workflow involves capturing 3D point positions for markers, discounting or possibly correcting any errors in correspondence by hand, then using software to link markers across time. There are usually errors, which are again discounted or corrected by hand. Motions are almost always captured to animate particular, known models. This means that one must map the representation of motion from the 3D position of markers to the configuration space of the model, which is typicaly abstracted as a **skeleton** — a kinematic tree of joints of known properties and modelled as points separated by segments of fixed, known lengths, that *approximates* the kinematics of the human body. Different approximations have different properties — the details are a matter of folklore — and one chooses based on the needs of the application and the number of degrees of freedom of the skeleton. Skeletonization is not innocent, and it is usual to use artists to clean up skeletonized data, essentially by adjusting it until it looks good. Data represented using one skeleton cannot necessarily be transferred to a different skeleton reliably. **Reviews** of available techniques in motion capture appear in, for example [10, 38, 56, 59, 61, 91].

For the moment, fix a skeleton. The configuration of the skeleton can be specified either in terms of its **joint angles**, or in terms of the position in 3D of the segment endpoints (**joint positions**). Not every set of points in 3D is a legal set of segment endpoints (the segments are of fixed lengths), so sets of points that are a legal set of segment endpoints must meet some **skeletal constraints**. The set of all legal configurations of the body is termed the **configuration space**; the joint angles are an explicit parametrization of this space, and sets of points in 3D taken with constraints can be seen as an implicit representation.

Sometimes one wants the motion capture data to drive a rendered figure. For example, when the actor moves an arm, the virtual character should do the same. The virtual character is represented as a pool of textured polygons, and one must determine how the vertices of these polygons change when the arm is lifted. The process of building a mapping from configuration — always represented as joint angles for this purpose — to polygon vertices is referred to as **skinning**.

An important practical problem is **footskate**, where the feet of a rendered motion appear to slide on the ground plane. In the vast majority of actual motions, the feet of the actor stay fixed when they are in contact with the floor (there are exceptions — skating, various sliding movements). This property is quite sensitive to measurement problems, which tend to result in reconstructions where some point quite close to, but not on, the bottom of the foot is stationary with respect to the ground. The result is that the reconstructed foot appears to slide on the ground (and sometimes penetrates it). The effect can be both noticeable and offensive visually. Footskate can be the result of: poorly placed markers; markers slipping; errors in correspondence across space or time; reconstruction errors; or attempts to edit, clean up or modify the motion. Part of the difficulty is that the requirement that the base of the foot lie on the ground results in complex and delicate constraints on the structure of the motion signal at many joints. These constraints appear to have the property that quite small, quite local changes in the signal violate them. It is likely that these properties are shared by other kinds of contact constraint (for example, moving with a hand on the wall), but the issue has not arisen that much in practice to date.

1.2.2 Composition, Motion Primitives and Motion Graphs

While human motion is complex, it does seem to be a composite of smaller pieces of motion. For example, when people walk they repeat roughly the same motion again and again. Many everyday motions are stereotyped. Think of reaching for a kitchen knife, chopping onions, climbing stairs, dressing, and so on. There is a fair body of practical evidence that motions are composites (or at least, that it is useful to pretend that they are). The simplest mechanism is **temporal composition**, where motions are strung together in time to produce a new, more complex motion. For example, a subject might walk into a room, halt, look around, walk to a chair and then sit down.

The use of motion capture data by, for example, the computer game industry reflects this belief. Typically, motions are created for a game by writing and capturing a script of motions, using a set of “complete” motions that start and end at one of a few **rest positions**. The motions can be thought of as building blocks which can be joined if one ends and the next starts at the same rest position. The choice of which block is joined to the end of the last block can be made by a game engine. Motions captured for a particular title are then usually discarded as re-use presents both economic and legal difficulties.

These blocks of motion can be thought of as **motion primitives**. There would be important advantages to knowing a large dictionary of motion primitives that can encode many motions well. Such a dictionary could be used to compress motion data. It could be used to produce long time-scale statistics about how motions are constructed, by representing motions with the dictionary and then looking for important co-occurrences. These seem to be non-trivial. For example, we know that people can walk backward and sometimes do; but if you want to move to a point a long way behind you, you will turn around and walk forwards toward the point. As another example, it is quite uncommon to reach in a direction you haven’t looked in recently. Long timescale activities — for example, visiting an ATM, or making dinner — can be seen as a sequence of motion primitives assembled according to a model. Building a dictionary of motion primitives seems to require iterative re-estimation. One uses an existing dictionary (equivalently, set of clustered motions) to segment a set of motion sequences, and then uses that segmentation to re-estimate the dictionary.

Estimating motion primitives well remains difficult. A more successful practical representation involves a more fluid encoding of possible transitions between motions, usually known as a **motion graph**. The details of how motion graphs are built and represented vary from author to author, but the simplest model regards every frame of motion as a node and inserts a directed edge from a frame to any frame that could succeed it. For example, a stack of observed motions is a motion graph, because there is a directed edge from each frame in a motion to the next frame in that motion. A more useful motion graph can be obtained by adding

computed edges, which identify transitions that could have been observed, but are not in the current dataset.

Computed edges can be inserted by matching. Write A_i for the i'th frame in a sequence A. Then if two frames are sufficiently similar, their futures (or pasts) could be interchanged. This means that if if A_i and B_j are similar, that means that four motion sequences are acceptable: $...A_{i-1}A_iA_{i+1}...$, $...B_{i-1}B_iB_{i+1}...$, $...A_{i-1}A_iB_{j+1}...$, and $...B_{j-1}B_jA_{i+1}...$. Frames can be matched using point locations and velocities. Once the graph is built, there are numerous methods for searching it to produce a motion that meets a demand, typically specified by a set of constraints. The underlying assumption is that any path in a motion graph will be a good motion.

For our purposes, what is important about motion graphs is that they work fairly well. Experience has shown that any path in a motion graph that does not involve too many computed edges does look very much like a human motion. These paths often have extremely high quality, though if there are many computed edges there is a tendency for the motion to pop or jitter (which is evidence that methods for identifying computed edges could be improved). This is evidence that human motion behaves as if it was produced by temporal composition.

Motions can be constructed by using different building blocks for different parts of the body. For example, it is possible to walk while scratching your head with one hand, and the arm motion involved in scratching your head with your left hand is basically a reflected version of the arm motion involved in scratching your head with your right hand. We refer to this idea as **composition across the body**. Such composite motions can be produced from motion capture data by cutting a limb off one sequence and attaching it to another sequence. Many such transplants are successful, but some apparently innocuous transplants generate motions that are extremely bad. It is difficult to be precise about the source of difficulty, but at least one kind of problem appears to result from passive reactions. For example, assume the actor punches his left arm in the air very hard; then there is typically a small transient wiggle in the right arm. If one transplants the right arm to another sequence where there is no such punch, the resulting sequence often looks very bad, with the right arm apparently the culprit. One might speculate that humans can identify movements that both don't look like as though they have been commanded by a normal central nervous system and can't be explained as a passive phenomenon.

1.3 TRACKING PEOPLE

Notes: *(1) Why it is important. (2) What makes it difficult.(3) What cues could we use. (3.5) what representation should we adopt (4) Strategies: Clothing independent implicit kinematics. (5) Strategies: Clothing independent explicit kinematics (6) Strategies: Clothing dependent explicit kinematics.*

Tracking people in video is an important practical problem. If we could tell how people behave inside and outside buildings, it might be possible to design more effective buildings. If we could reliably report the location of arms, legs, torso and head in video sequences, we could build much improved game interfaces and surveillance systems. Our observations on why detecting people is difficult apply to tracking people, too, as do our observations on what cues are available to help detect people.

Detection systems may have to deal with isolated images, but tracking systems never do. This means that tracking systems can exploit motion as a cue. Motion is almost certainly a useful cue for detecting people or segments. Motion can also contribute by predicting plausible locations for detections in the next frame, through some form of filtering procedure. This cue is currently — and, we thing, rightly — out of vogue, because people can produce large accelerations and move quite fast. This means that for 30Hz video, the configuration of the body in frame i doesn't constrain the configuration of the body in frame $i + 1$ all that strongly. While body configurations change quickly from frame to frame, appearance changes very slowly, particularly if one is careful about illumination. This is because people tend not to change clothes from frame to frame. Generally, building a good person tracker seems to involve paying close attention to image appearance and data association, rather than to dynamical models or probabilistic inference. As a result, recent methods strongly emphasize various tracking by detection ideas, and the main kinds of distinction between methods are the same as those for detection. Because we have multiple frames over which to build appearance models, and because it can be very valuable to tell people apart by the differences in their clothing, trackers with explicit kinematic models are always clothing dependent.

In tracking systems, our somewhat uncertain distinction between detection and parsing becomes a rich range of options for representing the body when we track. A range of levels of detail are useful. Representing a person as a single point is sometimes useful: for example, such representations are enough to tell where and when people gather in a public space, or during a fire drill. Alternatives include: representing the head and torso; representing the head, torso and arms; representing head, torso, arms and legs; and so on, down to the fingers. Tracking becomes increasingly difficult as the number of degrees of freedom goes up, and we are not aware of any successful attempts to track the body from torso to fingers (which are a lot smaller than torsos, which introduces other problems). Most procedures for tracking single point representations use the methods of chapter **??** directly, typically combining background subtraction with some form of blob appearance tracker. This section focuses on trackers that try to represent the body with fairly detailed kinematic models, because such trackers use procedures specialized for tracking people.

The state of the body could be represented in 3D or in 2D. If there are many cameras, a 3D state representation is natural, and multi-camera tracking of people against constrained backgrounds now works rather well. The flavour of this subject is more like reconstruction than like detection or recognition, and it doesn't fit very well into general pattern of single camera tracking. For reference, we give a brief review of tracking people in 3D using multiple cameras in section **??**. In many important cases — for example, an interface to a computer game — there will be only one camera. If we require a representation of the body in three dimensions, then we could use a 3D representation of state, perhaps joint locations in 3D, or a set of body segments in 3D modelled as surfaces. Alternatively, we could track the body using a 2D state representation, and then "lift" it to produce a 3D track. Relations between the 2D figure and the 3D track are complicated, and may be ambiguous. The heart of the question is the number of possible 3D configurations that could explain a single image, and this depends quite a lot on what we observe in the image.

Generally, we favor tracking using a 2D representation, then lifting the track to 3D and we will discuss only this strategy in any detail. This is mainly a question of clarity. Methods for tracking using 3D state representations must deal with data association and with lifting ambiguity simultaneously, and this leads to complexity. In contrast, tracking in 2D is in essence a data association problem, and lifting is in essence to do with ambiguity. Another advantage to working in 2D first, then lifting, is that the lifting process can use image evidence on longer timescales without having any significant effect on the complexity of the tracking algorithm. We will return to this argument in section **??**.

1.3.1 Clothing-dependent Explicit Kinematic Human Tracking

In section 1.3.1, we described methods to identify an appearance model for a person from a single image. Generally, the strategy was to find a small but plausible spatial domain in the image, then iterate configuration estimation and appearance estimation in that domain. In a motion sequence, we can build a much better appearance model by exploiting the fact that body segment appearance doesn't change over time. Furthermore, the sampling time of the video is relatively fast compared to body movement, which means we know roughly which search domain in the $n+1$'th frame corresponds to which in the n'th frame. This means that we can strengthen the appearance model by using multiple frames to estimate appearance. We can improve configuration estimates both by using the improved appearance model, and by exploiting the fact that segments move relatively slowly. Ferarri *et al.* show significant improvements in practice for upper body models estimated using these two constraints (Figure **??**).

There is an alternative method to obtain an appearance model. It turns out that people adopt a lateral walking configuration rather often, meaning that if we have a long enough sequence (minutes are usually enough), we will detect this configuration somewhere. Once we have detected it, we can read off an appearance model because we know where the arms, legs, torso and head are. The pictorial structure model can detect lateral walking configurations without knowing the color or texture of body segments. We set up ϕ to score whether there are image edges close to the edges of the segment rectangles, and use strong angular constraints in ψ to detect only the lateral walking configuration. The resulting detector can be tuned to have a very low false positive rate, though it will then have a low detect rate, too. Now we run this lateral walking detector over every frame in the sequence. Since the detector has a low false positive rate, we know when it responds that we have found a real person; and because we have localized their torso, arms, legs and head, we know what these segments look like.

We can now build a discriminative appearance model for arms, legs, etc. and use this in a new pictorial structure model to detect each instance of the person. We take example pixels from each detected segment and from its background, and use, say, logistic regression to build a classifier that gives a one at segment pixels and a zero otherwise. Applying these to the images yields a set of segment maps, and the ϕ for each segment scores how many ones appear inside the image rectangle on

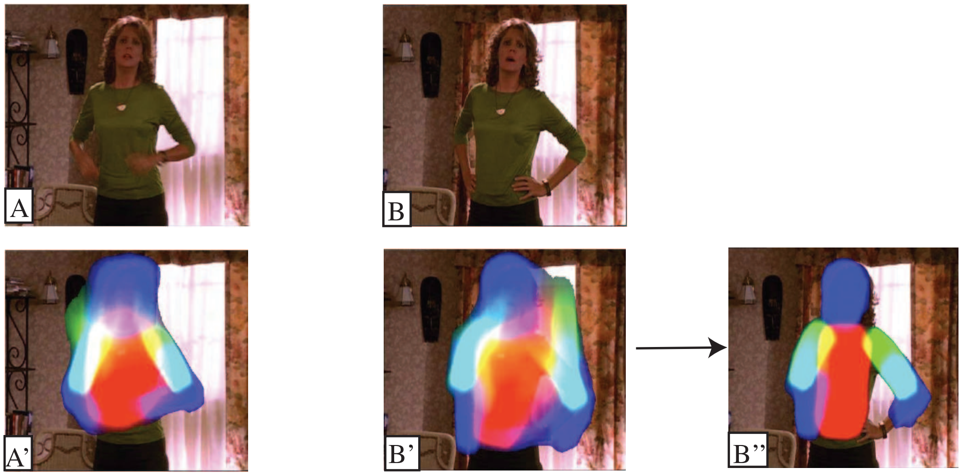


FIGURE 1.5: *Human body segments do not change appearance much over time, so that using multiple frames can yield a better appearance model and so a better parse.* **A** *shows a frame, and* **A'** *shows its parse, derived by Ferrari's method (from [**?**], described in section* **??** *and figure* **??**. *In this case, the parse has relatively low entropy — we have a fairly accurate model of where everything is. The frame in***B** *is more difficult, and a single frame method produces the parse of* **B'**, *which has relatively high entropy. By requiring that appearance be coherent over time, and that segments not move much from frame to frame, we can obtain the tighter parse of* **B"**. *Figure from "Progressive search space reduction for human pose estimation," V. Ferrari, M. Marín-Jiménez and A. Zisserman, CVPR 2008*Shown in draft in the fervent hope of receiving permission for final version

the relevant segment map. We can now pass over the video again, using a pictorial structure with weak constraints to detect instances of this person.

1.3.2 Clothing-independent Implicit Kinematic Human Tracking

Notes: *There are structural constraints not respected by the local body model that appear commonly, and we could benefit from this. There are few configurations for many movements. If we don't know segment appearance, then we need to work with edges. This suggests representing individual images as exemplars. Now our state is exemplar cross deformation. we track this with a particle filter*

Some human motions — walking, jumping, dancing — are highly repetitive, and the relatively free structure of a fully deformable model is not necessary to track them. If we are confident that we will be dealing with such motions, then we could benefit by using more restrictive models of spatial layout. For example, if we are tracking only walking people in lateral views, then there are relatively few configurations that we will see and so our estimate of layout should be better. There is another advantage to doing this; we can identify body configurations that are wholly out of line with what we expect, and report unusual behaviour.

Toyama and Blake encode image likelihoods using a mixture built out of tem-

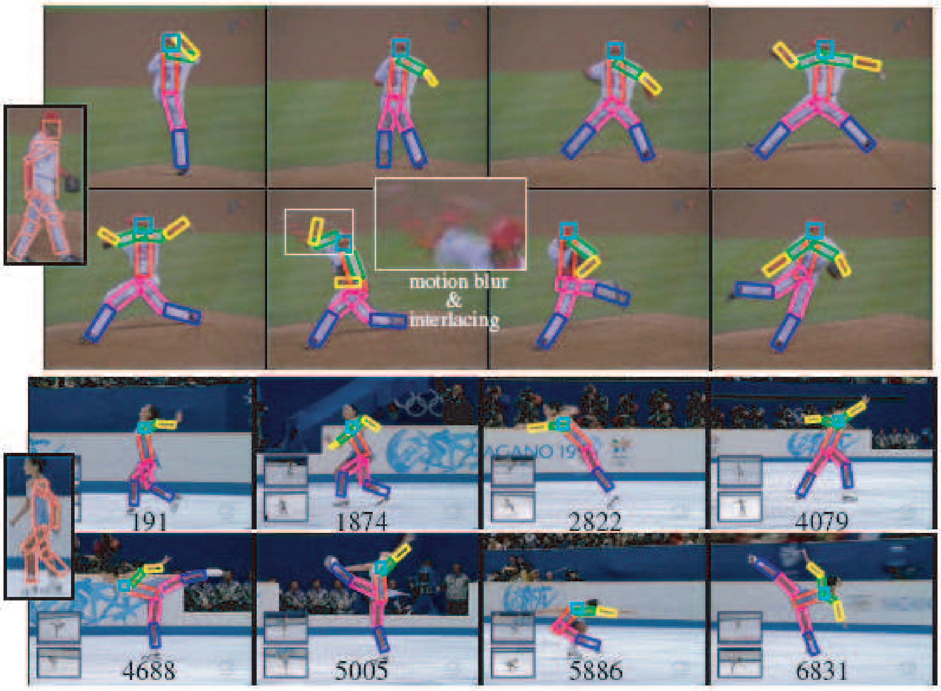
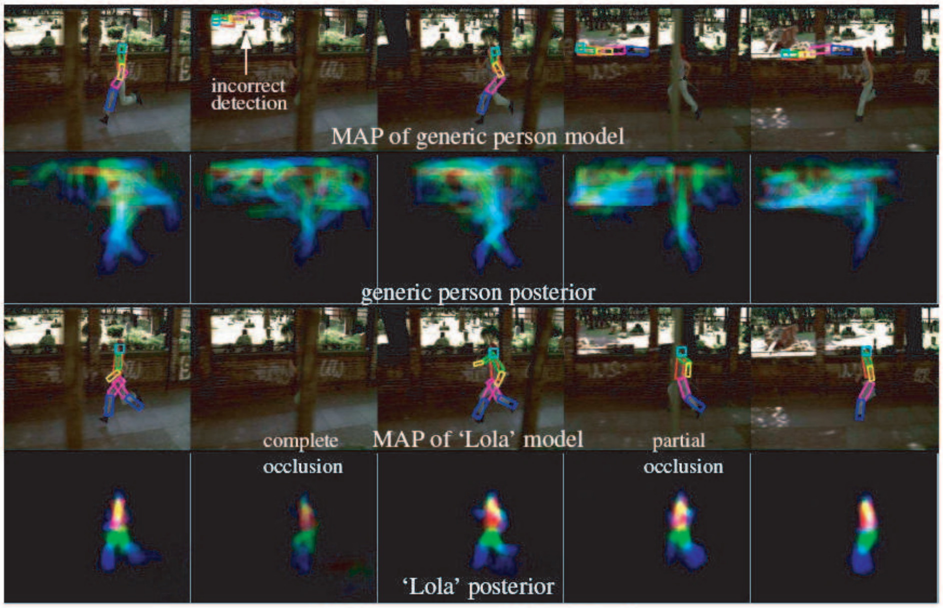


FIGURE 1.6: **Attribution:** *Deva's thesis Frames from sequences tracked with the methods of Ramanan* et al.*, where a discriminative appearance model is built using a specialized detector (figure* **??***), and then detected in each frame using a pictorial structures model. The figure shows commercial sports footage with fast and extreme motions. On the* **top***, results from a 300 frame sequence of a baseball pitch from the 2002 World Series. On the* **bottom***, results from the complete medal-winning performance of Michelle Kwan from the 1998 Winter Olympics. We label frame numbers from the 7600-frame sequence. For each sequence, the system first runs a walking pose finder on each frame, and uses the single frame with the best score (shown in the* **left insets***) to train the discriminative appearance models. In the baseball sequence, the system is able to track through frames with excessive motion blur and interlacing effects (the* **center inset***). In the skating sequqnce, the system is able to track through extreme poses for thousands of frames. The process is fully automatic.* Figure from Ramanan's UC Berkeley PhD thesis, "Tracking People and Recognizing their Activities", 2005 © 2005 D. Ramanan

plates, which they call **exemplars** [102, 101]. Assume we have a single template — which could be a curve, or an edge map, or some such. These templates may be subject to the action of some (perhaps local) group, for example translations, rotations, scale or deformations. We model the likelihood of an image patch given a template and its deformation with an exponential distribution on distance between the image patch and the deformed template (one could regard this as a simplified



% of frames correctly localized

Model	Torso	Arm	Leg
Generic	31.4	13.0	22.2
'Lola'	98.1	94.3	100

FIGURE 1.7: **Attribution:** *Deva's thesis Ramanan shows that tracking people is easier with an instance-specific model as opposed to a generic model [79]. The* **top two rows** *show detections of a pictorial structure where parts are modeled with edge templates. The figure shows both the MAP pose — as boxes — and a visualization of the entire posterior obtained by overlaying translucent, lightly colored samples (so major peaks in the posterior give strong coloring). Note that the generic edge model is confused by the texture in the background, as evident by the bumpy posterior map. The* **bottom two rows** *show results using a model specialized to the subject of the sequence, using methods described above (part appearances are learned from a stylized detection). This model does a much better job of data association; it eliminates most of the background pixels. The table quantifies this phenomenon by recording the percentage of frames where limbs are accurately localized — clearly the specialized model does a much better job.* Figure from Ramanan's UC Berkeley PhD thesis, "Tracking People and Recognizing their Activities", 2005 © 2005 D. Ramanan

maximum entropy model; we are not aware of successful attempts to add complexity at this point). The normalizing constant is estimated with Laplace's method. Multiple templates can be used to encode the important possible appearances of the foreground object. State is now (a) the template and (b) the deformation parameters, and the likelihood can be evaluated conditioned on state as above.

We can think of this method as a collection of template matchers linked over time with a dynamical model. The templates, and the dynamical model, are learned from training sequences. Because we are modelling the foreground, the training sequences can be chosen so that their background is simple, so that responses from (say) edge, curve, and the like detectors all originate on the moving person. Choosing templates now becomes a matter of clustering. Once templates have been chosen, a dynamical model is estimated by counting.

What makes the resulting method attractive is that it relies on *foreground enhancement* — the template groups together image components that, taken together, imply a person is present. The main difficulty with the method is that many templates may be needed to cover all views of a moving person. Furthermore, inferring state may be quite difficult.

1.4 3D FROM 2D: LIFTING

Notes: *Bunch of issues here: Original BarronKakadiaris/Taylor idea. Lifting is probably easier than people think. The ambiguities don't seem to exist in practice. Good methods: regress using nearest neighbours or do snippets.*

People in pictures typically are far from the camera compared to the range of depths they span (the body is quite flat), and so a scaled orthographic camera model is usually appropriate. One case where it fails is a person pointing towards the camera; if the hand is quite close, compared with the length of the arm, there may be distinct perspective effects over the hand and arm and in extreme cases the hand can occlude much of the body.

Regard each body segment as a cylinder and assume we know its length. If we know the camera scale, and can mark each end of the body segment, then we know the cosine of the angle between the image plane and the axis of the segment, which means we have the segment in 3D up to a twofold ambiguity and translation in depth (figure 1.8 gives examples). We can reconstruct each separate segment and obtain an ambiguity of translation in depth (which is important and often forgotten) and a two-fold ambiguity at each segment. We can now reconstruct the body by obtaining a reconstruction for each segment, and joining them up. Each segment has a single missing degree of freedom (depth), but the segments must join up, meaning that we have a discrete set of ambiguities. Depending on circumstances, one might work with from nine to eleven body segments (the head is often omitted; the torso can reasonably be modelled with several segments), yielding from 512 to 2048 possible reconstructions. These ambiguities persist for perspective images; examples appear in figure 1.9.

In this very simple model of the body, 3D reconstruction from a single image is ambiguous. However, the model oversimplifies in some important ways, and the true extent of ambiguity in this case is quite uncertain. One important oversimplification is that we assume that all 3D configurations are available. In practice, there are many constraints on the available joint rotations (for example, your elbow will move through about 70^0), so some of the ambiguous configurations might not be consistent with the kinematics of the body. Unfortunately, there is clear evidence that there are multiple kinematically acceptable reconstructions consistent with a single image (Figure **??**). It is not known whether there are multiple acceptable reconstructions associated with most images, or with only a few images.

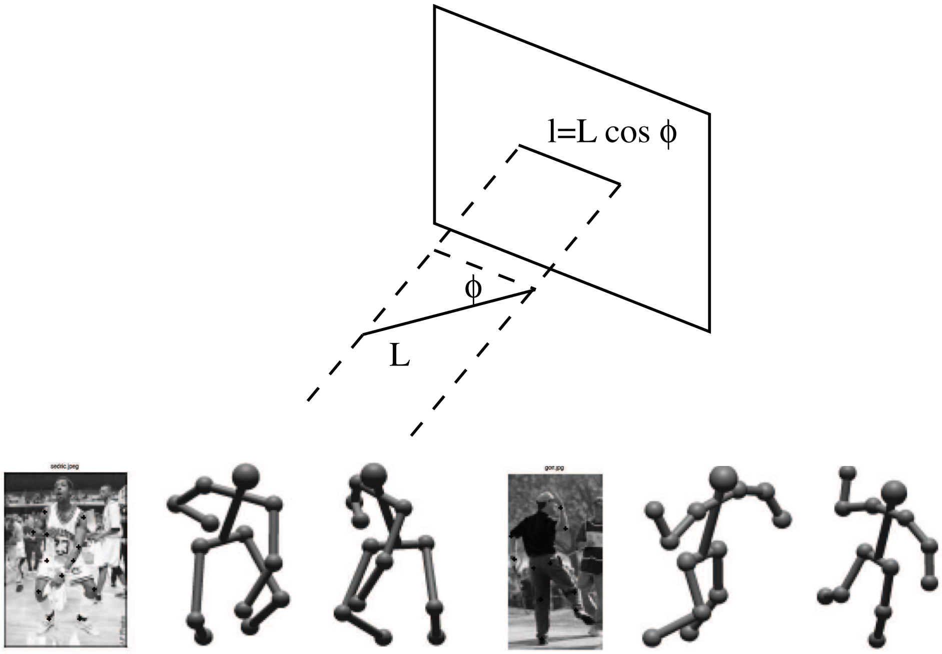


FIGURE 1.8: **Attribution:** *2d3Dlift/cvpr00.pdf, figure 5, p 682, CJpaper An orthographic view of a segment of known length L will have length $sL\cos\phi$, where ϕ is the angle of inclination of the segment to the camera and s is the camera scale linking metres to pixels (which is one in the figure above). In turn, this means that if we know the length of the body segment and can guess the camera scale, we can estimate $\cos\phi$ and so know the angle of inclination to the frame up to a twofold ambiguity. This method is effective;* **below** *we show two 3D reconstructions obtained by Taylor [99], for single orthographic views of human figures. The image appears* **left***, with joint vertices on the body identified by hand (the user also identifies which vertex on each segment is closer to the camera).* **Center** *shows a rendered reconstruction in the viewing camera, and* **right** *shows a rendering from a different view direction.* Figure from "Reconstruction of articulated objects from point correspondences in a single uncalibrated image", Taylor, Proc. Computer Vision and Pattern Recognition, 2000 © 2000 IEEE.

Another, more important oversimplification is that the body is not, in fact, an assembly of cylinders. Observing the shape of a hand, for example, might give enough information to tell whether the forearm is pointing towards the camera or away from it. There are methods for avoiding ambiguity that exploit this observation (section 1.6.1). Finally, we often observe motion sequences rather than a single frame, and there may be disambiguating information in the motion (section 1.4.2).

At this point, it is important to distinguish between two kinds of reconstruction. The first is an **absolute reconstruction**, which reconstructs the configuration of the body with respect to a global world coordinate system. The second is a **relative reconstruction**, where we seek the configuration of body segments with

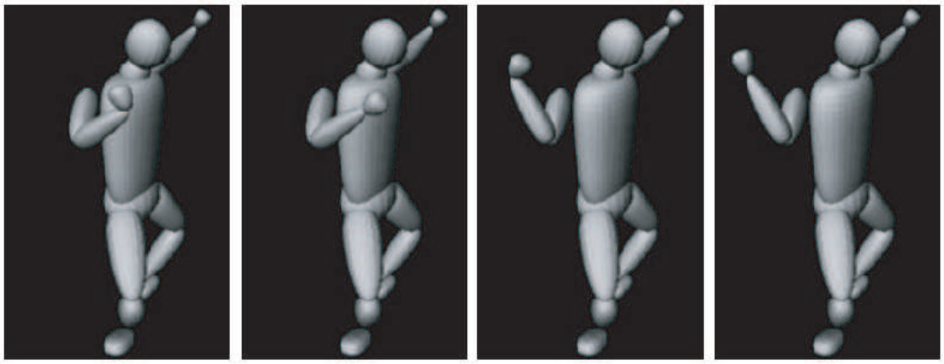


FIGURE 1.9: **Attribution:** *figure 2 of sminchisescu+triggs, kinematic jump processes, kinematicambiguity/01211339 Ambiguous reconstructions of a 3D figure, all consistent with a single view, from Sminchisescu and Triggs [94]. The ambiguities are most easily visualized by an argument about scaled orthographic cameras, given in the text, but persist for perspective views as these authors show. Note that the cocked wrist in the leftmost figure violates kinematic constraints — no person with an undamaged wrist can take this configuration.* Figure from "Kinematic jump processes for monocular 3D human tracking", Sminchisescu and Triggs, Proc. Computer Vision and Pattern Recognition, 2003 © 2003 IEEE.

respect to some **root coordinate system**. The root coordinate system is carried with the body, with its origin typically in the torso. All current work assumes a ground plane, and the root is usually oriented so that the z-direction is the up vector and rotation about z is usually given orienting either the hip or shoulder girdle along a coordinate direction. Absolute reconstruction is difficult, even with motion information, because each separate frame is missing a translation in depth and motion information is not usually enough to recover this. Absolute reconstruction with a moving camera is particularly tricky, because one would need good camera egomotion estimates to produce such a reconstruction (we are not aware of any in the literature at time of writing). Relative reconstruction is enough for most purposes. For example, absolute reconstruction doesn't seem to be necessary to label activities. As another example, most game interfaces are interested in running, reaching and jumping motions and the like, and a relative reconstruction is enough to identify these motions. It is important to be very careful reading the literature, which can be very confusing about this point, because most papers do not distinguish between absolute and relative reconstructions, and most methods sound as though they are producing absolute reconstructions but really produce relative ones.

1.4.1 Exploiting Appearance for Unambiguous Reconstructions

Disambiguating information might lie in the appearance of joints in the image, or in the appearance of the whole body. However, there is little theory that can guide us in building a model, and so it is more natural to exploit this information by

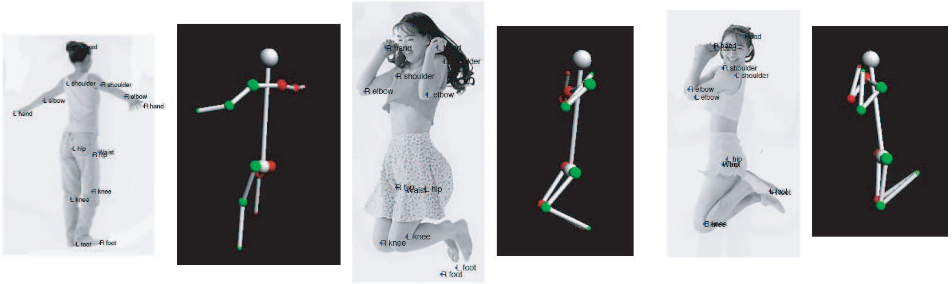


FIGURE 1.10: **Attribution:** *Mori+Malik, 2d3dlift/morimecv01.pdf, figures 6 and 7 Mori and Malik deal with discrete ambiguities by matching test image outlines to examplars, which have keypoints marked [64, 65]. The keypoint markup includes which end of the segment is closer to the view. The images on the* **left** *show example test images, with keypoints established by the matching strategy superimposed. The resulting reconstruction appears on the* **right***.* Figure from "Estimating Human Body Configurations using Shape Context Matching", Mori and Malik, IEEE Workshop on Models versus Exemplars in Computer Vision 2001 © 2001 IEEE.

matching to labelled examples in some way. We could match either local patches around each joint, or some representation of the whole body.

Local joint models: Mori and Malik deal with discrete ambiguities by matching [64, 65]. They have a set of example images with joint positions marked. The outline of the body in each example is sampled, and each sample point is encoded with a **shape context** (an encoding that represents local image structure at high resolution and longer scale image structure at a lower resolution). Keypoints are marked in the examples by hand, and this marking includes a representation of which end of the body segment is closer to the camera. The outline of the body is identified in a test image (Mori and Malik use an edge detector; a cluttered background might present issues here), and sample points on the outline are matched to sample points in examples. A global matching procedure then identifies appropriate examplars for each body segment and an appropriate 2D configuration. The body is represented as a set of segments, allowing (a) kinematic deformations in 2D and (b) different body segments in the test image to be matched to segments in different training images. The best matching example keypoint can be extracted from the matching procedure, and an estimate of the position of that keypoint in the test image is obtained from a least-squares fit transformation which aligns a number of sample points around that keypoint. The result is a markup of the test image with labelled joint positions and with which end of the segment is closest to the camera. A 3D reconstruction follows, as above (figure 1.10 gives some examples).

Whole body matching: Mapping an image of the body to a set of joint angles is regression, and the simplest regression method is to match the input to its nearest neighbor in a large training set, then output the value associated with that nearest neighbor. Shakhnarovich *et al.* built a data set of 3D configurations and rendered frames, obtained using POSER (a program that renders human figures, from Creative Labs). They show error rates on held out data for a variety of regression methods applied to the pool of neighbours obtained using parameter sensitive hashing. Generally, performance improves with more neighbours, with using a linear (rather than constant) locally weighted regression, and if the method is robust. The best is a robust linear locally weighted regression. Their method produces estimates of joint angles with RMS errors of approximately 20^o for a 13 degree of freedom upper body model [89]; a version of this approach can produce full 3D shape estimates [39].

1.4.2 Exploiting Motion for Unambiguous Reconstructions

In many applications there is a video sequence of a moving person. In such cases, it does not make sense to infer the 3D structure for each frame. It is a reliable rule of thumb from the animation community that most body motions are quite slow compared to reasonable video frame rates (evidence includes, for example, the relative ease with which motion capture sequences can be compressed with minimal loss []). This means that reconstructed body configurations for each frame will not be independent, and so each frame should affect the reconstructions of future and past frames. There might be quite strong constraints because the 3D reconstructions must join up in time well, and the 3D reconstruction of every frame in a sequence must be kinematically acceptable. This means that a single kinematically unacceptable reconstruction might be able to rule out a long ambiguous sequence.

One can incorporate dynamical information into the distance cost matching entire 3D motion paths to 2D image tracks, a method due to Howe [45]. For each frame of a motion sequence, we render every motion capture frame in our collection using a discretized grid containing every possible camera and every possible root coordinate system. Now we must construct a sequence of 3D motion reconstructions that (a) joins up well and (b) looks like the tracked frames. This is an optimization problem. We build a transition cost for going from each triple of (motion capture frame, camera, root coordinate system) to every other such triple. This cost should penalize excessively large segment and camera velocities. We compute a match cost comparing the rendered frame with the tracked frame. Write F_i for the i'th frame in tracked sequence, S for a reconstruction of that sequence and (L_i, C_i, R_i) for the reconstruction frame and camera corresponding to F_i. The cost function for a reconstruction is then

$$\text{cost}(S) = \sum_{i \in S} \text{transition cost}((L_i, C_i, R_i) \rightarrow (L_{i+1}, C_{i+1}, R_{i+1}) + \text{match cost}((L_i, C_i, R_i) \rightarrow F_i)$$

and in principle we can minimize this cost with dynamic programming. In practice, this would be very difficult to do, because there are a very large number of triples (L_i, C_i, R_i).

Some of this complexity is quite easily reduced. The number of cameras that could apply is quite small. For example, in many practical applications, the camera is orthographic and fixed, and the image plane is parallel to the up vector. This means that we can choose a fixed camera, and all unknown parameters are in the root coordinate system. Furthermore, we can estimate the image plane location of the root with elementary methods from the track. For example, we could place the root origin at the hips, and then estimate the location from the track. In this case, the unknown root parameters are translation in depth with respect to the camera, and rotation about the up vector. Our only cue for translation in depth with respect to the camera is a hope that changes in body configuration will reveal any significant changes in this parameter. This hope is probably misplaced, and it is better to leave out this parameter in the transition cost. It does not affect the match cost, which is why it is difficult to estimate. In this case, we finally need to discretize camera rotation, and this cannot be estimated with great precision, so the grid can be fairly coarse. Another important case that is relatively easy occurs when the camera is fixed, known, and looks down on the subjects. As long as we assume that people keep their feet on the floor, the translation of the root is easily estimated from the feet. Again, we need to search only root rotation, and again, this can be done with a fairly coarse grid. In either case, if the motion capture data set is very large, we may need to prune the frames further. One possibility is to cut redundant frames out of the motion capture dataset. Another is to avoid searching any triple where the match cost exceeds a threshold [45].

We can extend the method described to take into accelerations and higher order dynamics into account by matching short **snippets** (short runs of frames centered about a given frame) of motion capture to short snippets of video. To do this, we need to assume that the root moves relatively slowly with respect to the camera, so that using a single camera and root configuration for each snippet is acceptable.

Some ambiguities seem to have a long-term character. For example, it remains very difficult to tell whether the left leg or the right leg is leading in a lateral view of a walking figure. This is because very little in the image changes between these cases — there is little contrast between the trouser legs, so that it is hard to tell whether the left thigh occludes the right, or vice versa. Ambiguities like these might be resolvable by propagating disambiguating evidence over long time scales. For example, if one does not have a face detector, then it can be very difficult to tell which way a person is facing in a lateral standing view. However, if the person walks off (and if one assumes that the camera does not move fast), they reveal the direction in which they are facing, and this information can be propagated.

1.5 BODY RECONSTRUCTION IN 3D WITH MULTIPLE CAMERAS

Assume we have several calibrated cameras viewing a moving person. If we have an appropriate surface model of that person's body, we can reconstruct by finding the 3D configuration that generates images most similar to those we observe. The main questions are the choice of model, the choice of cost function for testing similarity, and how one searches for the best reconstruction.

Kehl *et al.* use a textured 3D mesh as a body model [50]. This mesh is the skin of a skeleton, and is controlled by its joint angle representation (section **??** for these terms). The texture maps are obtained from a modelling view. The cost function is the distance between sample points on the mesh (which are a function of the skeleton's kinematic parameters) and a visual hull. The hull is obtained by intersecting cones over foreground regions from between 4 and 8 calibrated cameras. The minimization procedure is a sophisticated variant of stochastic gradient descent. An alternative to comparing the visual hull with the 3D reconstruction

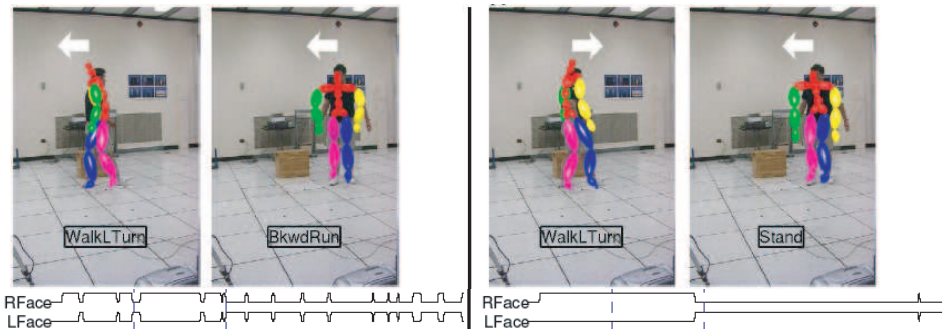


FIGURE 1.11: **Attribution:** *Figure 5.4 from Ramanan's thesis* **Left frames** *are taken from a walking sequence, matched to motion capture data using the method of Ramanan and Forsyth [80]. Matches are independent from frame to frame. Note that the lateral view of the body (***far left***) is ambiguous, and can be reconstructed inaccurately. This ambiguity does not persist, because the camera cannot move freely from frame to frame.* **Right frames** *show reconstructions obtained using dynamic programming to enforce a model of camera cost. The correct reconstruction is usually available, because the person does not stay in an ambiguous configuration. The frames are taken from a time sequence, and the graphs* **below** *show an automatically computed annotation sequence — facing left vs. facing right — as a function of time. Note that the case on the* **left** *shows an essentially random choice of direction when the ambiguity is present (the person appears to flip from facing left to facing right regularly). This is because the free rotation of the camera means the ambiguity appears on a per-frame basis. For the case on the* **right***, the smoothing created by charging for fast camera rotations means that the labels change seldom (and are, in fact, correct).* Figure from Ramanan's UC Berkeley PhD thesis, "Tracking People and Recognizing their Activities", 2005 © 2005 D. Ramanan

is to compute the silhouette of the 3D reconstruction, then compare that with the silhouette in each view. This can be done quickly in graphics hardware, yielding a cost function that can be evaluated very fast, allowing real-time tracking [17].

Stereo matches can give greater depth precision than the visual hull can provide. Plänkers and Fua estimate parameters for a model of the body consisting of a skeleton, metaball muscle model, and skin using stereo and, optionally, silhouette information [76]; the method appears to work with a complex background. Delamarre and Faugeras use a form of iterated closest point matching to produce forces that drive a 3D segment model into correspondence with the silhouette in three calibrated views [24, 25]. Drummond and Cipolla model the body with quadric segments, and track by applying a linearized flow model (as per section **??**; [12, 13]) to a search for edge points close to projected sample points on the model [32] (see also [31] for more information on the formalism, and [30, 33] for information about tracking changes in camera parameters). Shahrokni *et al.* use a similar general approach, but employ a novel texture segmentation model to find silhouette points [87]. They search along a scan line near and approximately normal to the

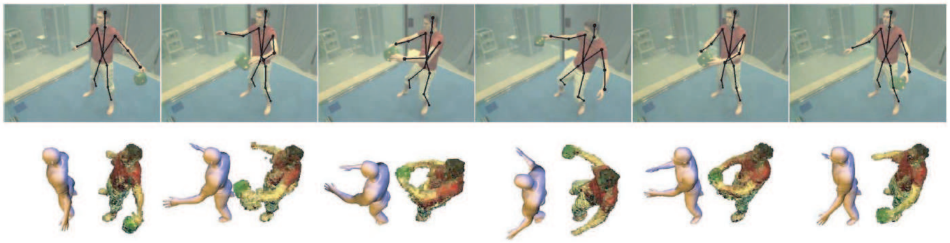


FIGURE 1.12: **Attribution:** *figure 12 of kehl ea, full body tracking using multiple views, multipleview/01467432 Kehl* et al. *represent the body as a textured 3D mesh, controlled by a skeleton with a texture map obtained from a modelling view. They obtain a volumetric reconstruction from a set of calibrated cameras, then track the body by minimizing distance between sampling points on the mesh and the volumetric reconstruction. The* **top row** *shows frames from one camera with reprojected skeleton superimposed; the* **bottom row** *shows the surface reconstruction at the* **left** *of each frame and the original volumetric reconstruction at the* **right**. *The reconstruction is accurate, despite some difficulties in the volumetric measurement.* Figure from "Full Body Tracking from Multiple Views Using Stochastic Sampling", Kehl *et al.* , Proc. Computer Vision and Pattern Recognition, 2005 © 2005 IEEE.

predicted silhouette to find points where there is a high posterior of a texture edge (see also an alternative method for finding texture silhouettes using a classifier in [88]; and using an entropy measure in [86]).

Texture information can be registered to the body model. Starck and Hilton obtain the best configuration of a 17 joint, meshed 3D model of the human body to fit stereo, silhouette and feature matches for each frame; texture is then reprojected onto the body (in [96]; see also [41, 98]). The texture is then backprojected onto the reconstruction and composited to give a single texture map. In recent work, Starck and Hilton show that correspondences between texture maps induced in separate frames yield temporal correspondences and so information on how relevant surfaces deform [97]. Models of this form allow relatively straightforward synthesis of new views [95]. These methods are oriented to performance capture, and appear to have been demonstrated for simple backgrounds only.

In principle, texture information registered to the body should yield a match score and improve matches, if the texture does not move with respect to the skeleton. We are not aware of methods that use this cue, though it may prove useful if one wants a detailed surface reconstruction of a model wearing tight garments. However, one can use a flow model to register texture from frame to frame. Yamamoto *et al.* use a linear flow model derived from the kinematic model (cf section **??**) with three cameras to obtain good tracks from hand-initialized data; they use three calibrated cameras [108]. The paper describes no difficulties resulting from movement of texture with respect to the body, but we expect that this effect significantly limits the precision of available reconstructions (see also figure **??**, and the discussion in section **??**). Theobalt *et al.* describe improved configurations obtained from the method of Carranza *et al.* ([17]) by incorporating an optic flow model to correct the estimates of configuration [100]. Subjects are not wearing very tight clothing, and there again seem to be no difficulties resulting from movement of texture with respect to the body.

Generally, search methods involve either standard optimization techniques or fairly standard variants. However, Deutscher *et al.* use a form of randomized search to align a 3D model with silhouette edges [26, 28]. Sigal *et al.* use a form of belief propagation to infer configuration in three or four views; the method uses detectors to guide a form of search [90]. Carranza *et al.* use a surface model, controlled by a 17 joint skeleton [17]. The search for a reconstruction at a time instant uses the reconstruction at the previous instant as a start point; however, because motion can be fast, and the sampling rate is relatively slow (15 Hz, p 571), a form of grid search at each limb separately is necessary to avoid local minima. A texture estimate is obtained by rectifying all images to the surface model, and blending.

Cheung *et al.* give an extensive discussion of representations of the visual hull and methods of obtaining them; the methods they describe can incorporate temporal information, color information, stereopsis and silhouette information [18]. Cheung *et al.* then use these methods to build a body model from a series of calibration sequences, which give both surface and skeleton information [19]. This model is then tracked by minimizing the sum of two scores. The first compares the deformed body model with the silhouettes in each image at a given timestep. The second compares an object reconstruction obtained at a given timestep with the silhouettes in each modelling frame. As authors note, there are 3D situations that are either kinematically ambiguous or at least very difficult for a tracking algorithm of this form. The first occurs when body parts are close together (for example, an arm pressed against the torso) and may lead to a self-intersecting reconstruction. This difficulty appears to be intrinsic to the use of silhouette features. The second occurs when the arm is straight, making rotation about the axis of the humerus ambiguous. The difficulty is that the photometric detail is too weak to force the method to the right configuration of the hand. Curiously, although Mori and Malik have shown that one can obtain the positions of landmarks such as the location of the hand, the knee and so on automatically, there appears to be no multiple view reconstruction work that identifies landmarks in several views (with, for example, the method of Mori and Malik, section **??**) and builds a geometric reconstruction this way.

1.6 WHAT ARE PEOPLE DOING?

The reason for all this work building signal representations is to determine what people are doing.

`simple discriminative methods work well`

Simple discriminative methods can work well at spotting some activities. There is a significant literature on classifying short video sequences into a small set of activity classes (for example, “walking”, “running”, “jumping” and so on). For many natural choices of classes, this problem is largely solved — one can get very good results with quite straightforward methods on appropriate datasets. We sketch such methods briefly, because they can be both useful and accurate. However, it is very difficult to move to more complex problems, because we do not possess any taxonomy into which a wide range of activities could be classified. This makes it hard to build discriminative methods, because we don't know the classes into which we should classify. Worse, there might not be such taxonomy, because people seem to interpret many activities in terms of the intentions of the actors, rather than names for what they are doing. Even if there is a taxonomy, the tendency of motions to compose (see above) suggests that it is very complicated indeed. We can be nearly certain that we will need to name activities which we have never seen before.

Another important difficulty is that performance figures for activity problems can be profoundly misleading. Very often, we want to demonstrate that we can identify uncommon phenomena, for which we posess few examples, with high accuracy. This is not the same as identifying common phenomena well. For example, labelling every person in surveillance video of a shopping mall as "walking" is probably very accurate (almost certainly in the high ninety percents), because that is what people tend to do in surveillance video. It is, however, wholly unhelpful, because surveillance problems require us to identify unusual or threatening behavior with high accuracy.

1.6.1 Appearance Features Moving People

An alternative to tracking the body and producing a 3D representation of its movement is to build image-based features that encode the body configuration well, and then match them directly. This idea has had considerable success, because many body motions produce quite characteristic space-time patterns. For example, if one were to stack a series of frames of video into an XYT image, there are quite distinctive structures, often called **braids**, that appear at the legs of walking pedestrians (Figure **??**). There are now several space-time representations available, all of which perform similarly well. We describe one construction here in detail because it illustrates the general points fairly well, but emphasize it is one possibility drawn from a family.

Building an Appearance Feature.

We would like to label individual frames of video, using a representation that encodes whatever is likely to be useful. There are some rough guidelines on how to build such a representation.

For a single frame, the shape that the body adopts is likely to be useful. We assume that we can segment the body from the background, but expect that this segmentation is somewhat rough. However, a detailed representation of shape is probably not needed. Flow is likely to be useful. We will estimate optic flow at each pixel by matching small image windows with sum-of-squared differences (see section **??**). This procedure is likely to produce a somewhat inaccurate estimate of flow, particularly if people are relatively small in the frame (Figure **??**). We will need to do some form of flow smoothing, in a way that preserves what is important.

The difficulty with smoothing flow is that it is a field of directions. If we were to smooth the components of flow one by one, then it might be possible to average a large upward going flow vector with a large downward going flow vector

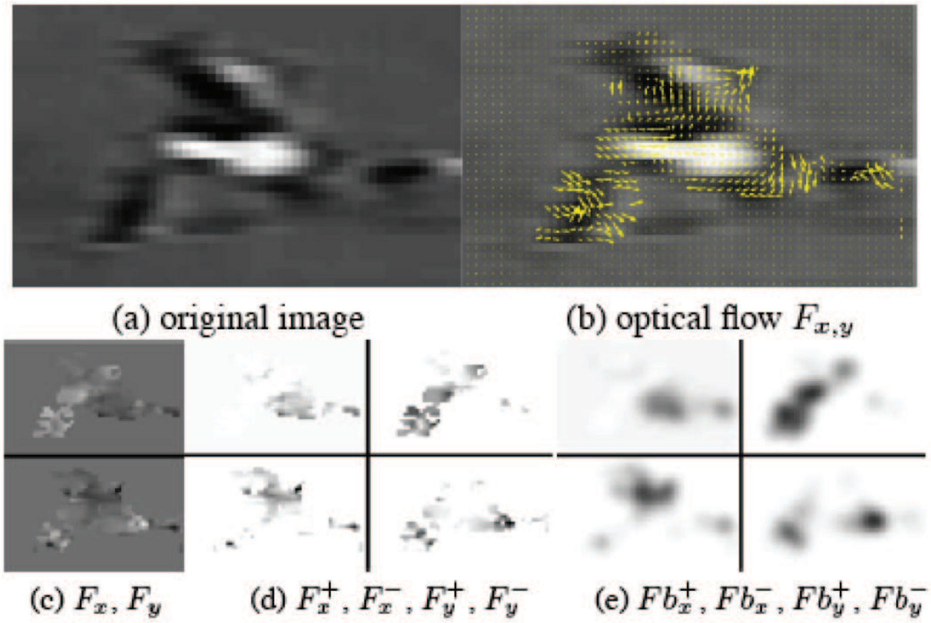

(a) original image
(b) optical flow $F_{x,y}$
(c) F_x, F_y
(d) $F_x^+, F_x^-, F_y^+, F_y^-$
(e) $Fb_x^+, Fb_x^-, Fb_y^+, Fb_y^-$

FIGURE 1.13: *Optic flow estimates from stabilized human figures can be very noisy, because the figure has low resolution in the image. Straightforward smoothing does not produce a useful feature, because one might average a large upward going flow vector with a large downward going flow vector to get a zero flow. An alternative procedure, due to Efros* et al. *[], is to rectify the flow components into positive (resp. negative) x (resp. y) directions, then smooth these rectified values.* **A** *shows a frame from a video sequence and* **B** *shows the flow vectors estimated by Lucas-Kanade (section* **??***).* **C** *shows the x and y components of flow, with white being maximum and black being minimum values;* **D** *shows the four rectified flow components, and* **E** *shows the smoothed components. Note that a fairly clear approximate picture of the overall motion appears.*

and conclude that the result is no flow. This is a poor representation. A useful and straightforward strategy, originally due to Efros *et al.* , is to take the x- and y-components of flow, rectify them to get four flow maps (the magnitude of the positive component of the flow in the x-direction, of the negative flow in the x-direction, and of each in the y-direction), and then smooth these maps (Figure **??**). This approach is simple, and makes large movements in any direction obvious in the flow maps.

The spatial distribution of flow in these maps is important, though, as for the shape of the segmented body, it is unlikely that we can measure or benefit from fine details. A histogram is a good way to produce a rough spatial map, though averaging over the whole body involves more smoothing than is desirable. We can place an axis-aligned box around the segmented body, rectify that box to a fixed size square, and describe each quarter of the square. In turn, each quarter needs a somewhat detailed spatial representation; a fair choice is to break it into windows, and count how many foreground pixels lie in each window. Tran *et al.* break each quarter into 18 pie slices, and count the percentage of foreground pixels in each slice. Similarly, they average the flow channels in each slice, too. This gives a histogram for each frame.

Past and future frames are almost certainly helpful, because (for example) to tell that someone is stopping we will need to know that they were moving; to tell that someone is starting we will need to know that they were stationary and will move; and so on. However, relatively little detail is going to be needed from the frames in the deep future or the distant past. In turn, this suggests collecting histograms for all frames in a window, but suppressing detail for some of those frames. Tran *et al.* describe a frame with (a) its own histogram (b) the first 50 principal components of the descriptors of a window of size 5, centered at that frame and (c) the first 5 principal components of the windows of size 5, centered at the $i + 5$ th and $i - 5$ th frames.

Matching Appearance Features.

Appearance features are very effective at simple activity classification problems and there are now several important datasets which can be used to test an appearance feature. `description of datasets`

In outline, one segments the body, computes an appearance feature like that above, then uses labelled data to build classifiers for activity labels. This is a successful procedure, as far as it goes, but there are some points that it is important to get right. In almost any conceivable application, we expect to see activities that do not belong to any of the available classes. This means that we must (a) allow our classifier to reject test examples and (b) evaluate its performance at doing so. A particular danger is caused by the internal correlations in motion sequences. As an extreme example, it is a very bad idea to use all the odd-numbered frames as training data and the even-numbered frames as test data. Caution suggests that no sequence should contain frames that are used for training and also frames that are used for testing. It is important to test with multiple actors and with multiple clothing styles, but the evaluation procedure should take this into account. Ideally, all test sequences feature actors different from those in the training sequences. Most methods score close to 100% accuracy on most recent datasets. This means that innovations are very difficult to evaluate (how can you tell if what you did made anything better if the basic method made only one mistake?) and means that more data will need to be collected which aims to identify what a particular construction can do well or badly. The ideals here can be very expensive in data, and methods to develop large, reasonably well-labelled activity dataset remain of considerable interest.

Aspect and Appearance Features.

The same activity can look quite different in different views, an effect known as **aspect**. We take an extensive view of this phenomenon: objects can change appearance because one sees a different outline, because occlusion relations change, because changes in viewing direction affect apparent color and texture, or because illumination has changed. This effect creates important difficulties for appearance methods, because we might need to possess several examples of the same activity under each set of viewing conditions. Figure **??** shows how significant changes in aspect can be; the images are taken from the IXMAS dataset, which at time of writing was the only dataset that carefully investigates the effect of aspect on the appearance of activities.

A core problem is to build a recognizer that can be trained from some aspects, and will work successfully on new views: we refer to this property as **transfer across aspect**. The problem remains largely open, though Farhadi *et al.* describe one method that is quite successful on the IXMAS dataset []. The trick is to build a family of classifiers for each activity; the particular classifier from the family is chosen by an estimate of the viewing conditions. An alternative strategy might be to build features that are largely unaffected by change in viewing direction. Junejo *et al.* show that this can be done by encoding the similarity between a frame and future and past frames; it turns out that this degree of similarity is largely unaffected by view direction, and is quite discriminative.

1.6.2 Hidden Markov Models and Activity

Notes:

Hidden Markov models (HMM's) pervade studies of motion, gesture and activity, and a complete review of their applications here may now be impossible. HMM's are models of sequences, and at their heart is a clock. One has a set of hidden states; at each tick of the clock, a Markov process chooses a new state, dependent on the previous state and nothing else; and an emission process produces an observation from the new state. There are clean solutions for the standard problems of learning (determining an appropriate state transition model and emission model for a given state model) and inference (determine which hidden states occurred given a set of observed states). HMM's have been used for understanding human behaviour but typically with quite small state models.

Very large state models are common in speech recognition, where HMM's have been hugely influential. We do not propose to engage in speech research, and so do not review the area here. It is purely a source of inspiration by analogy. Viewed from a great height, a typical speech system has a series of components: a language model showing how words are built up into sentences; a pronunciation dictionary, giving sequences of context independent phones that correspond to words; a context dependency model, showing how local influences produce context dependent phones (cphones hereafter) from context independent phones; an acoustic observation model showing how acoustic observations result from context dependent phones (this is an extremely compact description of a highly sophisticated area; more extensive descriptions appear in [46, 78]). The resulting object is a vast HMM — in our example, states can be thought of as being tagged with word-cphone-phone-sample — to explain each sample.

This HMM has some important, attractive features. Learning and authoring can be broken into tractable subproblems — the language model might be learned with one kind of dataset, the pronunciation dictionary with another — and as a result, we obtain an HMM on a massive scale, but with little difficulty in authoring it. While the state space is so big that dynamic programming must be sacrificed for a beam search, the state transition model is not impossible to learn, because most state transitions don't occur. Furthermore, the model is forced to share parameters in important ways — a phoneme in one word has the same model as that phoneme in a different word. The currently dominant method for authoring such models involves finite state transducers (section **??**); we propose to

Methods based on Hidden Markov Models: HMM's have been very widely adopted in activity recognition, but the models used have tended to be small (for example, one sees three and five state models in [11, **?**]). Yamato *et al.* describe recognizing tennis strokes with HMM's [109]. Wilson and Bobick describe the use of HMM's for recognizing gestures such as pushes [106]. Yang et al use HMM's to recognize handwriting gestures [110]. Feng and Perona [36] call actions "movelets", and build a vocabulary by vector quantizing a representation of image shape, as a collection of rectangle, varying over time. These codewords are then strung together by an HMM, representing activities.

There has been a great deal of interest in models obtained by modifying the HMM structure. The intention is to improve the expressive power of the model without complicating the processes of learning or inference. Brand et al use coupled HMM's (CHMM's), which involve some number of simultaneous HMM's operating to the same clock, where the choice of a particular model's hidden state is affected by all other model's states [11, **?**]. Such an object is clearly itself an HMM, but authors demonstrate a training method that reduces the number of parameters to learn by coupling the two models. They show these models can distinguish between a set of T'ai Chi moves.

Oliver et al [70, 69] represent behaviours using layered hidden Markov models (LHMM's). These models involve a bank of HMM's at the lowest level, each generating some portion of the observation. The observations at higher levels are the maximum likelihood hidden state sequences for the lower levels. The resulting object is an HMM, but of complex structure; the LHMM form offers authoring advantages. This representation outperforms a straightforward HMM in recognizing such activities as phone conversation from both vision and acoustic data.Similarly, Mori et al build a hierarchical representation out of HMM's to recognize everyday gesture [66].

Wilson and Bobick [**?**] use a form of HMM where an unknown, global parameter applies to all emission models (which they call a parametric hidden Markov model or PHMM) to model gestures with a parametric form (such as might accompany "it was *this* big"). Data is from stereo or a Polhemus. There are recognition results for classes of gesture such as pointing. Kettnaker and Brand [**?**](also, Brand and Kettnaker, [**?**]) fit an HMM while penalizing model entropy; this tends to reduce the number of non-zero parameters, so that one can fit models with quite large state spaces satisfactorily (such models are sometimes known as Entropic HMM's or EHMM's). Galata *et al.* use variable length Markov models (VLMM's: a model that generates a state stochastically based on a variable but bounded length history) to encode behaviour and obtain a reduction in perplexity by doing so [**?**, **?**].

Building variant HMM's is a way to simplify learning the state transition process from data (if the state space is large, the number of parameters is a problem). But there is an alternative — one could author the state transition process in such a way that it has relatively few free parameters, despite a very large state space, and then learn those parameters.

Finite state methods have been used directly. Hongeng *et al.* demonstrate recognition of multiperson activities from video of people at coarse scales (few kinematic details are available); activities include conversing and blocking [44]. Zhao and Nevatia use a finite-state model of walking, running and standing, built from motion capture [111]. Hong *et al.* use finite state machines to model gesture [43]. We are not aware of material that attempts to build large hierarchical finite state machines, patterned after speech recognition programs, and using opportunistic learning, as we propose to do.

1.6.3 Composite Representations of Activity

We have seen two schemes for extracting a representation of the body from video. In the first, one produces a 3D representation of body configuration, either by tracking in 3D or — and we prefer this — tracking in video and then lifting to 3D. In the second, one encodes the appearance of the body over time directly. Appearance representations are attractive, because they perform well in discriminative tests. While appearance representations suffer from aspect effects, there is some progress on managing the problem. There are very serious noise problems with 3D representations, because trackers still make errors. These errors can be very large, and can disrupt the lifted track over time through the smoothing processes we described. As a result, 3D representations can be quite noisy, and difficult to work with.

Nonetheless, we believe that 3D representations are currently better suited to studying human activity. This is because composition seems to be a fundamental property of activity (section **??**). Both appearance and 3D representations respond well to composition across time. One reasonable strategy to recognize a sequence of activities is to apply a classifier to a window around each frame in the sequence, then smooth the labels over time. A technically more sophisticated way to achieve this is to use a conditional random field []. There is no real difficulty with using either an appearance or a 3D representation here.

There is a very real difference between the representations when one considers composition across the body. In this case, we may need to use the fact that we have seen a waving arm and walking legs to recognize someone walking and waving *without* having seen an example of that particular activity. This is where the really big difference between a 3D representation and an appearance representation comes into play. A 3D representation must segment the body into components and appearance representations do not (if they did, one could then immediately build a 3D representation out of the pieces). This difference is important for recognizing activities that are composed across the body, because this segmentation tells us where composite representations could be joined up.

There is some literature on recognizing temporal composite activities, but there is very little on activities composed across the body. One difficulty is that this case does not fit well in known recognition paradigms. We expect that there are very many composite activities — which one should we report for a particular video sequence? The most probable might be a poor report, because it is quite likely that we will get it wrong (because there are so many possible reports). More important, it might not be relevant to our reason for observing activity. For example, if we observe the behavior of other people to avoid getting assaulted, then we really just want to be alerted to the occurrence of a subset of the collection of composite activities.

It is a little easier to formulate the problem as search. Assume we have a set of video clips showing various activities. We must then rank the clips in order of relevance to a search query for a novel activity. The query is written in some query language that allows composition across time and across the body, and the real test of a process like this is to get good responses to search queries that (a) are complex and (b) were not used in setting up the search (i.e. training the methods used in searching).

Ikizler and Forsyth demonstrate a method of this form [], which we describe briefly because it shows one way to attack quite general activity problems. The video is represented in using tracks lifted to 3D. They build motion models for arms and for legs executing one of a set of 13 activity labels using labelled motion capture data. The models are simple hidden Markov models, which they compare to phoneme models in a speech representation. They now have a stack of local motion models. They add transitions between states in different models of the arm (respectively leg) if the transition implies a relatively small movement (the criteria for building motion graphs of section **??** could be applied here). Now they build a larger model, each of whose states is a pair of arm and leg states with consistent torso, and using the transitions of the arm and leg models. The result is a large finite state model for body configuration; however, relatively few parameters must be learned to create the model. The measurements are vector quantized configurations in 3D. The emission model is learned from data.

The query language is built around units for arms and legs that are strung together with a finite-state automaton. The units are motion annotations like walk, run, jump, turn, reach and so on. One writes an automaton to accept any query of interest — for example, strings that look like `anything` followed by `arms-walk, legs-walk` followed by either `arms-reach, legs-walk` or `arms-wave, legs-walk` followed by `arms-walk, legs-walk` then followed by `anything`. There is then a technical trick to compute the posterior that the video contains any string accepted by the automaton. Results are quite promising; generally, the videos at the top of the ranking are relevant, and those at the bottom are less so. The main difficulties are caused by the noisy lifting process, though replacing this process with discriminative procedures doesn't seem to help much.

1.6.4 Alternative Cues to Human Activity

`Objects that are nearby; location of the person`

1.6.5 Important Open Problems

`Different representations work for different applications`

`Some cues are wierd - for example, location, etc.`

Some problems are well understood. If people are relatively small in the video frame, and the background is stable, it is easy to detect the people by subtracting a background image from the current frame. If the absolute value of the difference is large, this **background subtraction** declares the pixel to be a foreground pixel;

by linking foreground blobs over time, we obtain a track. Chris Stauffer and Eric Grimson have demonstrated that these tracks reveal a great deal about what people are doing. For example, in views of a parking lot, the shape of the track will show `What? can't recall`. Wei Yan and David Forsyth have demonstrated that observations like these can reveal information useful in architectural design, such as how long people sit at a fountain and what paths they take when they walk through an open plaza.

Video that shows rather structured behaviours, like ballet, gymnastics, or tai chi, where there are quite specific vocabularies that refer to very precisely delineated activities on simple backgrounds, is quite easy to deal with. Very good results are obtained by using background subtraction to identify the major moving regions, building features using either the HOG construction or something like it (as an added wrinkle we must keep track of flow, rather than just orientation), and then presenting these features to a classifier.

More general problems remain open. One source of difficulty is that we lack a simple vocabulary of human behavior. Behavior is quite like color, because people tend to think they know a lot of behavior names but can't produce long lists of such words on demand. There is quite a lot of evidence that behavior composes — you can, for example, drink a milkshake while visiting an ATM — but we don't yet know what the pieces are, how the composition works, or how many composites there might be. A second source of difficulty is that we don't know what features expose what is happening. For example, knowing someone is close to an ATM may be enough to tell that they're visiting the ATM. A third difficulty is that the usual reasoning about the relationship between training and test data is untrustworthy. For example, we cannot argue that a pedestrian detector is safe simply because it performs well on a large dataset, because that dataset may well omit important, but rare, phenomena (for example, people mounting bicycles). You can't run someone over because he does something unusual. The big research question is to link observations of the body and the objects nearby to the goals and intentions of the moving people.

1.7 NOTES

Notes: *much literature on 3D from multiple cameras; on methods and ambiguities in 3D from 2D, particularly using particle filters; we prefer the data association issue; interest points seem good at activity if we can't segment the body;*

There has been extensive experimental work on comparing features for pedestrian detection, and the original Dalal and Triggs paper compares HOG descriptors with the original method of Papageorgiou and Poggio [74]; with an extended version of the Haar wavelets of Mohan *et al.* [62]; with the PCA-Sift of Ke and Sukthankar ([49]; see also [60]); and with the shape contexts of Belongie *et al.* [6], and also is a mine of detailed information on tuning of features.

Index

Bibliography

[1] A. Agarwal and B. Triggs. Monocular human motion capture with a mixture of regressors. In *Workshop on Vision for Human Computer Interaction at CVPR'05*, 2005.

[2] Ankur Agarwal and Bill Triggs. Learning to track 3d human motion from silhouettes. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 2, New York, NY, USA, 2004. ACM Press.

[3] G.J. Agin and T.O. Binford. Computer description of curved objects. In *Int. Joint Conf. Artificial Intelligence*, pages 629–640, 1973.

[4] G.J. Agin and T.O. Binford. Computer description of curved objects. *IEEE Trans. Computer*, 25(4):439–449, April 1976.

[5] Yaakov Bar-Shalom and Xiao-Rong Li. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[6] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE T. Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[7] V.E. Beneš. Exact finite-dimensional filters with certain diffusion non linear drift. *Stochastics*, 5:65–92, 1981.

[8] T.O. Binford. Inferring surfaces from images. *Artificial Intelligence*, 17(1-3):205–244, August 1981.

[9] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[10] B. Bodenheimer, C. Rose, S. Rosenthal, and J. Pella. The process of motion capture: Dealing with the data. In *Computer Animation and Simulation '97. Proceedings of the Eurographics Workshop*, 1997.

[11] M. Brand. Coupled hidden markov models for complex action recognition. Media lab vision and modelling tr-407, MIT, 1997.

[12] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 8–15, 1998.

[13] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *Int. J. Computer Vision*, 56(3):179–194, February 2004.

[14] Louis Brown. *A Radar History of World War II: Technical and Military Imperatives*. Institute of Physics Press, 2000.

[15] Robert Buderi. *The Invention that Changed the World*. Touchstone Press, 1998. reprint.

[16] B. Calais-Germain. *Anatomy of Movement*. Eastland Press, 1993.

[17] Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM Trans. Graph.*, 22(3):569–577, 2003.

[18] Kong-Man (German) Cheung, Simon Baker, and Takeo Kanade. Shape-from-silhouette across time part i: Theory and algorithms. *Int. J. Comput. Vision*, 62(3):221–247, 2005.

[19] Kong-Man (German) Cheung, Simon Baker, and Takeo Kanade. Shape-from-silhouette across time part ii: Applications to human modeling and markerless motion tracking. *Int. J. Comput. Vision*, 63(3):225–245, 2005.

[20] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. von Seelen. Walking pedestrian recognition. *Intelligent Transportation Systems*, 1(3):155–163, September 2000.

[21] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages I: 886–893, 2005.

[22] F.E. Daum. Beyond kalman filters: practical design of nonlinear filters. In *Proc. SPIE*, volume 2561, pages 252–262, 1995.

[23] F.E. Daum. Exact finite dimensional nonlinear filters. *IEEE. Trans. Automatic Control*, 31:616–622, 1995.

[24] Quentin Delamarre and Olivier Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 716, Washington, DC, USA, 1999. IEEE Computer Society.

[25] Quentin Delamarre and Olivier Faugeras. 3d articulated models and multiview tracking with physical forces. *Comput. Vis. Image Underst.*, 81(3):328–357, 2001.

[26] J. Deutscher, A. Blake, and I.D. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages II: 126–133, 2000.

[27] J. Deutscher, B. North, B. Bascle, and A. Blake. Tracking through singularities and discontinuities by random sampling. In *Int. Conf. on Computer Vision*, pages 1144–1149, 1999.

[28] J. Deutscher and I.D. Reid. Articulated body motion capture by stochastic search. *Int. J. Computer Vision*, 61(2):185–205, February 2005.

[29] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[30] Tom Drummond and Roberto Cipolla. Real-time tracking of complex structures with on-line camera calibration. In Tony P. Pridmore and Dave Elliman, editors, *Proceedings of the British Machine Vision Conference 1999, BMVC 1999, Nottingham, 13-16 September 1999*, 1999.

[31] Tom Drummond and Roberto Cipolla. Real-time tracking of multiple articulated structures in multiple views. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 20–36, London, UK, 2000. Springer-Verlag.

[32] Tom Drummond and Roberto Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *ICCV*, pages 315–320, 2001.

[33] T.W. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE T. Pattern Analysis and Machine Intelligence*, 24(7):932–946, July 2002.

[34] A.E. Engin and S.T. Tumer. Three-dimensional kinematic modelling of the human shoulder complex - part i: Physical model and determination of joint sinus cones. *ASME Journal of Biomechanical Engineering*, 111:107–112, 1989.

[35] A. Farina, D. Benvenuti, and B. Ristic. A comparative study of the benes filtering problem. *Signal Processing*, 82:133–147, 2002.

[36] Xiaolin Feng and P. Perona. Human action recognition by sequence of movelet codewords. In *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pages 717–721, 2002.

[37] J. Gibson. *The perception of the visual world*. Houghton-Mifflin, 1950.

[38] Michael Gleicher. Animation from observation: Motion capture and motion editing. *SIGGRAPH Comput. Graph.*, 33(4):51–54, 2000.

[39] K. Grauman, G. Shakhnarovich, and T.J. Darrell. Virtual visual hulls: Example-based 3d shape inference from silhouettes. In *SMVP04*, pages 26–37, 2004.

[40] R. Gross, I. Matthews, and S. Baker. Appearance-based face recognition and light-fields. *IEEE T. Pattern Analysis and Machine Intelligence*, 26(4):449– 465, 2004.

[41] Adrian Hilton and Jonathan Starck. Multiple view reconstruction of people. In *2nd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2004), 6-9 September 2004, Thessaloniki, Greece*, pages 357–364, 2004.

[42] G.E. Hinton. Relaxation and its role in vision. Technical report, University of Edinburgh, 1978. PhD Thesis.

[43] P. Hong, M. Turk, and T.S. Huang. Gesture modeling and recognition using finite state machines. In *AFGR00*, pages 410–415, 2000.

[44] S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *CVIU*, 96(2):129–162, November 2004.

[45] N.R. Howe. Silhouette lookup for automatic pose tracking. In *IEEE Workshop on Articulated and Non-Rigid Motion*, page 15, 2004.

[46] F. Jelinek. *Statistical Methods for Speech Recognition (Language, Speech and Communication)*. MIT Press, 1999.

[47] R. V. Jones. *Most Secret War*. Wordsworth Military Library, 1998. reprint.

[48] Shanon X. Ju, Michael J. Black, and Yaser Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proc. Int. Conference on Face and Gesture*, pages 561–567, 1996.

[49] Y. Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages II: 506–513, 2004.

[50] Roland Kehl, Matthieu Bray, and Luc Van Gool. Full body tracking from multiple views using stochastic sampling. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 129–136, Washington, DC, USA, 2005. IEEE Computer Society.

[51] A.G. Kirk, J.F. O'Brien, and D.A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[52] M.W. Lee and I. Cohen. Human upper body pose estimation in static images. In *European Conference on Computer Vision*, pages Vol II: 126–138, 2004.

[53] M.W. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages II: 334–341, 2004.

[54] M.W. Lee and R. Nevatia. Dynamic human pose estimation using markov chain monte carlo approach. In *IEEE Workshop on Motion and Video Computing*, pages 168–175, 2005.

[55] J.S. Liu and R. Chen. Sequential monte-carlo methods for dynamic systems. Technical report, Stanford University, 1999. preprint.

[56] Matt Liverman. *The Animator's Motion Capture Guide : Organizing, Managing,Editing*. Charles River Media, 2004.

[57] C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[58] D. Marr and H.K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. Roy. Soc. B*, 200:269–294, 1978.

[59] Alberto Menache. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan-Kaufmann, 1999.

[60] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE T. Pattern Analysis and Machine Intelligence*, 2004. accepted.

[61] T.B. Moeslund. Summaries of 107 computer vision-based human motion capture papers. Technical Report LLA 99-01, University of Aalborg, 1999.

[62] A. Mohan, C.P. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE T. Pattern Analysis and Machine Intelligence*, 23(4):349–361, April 2001.

[63] A. Mohr and M. Gleicher. Building efficient, accurate character skins from examples. *ACM TOG*, 22(3):562–568, 2003.

[64] G. Mori, , and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision LNCS 2352*, volume 3, pages 666–680, 2002.

[65] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005. to appear.

[66] T. Mori, Y. Segawa, M. Shimosaka, and T. Sato. Hierarchical recognition of daily human actions based on continuous hidden markov models. In *Automatic Face and Gesture Recognition*, pages 779–784, 2004.

[67] S.A. Niyogi and E.H. Adelson. Analyzing gait with spatiotemporal surfaces. In *Proc. IEEE Workshop on Nonrigid and Articulated Motion*, pages 64– 69, 1994.

[68] S.A. Niyogi and E.H. Adelson. Analyzing and recognizing walking figures in xyt. Media lab vision and modelling tr-223, MIT, 1995.

[69] N. Oliver, A. Garg, and E. Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *CVIU*, 96(2):163–180, November 2004.

[70] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 3–8, 2002.

[71] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE T. Pattern Analysis and Machine Intelligence*, 2:522–546, 1980.

[72] C.J. Pai, H.R. Tyan, Y.M. Liang, H.Y.M. Liao, and S.W. Chen. Pedestrian detection and tracking at crossroads. In *IEEE Int. Conf. Image Processing*, pages II: 101–104, 2003.

[73] C.J. Pai, H.R. Tyan, Y.M. Liang, H.Y.M. Liao, and S.W. Chen. Pedestrian detection and tracking at crossroads. *Pattern Recognition*, 37(5):1025–1034, May 2004.

[74] C. Papageorgiou and T. Poggio. A trainable system for object detection. *Int. J. Computer Vision*, 38(1):15–33, June 2000.

[75] C.P. Papageorgiou and T. Poggio. A pattern classification approach to dynamical object detection. In *Int. Conf. on Computer Vision*, pages 1223–1228, 1999.

[76] Ralf Plänkers and Pascal Fua. Tracking and modeling people in video sequences. *Comput. Vis. Image Underst.*, 81(3):285–302, 2001.

[77] R. Polana and R. Nelson. Recognizing activities. In *Proceedings IAPR International Conference on Pattern Recognition*, pages A:815–818, 1994.

[78] L. Rabiner and B-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[79] D. Ramanan. *Tracking People and Recognizing their Activities*. PhD thesis, U.C. Berkeley, 2005.

[80] D. Ramanan and D.A. Forsyth. Automatic annotation of everyday movements. In *Advances in Neural Information Processing*, 2003.

[81] D. Ramanan and D.A. Forsyth. Finding and tracking people from the bottom up. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages II: 467–474, 2003.

[82] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[83] K. Rohr. Incremental recognition of pedestrians from image sequences. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 9–13, 1993.

[84] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94–115, 1994.

[85] G.C. Schmidt. Designing nonlinear filters based on Daum's theory. *Journal of Guidance, Control and Dynamics*, 16:371–376, 1993.

[86] A. Shahrokni, T. Drummond, and P. Fua. Fast Texture-Based Tracking and Delineation Using Texture Entropy. In *International Conference on Computer Vision*, 2005.

[87] A. Shahrokni, T. Drummond, V. Lepetit, and P. Fua. Markov-based Silhouette Extraction for Three–Dimensional Body Tracking in Presence of Cluttered Background. In *British Machine Vision Conference*, Kingston, UK, 2004.

[88] A. Shahrokni, F. Fleuret, and P. Fua. Classifier-based Contour Tracking for Rigid and Deformable Objects. In *British Machine Vision Conference*, Oxford, UK, 2005.

[89] G. Shakhnarovich, P. Viola, and T.J. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Int. Conf. on Computer Vision*, pages 750–757, 2003.

[90] L. Sigal, S. Bhatia, S. Roth, M.J. Black, and M. Isard. Tracking loose-limbed people. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages I: 421–428, 2004.

[91] Marius-Calin Silaghi, Ralf Plänkers, Ronan Boulic, Pascal Fua, and Daniel Thalmann. Local and global skeleton fitting techniques for optical motion capture. In *Modelling and Motion Capture Techniques for Virtual Environments*, pages 26–40, November 1998. Proceedings of CAPTECH '98.

[92] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages I:447–454, 2001.

[93] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *The International Journal of Robotics Research*, 22(6):371–391, 2003.

[94] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3d human tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages I: 69–76, 2003.

[95] J. Starck and A. Hilton. Virtual view synthesis of people from multiple view video sequences. *Graphical Models*, 67(6):600–620, 2005.

[96] Jonathan Starck and Adrian Hilton. Model-based multiple view reconstruction of people. In *Int. Conf. on Computer Vision*, pages 915–922, 2003.

[97] Jonathan Starck and Adrian Hilton. Spherical matching for temporal correspondence of non-rigid surfaces. In *Int. Conf. on Computer Vision*, 2005.

[98] Jonathan Starck, Adrian Hilton, and John Illingworth. Human shape estimation in a multi-camera studio. In *BMVC*, 2001.

[99] C.J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 677–84, 2000.

[100] Christian Theobalt, Joel Carranza, Marcus A. Magnor, and Hans-Peter Seidel. Enhancing silhouette-based human motion capture with 3d motion fields. In *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 185, Washington, DC, USA, 2003. IEEE Computer Society.

[101] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Int. Conf. on Computer Vision*, pages II: 50–57, 2001.

[102] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. J. Computer Vision*, 48(1):9–19, June 2002.

[103] S.T. Tumer and A.E. Engin. Three-dimensional kinematic modelling of the human shoulder complex - part ii: Mathematical modelling and solution via optimization. *ASME Journal of Biomechanical Engineering*, 111:113–121, 1989.

[104] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Int. Conf. on Computer Vision*, pages 734–741, 2003.

[105] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *Int. J. Computer Vision*, 63(2):153–161, July 2005.

[106] A.W. Wilson and A.F. Bobick. Learning visual behavior for gesture analysis. In *IEEE Symposium on Computer Vision*, pages 229–234, 1995.

[107] A. Gelb with Staff of the Analytical Sciences Corporation. *Applied Optimal Estimation*. MIT Press, 1974.

[108] M. Yamamoto, A. Sato, S. Kawada, T. Kondo, and Y. Osaki. Incremental tracking of human actions from multiple views. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 2, Washington, DC, USA, 1998. IEEE Computer Society.

[109] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[110] J. Yang, Y. Xu, and C. S. Chen. Human action learning via hidden markov model. *IEEE Transactions on Systems Man and Cybernetics*, 27:34–44, 1997.

[111] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *PAMI*, 26(9):1208–1221, September 2004.