M.I.T Media Laboratory Perceptual Computing Section Technical Report No. ?
Vision and Modeling group

A Categorical Indexing Framework for Human Movements

James W. Davis
MIT Media Laboratory, 20 Ames St., Cambridge, MA 02139
(jdavis@media.mit.edu)

Abstract

We present a framework for representing human movements such as would be found in the performance of a wide variety of exercise movements involving multiple limb motions. The framework provides the basis for a hierarchical index to categorize movements. The approach does not optimize an indexing method based on a sample of "training data", but rather is based on patterns that appear in the sense data due to the underlying generating processes and context-sensitive feature/property detectors. In this Gibsonian-style method, the patterns can be organized to create a hierarchical classification for indexing and recognition.

1 Introduction

Given a set of observations, how do we decide which model best explains the observed facts? Two problems which arise in answering this question include how we find the model within a large set of targets, and also what our criteria is for a best match. Exhaustive search is a non-compelling means of arriving at a conclusion, and can fall into complexity and timing issues when the possible target set is large (especially if the model representation is non-trivial). Here, we seek to create a sensible hypothesis-and-test paradigm which limits (or orders) the available targets for the recognition task. Feasible targets are those which could have produced (generated) the observed data. Optimality for the search and match will be defined as capturing the *modal* (or *categorical*) nature of the generating process.

We present our framework within the domain of human movement recognition, where the goal is to identify simple body motions (or gestures). Significant emphasis has been, and continues to be, placed on this area of research [1, 3, 20, 4, 6, 7, 11, 12, 13, 17, 19]. Recently, we presented an approach which shows promising real-time results for a wide variety of human movements using patterns of motion as the underlying representation [4]. We begin our discussion by presenting an extension to this approach, which helps remove inherent problems within that method. We then explore the categorical indexing framework for that enhanced representation.

Our indexing approach rests on the idea that categorization of visual movement scenes can be achieved by examining a hierarchical relationship between key features of the motion patterns. We call this hierarchical organization a "structure lattice". Originally, structure lattices have been used for *interpretating* or *explaining* visual scenes by placing a context-sensitive ordering on the specializations of key features [8, 9, 15, 14, 18]. Here we take a slightly different view, and believe that these lattices (along with preference orderings) are also useful for model-indexing by pruning down the possible set of recognition targets in a compelling way.

The remainder of this paper is organized as follows. First, we briefly examine a real-time approach for recognizing simple body movements (Section 2.1). We then follow with a reformulation and extension of this work (Section 2.2). Next, an indexing framework based on an ordering of a structure lattice for the data is presented (Section 3). Finally, we conclude with a summary of the approach (Section 5).

2 Supporting Research

2.1 Temporal Templates

In previous work [4], we presented a real-time approach for representing and recognizing simple human movements. The motivation for the approach was how easily people can recognize common human movements (like sitting) from low-resolution imagery where the image features are basically not perceivable. The approach relies on "patterns of motion" as the representation for identifying various human movements. The basis of the representation is a *Temporal Template* — an image where the value at each point is a function of the motion properties at the corresponding spatial location in an image sequence. A recognition method was also developed in this work which matches these Temporal Templates against stored instances of known movements. The method automatically performs temporal segmentation, is invariant to linear changes in speed, and runs in real-time on a standard computer platform.

The goal of the approach is to construct a view-specific static representation of movement which encodes the act of motion over time. Consider the sitting sequence as shown in Figure 1. The top row shows someone in the act of sitting, as viewed from the side. The template representation attempts to encode "how" the motion is moving through the sequence. To achieve this, a simple replacement and decay operator is used to layer binary motion images [1] and "weight" the motion images by recency of time (i.e.

[1] We define binarized motion as either "motion" or "no-motion", and is found by thresholding dense optical flow. One could also use binary image-differencing to achieve similar results.

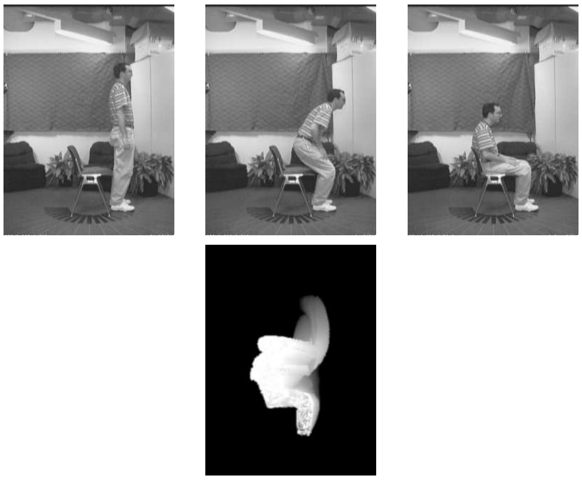

Figure 1: Someone in the act of sitting as viewed from the side. Also shown is the Temporal Template for the movement.

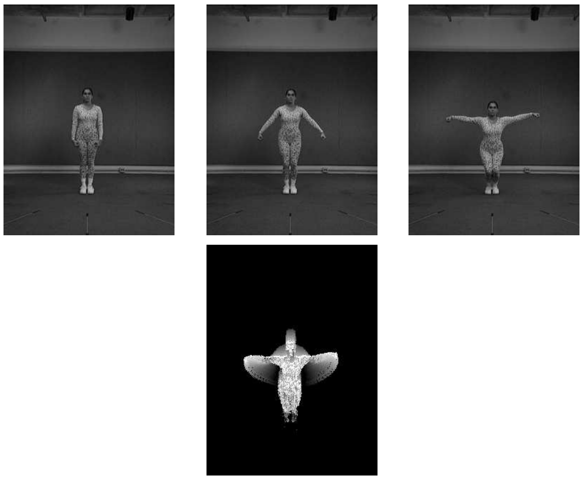

Figure 2: A person squatting down while extending and raising her arms. The resulting Temporal Template is also shown.

pixels that more recently moved are brighter in the image). The resulting image template at the end of the sitting movement is also shown in Figure 1. Note that there is a collapsing of the space-time volume into a single form, but that the representation still perceptually captures the essence of the movement and its temporal structure. For recognition, seven higher-order moments are used as shape descriptors and energy localizers for the image. These seven descriptors are then statistically matched to stored examples of different movements.

Though this simple approach has shown promising results with minimal computational power, there are, of course, problems which can arise. First, the representation is calculated by performing a global motion accumulation. Therefore, any spurious motion gets added into the final template. Second, the recognition method computes global shape/energy features, so any sizeable patches of motion missing (or added) can greatly effect the resulting characterization. And lastly, no method of reducing the possible target set of models has been offered, only exhaustive search has been used. The approach below offers a means of directly addressing these problems.

2.2 Reformulation and Extension

In short, the major problem with the Temporal Template method is that it is a "global" approach. It blindly adds regions to an accumulating mass (or does not account for missing regions) and then examines the region as a whole for discrimination. To help combat this type of sensitivity, we explore the role of local region operations. If particular patches of coherent motion can be identified, we can use the information about the individual patches and their relation to other patches to help combat the inclusion/exclusion of problematic motion regions. Most importantly, and the topic of this paper, is that these regions can offer a compelling method for indexing into a set of target movements (unlike before), by cataloging the possible specialized relations between properties of the motion pieces.

Consider the movement sequence shown in Figure 2 of a person squatting while extending the arms out to the side. The standard Temporal Template for this movement is also shown in Figure 2. It is easy to see that if a portion of the arm span in the template were removed (by occlusion), then the global feature characterization would be flawed, causing catastrophic recognition results.

We now focus on the use of the motion regions as a means for model-indexing. The motion blobs individually are not necessarily discriminatory, for their shape and other properties are not guaranteed to be salient features when compared to a large set of motion pieces. Also, there would need to be information retained on the spatial relation between the blobs (e.g. left-of, above, etc.) to be able to characterize the total signal. To us, it feels that the strongest way to evaluate the motion signal is to characterize the relations both *inside* and *between* the motion regions. We can then place an ordering on the relations *between* blobs to determine which properties can drive the model-indexing for recognition to a more limited target group. The ordering should be general enough to extend to multiple instances which can occur within a given context (e.g. human motion), rather than an optimized solution for a strict target set of known motions (thus requiring a re-optimization when a new exemplar is added). Once a possible target has been hypothesized, the *internal* region information can be used to verify a target. We now take a closer look into the generation, ordering, and indexing capabilities of such an approach for identifying a set of plausible targets.

3 Cataloging Movements

In this section, we present an approach for characterizing the relations *between* the motion blobs making up the motion signal, to reflect the nature of the underlying generating process of the motion (i.e. a moving human). We use context-specific concepts to create a state-space of possible relations between motion blobs. An ordering is then placed on the state-space to reflect common underlying properties associated with our domain for human movements. We believe the resulting ordering is well motivated and has certain credibility as a method for indexing.

3.1 Structure Lattice

As defined in [18], a structure lattice is a vehicle that takes context-sensitive, primitive concepts about structural regularities, and composes them to produce a set of possible configuration states. The lattice displays a partial ordering of the categorical states, where the ordering is derived by noticing that some states are special or limiting cases of others. The lattice shows the specialization relations (i.e. non-accidental features [10, 14]) between the categories of structures in the *world* that will appear in the representation given the observable concept properties. At the top of the partial ordering is a state of random generalization between the objects. This is the least structured node. Then at each level below, a dimension of specialization is incrementally imposed until reaching the bottom node(s) in the ordering. The bottom node(s) is completely specified with regularities (most structured). Structure lattices have been previously used for *interpretating* or *explaining* visual scenes [8, 9, 15, 14, 18]. Here, we take a different view, and believe that the lattices are also useful for model-indexing.

Let's begin by defining our context-sensitive concepts. The working context for this research is blobs of motion that are generated by human beings performing simple exercise movements (e.g. jumping jacks). We define relations *between* blobs in a pairwise fashion to reduce complexity (not unreasonable, as mentioned in [16] and also due to the symmetric nature of humans), and therefore develop concepts that work on relations between only two blobs. We later justify using the pair-wise structure lattice as sufficient for indexing. As for our concepts, we have chosen binary properties[2] of Form (F), RandomFate (R), CommonFate (C), OppositeFate (O), and VerticlePlacement (V). The concept of Form (See Figure 3) examines whether two blobs are about the same size and have similar elongation (this is a rough shape estimation). To account for the motion flow, we have created three concepts. First, CommonFate happens when the two blobs exhibit similar flow within a given range (a few examples are shown in Figure 4). Opposite fate describes the flow when the two blobs have opposing or opposite directional characteristics within some angular limits (two examples are shown in Figure 5). Most general is RandomFate, where this accounts for any motions present between the two blobs (See Figure 6). The last concept is VerticalPlacement (See Figure 7),

[2] One could consider using probabilities.

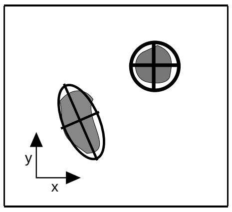

Figure 3: A bounding ellipse can act as a rough measure of form to check for similar sizes and shapes between two blobs.

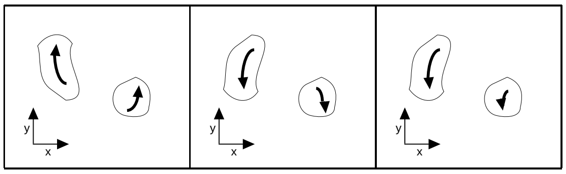

Figure 4: Three styles of common fate motion for two blobs.

where the mean location of each blob is checked to see if they both reside at the same (or similar) vertical alignment (the horizontal counterpart is also a possible concept). Notice that all of these concepts are invariant to size, therefore we can scale the imagery (or have different sized people) and the resulting properties are unaffected. We justify these concepts by noticing that human movements can contain a great deal of symmetry (due to physical body symmetries), and that many movements in our context involve multiple limb motions working together. Therefore, the form, motion similarity, and vertical placements basically encode many arm/leg movement combinations where symmetries and similarities play an important role.

Simple methods can be used for deriving these concepts. As for Form, we can measure the elongation of a blob by examining the ratio of the axis lengths of a bounding ellipse, and also get a measure of the size by looking at the maximum axis length. Together, these two measurements can be used to tell if the two blobs have similar shape. The VerticalPlacement is simply the location of the mean for the motion blob. For the motion concepts, a rough measure of the motion for a region can be generated by creating a histogram of the quantized motion flow directions within the region.

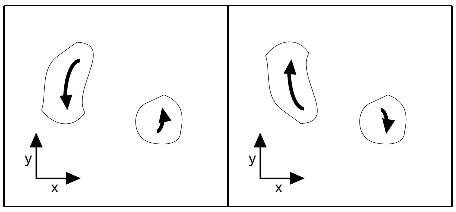

Figure 5: Two styles of opposite fate motion for two blobs.

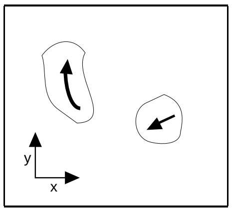

Figure 6: Random motions between two blobs.

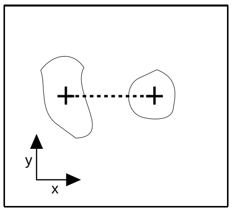

Figure 7: Two blobs can found to reside around the same vertical placement by examining their mean locations.

This is reasonable because a "patch" of motion should be smooth and mostly homogeneous due to the simple human motions (and segmentation algorithm). Since the motion comparisons for fate are mostly symmetric validations between blobs, then flipping, shifting, or mirroring the histogram can account for the above common and opposite fate descriptions.

In generating the structure lattice, we begin with a top-level node, which has the most degrees of freedom and is completely unspecified (See Figure 8). Therefore we have the state $\bar{F}R\bar{V}$, which is defined to explain two blobs which *do not have* similar form ($\bar{F}$ = not-Form), motion similarity ($R = \bar{C}\bar{O}$ = random motions), and vertical placement ($\bar{V}$ = not at same vertical level). For the next level of the lattice, we constrain one concept at a time to a "specialized" instance (in our case, specialized means having a property X rather than $\bar{X}$) and generate a new node. Since the RandomFate concept is made specific by generating either a CommonFate or OppositeFate property, we thus generate four nodes from the original top node (See Figure 8). The leftmost node constrains the Form, the next two nodes, constrain the Fate (where either CommonFate or OppositeFate are true, but not both), and the last node constrains the VerticalPlacement. The next row of nodes in the lattice combines two constraints from the previous row to generate five new nodes. From a combinatoric standpoint, there are six *possible* nodes given the concepts, but the combination of $\bar{F}CO\bar{V}$ is not consistent because the motions can not simultaneously be "common" **and** "opposite". The following row combines three constraints, and as expected certain combinations are not permitted ($\bar{F}COV$ and $FCO\bar{V}$) for the same reasons given above, leaving only $F\bar{C}OV$ and $FC\bar{O}V$. We note that if two blobs are generated by the two arms fanning up or down (yielding $FC\bar{O}V$),

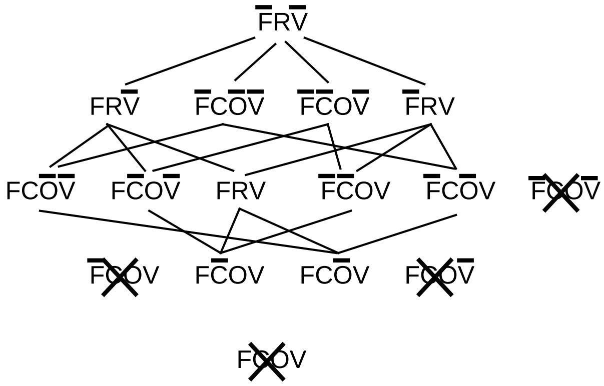

Figure 8: The resulting structure lattice for the generating concepts. The inconsistent states are crossed out.

then there is a spatial-temporal coincidence of the two blobs at the line of vertical symmetry by extrapolating the blob motions. Since we cannot conjoin both C and O, there is no bottom terminal node of $FCOV$. We therefore end up with the lattice shown in Figure 8, with the inconsistent nodes crossed out. The resulting number of nodes is 12 rather than the full 16. This generation of plausible states and identification of non-possible states is yet another argument for using a structure lattice. If one were to simply use a polling mechanism of properties, they would have no grounded means of determining inconsistencies or possible failures of routines in relation to the properties. The lattice mechanism makes explicit the world (or representation) possibilities and inconsistencies (such as the combination CO being impossible).

In the above example, the source for inconsistencies were from a specialization of the RandomFate motions into either CommonFate or OppositeFate, but not both. This exclusiveness, one may argue, could easily be known and accounted for because they both derive from a more general property (RandomFate). But consider the effect of a new concept of Coincidence (touching) paired with CommonFate. If the method that segments the motion blobs acts on the homogeneity of connected (touching) motion regions, then these two properties would never exist together at the same time. This inconsistency is a direct result of the segmentation algorithm, rather than the specialization problem discussed above. Such inconsistencies may not be directly known unless found by observation or by generating the structure lattice which make those relations explicit (and thus brought to attention). Just using a collection of properties/concepts together with a polling (or property counting) method does not capture the underlying constraints and thus might be problematic during later processing.

We note that we could have chosen concepts other than the ones above (e.g. Coincidence), and thus a different lattice would have resulted. In addition to adding more spatial concepts, we could also include stronger temporal constraints between the blobs. Currently, two blobs could have common form, fate, and vertical placement, but may have different temporal durations. The current set of concepts do not well represent this temporal discrepancy. We therefore could include additional concepts of temporal relations (e.g. before, after, during, equal, and others as described by Allen [2]) between the blobs to increase the descriptive power.

Though generating the lattice is currently a manual task[3] and also may be taxing when the number of concepts becomes large, or more complex, we feel this method of generating plausible states is compelling and useful. We have shown a method for generating the state-space configuration, and now we discuss ordering these states by preference, which then directly effects the indexing methodology.

3.2 Preference Ordering

The structure lattice, as described in the previous section, simply presents all the categories available in the chosen context. These states are ordered only with respect to the increasing specialization of the structure. We can expect certain regularities to appear in one context (e.g. exercise movements), while in another context (e.g. discourse gestures) the regularities may be different. This suggests that given a context, there is a preference ordering on the expected regularities [18]. A preference ordering therefore specifies which kinds of concepts are preferred to others. We note here that much of the contextual information has been embedded into the generating concepts, so the context setting has been well-forged into the concepts themselves.

For our interests in model-indexing (with simple exercise movements), we choose to prefer those states which **have** the particular properties common in the world[4]. This means that the more preferred states for us are the ones with the most concepts (i.e. prefer blobs that have the same shape, motion, and vertical placement). In other words, if we see two blobs that have the same form, motion, and vertical placement, then this situation is considered special (or more special than two blobs that have none, or only a few, of those properties). To summarize the preferences for our concepts, we have $F \succ \bar{F}$, $V \succ \bar{V}$, $C \succ \bar{C}$, $O \succ \bar{O}$, and $\{C, O\} \succ R$, where $\succ$ means "is preferred to". Within our contextual domain, we don't have any reason to prefer motion C over O (or visa versa), therefore we can say that C is "equally-preferred" to O, or $C \sim O$. By ordering the state space from Figure 8 with these preferences, we see that a preference graph can be formed (See Figure 9) where the top-most states are most preferred and the same-level states are set to be equally preferred. In this case, the preference graph is an inverted version of the structure lattice (from Figure 8). We can also derive a *full* preference order where preferences *between* properties are specified (unlike the above partial ordering). Such a full ordering may give higher priority to form over position

[3] Feldman [5] explores conditions that allow lattices to be built automatically.

[4] The context sensitive priors are typically used as means of selecting the preferences.

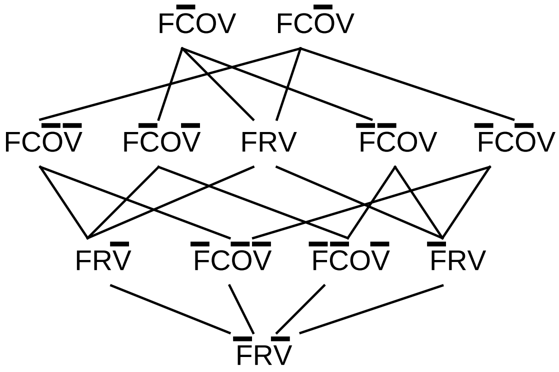

Figure 9: A preference order imposed on the structure lattice yields a new graph with the top-most nodes being most preferred, and where the same-level states are set to be equally preferred. This graph is an inverted version of Figure 8.

$(F \succ V)$, or motion over form $(C, O \succ F)$.

It is possible to have several maximally preferred nodes where it remains undecided to which state should be the most preferred, as in our case with $F\bar{C}OV$ and $FC\bar{O}V$. In a visual interpretation domain (as in explaining the Necker cube illusion[5]), we are left with multiple maximal nodes, giving many possible interpretations. From a perception standpoint, a single solution is much more desired, and there are ways to push for a single node (See [9, 18]). But for our goal of indexing, we feel this is not necessary.

We can collapse (horizontally) the preference graph into a list of state sets for each level to generally represent the ordering as a "preference-listing":

$$\begin{gathered}\{F\bar{C}OV, FC\bar{O}V\}\\ \{FC\bar{O}\bar{V}, F\bar{C}O\bar{V}, FRV, \bar{F}\bar{C}OV, \bar{F}C\bar{O}V\}\\ \{FR\bar{V}, \bar{F}\bar{C}O\bar{V}, \bar{F}C\bar{O}\bar{V}, \bar{F}RV\}\\ \{\bar{F}R\bar{V}\}\end{gathered}$$

with the top set as being most preferred, and the bottom set being least preferred.

At this point, we would like to bring attention back to the structure lattice. Consider a lower portion of some lattice that has terminal nodes residing on different levels (See Figure 10). We may want that terminal node which is **not** on the bottom level to be included in the top level of preference-listing along with the other terminal nodes. We say this because such a node that has no other possible specializations (no nodes below) should be considered an extreme case. Therefore, we could include this node with the other maximally preferred nodes without the conceptual loss.

We now describe the use of the above preference-listing as a means of hierarchical model-indexing. This listing basically acts as a hierarchy of preferences from which we can then select possible target models based on contextual regularities.

[5] See a similar problem in [9].

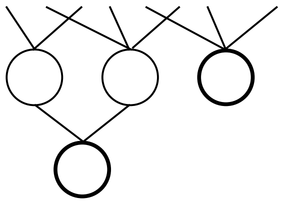

Figure 10: A particular case in a structure lattice where a terminal node is not on the bottom level of the graph.

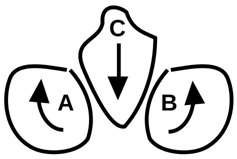

Figure 11: Motion blobs (abstracted to line drawings) for the movement from Figure 2.

4 Hierachical Model-Indexing

With the preference-listing derived in the previous section, we now have a grounded structure for finding "similar" models (derived from the Temporal Templates). By similar we loosely mean that some motion blobs in one Temporal Template have the same properties of some blobs in another Temporal Template. For a simple example, let's consider the motion patterns for the movement template from Figure 2 with three motion blobs (abstracted to line drawings), as shown in Figure 11. For this set of motion regions, we can make a list of the pairwise blob combinations and then rank and collect them with respect to the preference-listing described above. Though there are $\frac{N(N-1)}{2}$ possible pairs from N blobs, we believe that the number of total blobs in any template of a simple movement should be small in the absence of extreme noise, so the combinatorics is not a problem. The preference-listing imposed on this set of blob combinations results in the following indexing hierarchy:

$$\begin{array}{c}\{(A,B)\}\\ \{NONE\}\\ \{NONE\}\\ \{(A,C),(B,C)\}\end{array}$$

where A, B, and C are labels for the particular blob regions, and each level corresponds to a level in the preference-listing. Here we see that the preference ranking places the blobs A and B (from the arm motions) together because of the high specialization (with common form, motion, and vertical placement). The remaining combinations do not tell us as much and are placed low (at the bottom in this case) in the hierarchy. Notice that there are no mid-level mappings in this example. This is the hierarchical form which is used for categorical indexing of the movement blobs.

The above indexing hierarchy for that set of blobs is used to find models in the target library by first searching for those models which also contain a top-level (most specialized) entry. Then, from those models selected, we further eliminate more model choices by searching for the second-level entries, and so forth, until we reach the bottom-level (least specialized). This method reduces the possible set of targets by initially using the high degree of specialization at the upper levels in the preference-listing hierarchy. If all models (movements) had this high degree of specialization, then this method would not prune so quickly. But we believe this is not the case when dealing with movements in our domain.

Nowhere yet do we assert the correspondence between the blob pairs *through* the levels while indexing. So far, we only looked for models which have blobs that have those properties found in the testing preference-listing. One movement may have a set (A, B) at one level and (A, C) at another level, where the two sets are related/linked by A. When looking for possible targets, we need to make sure this constraint through the levels is retained. For example, we want to exclude the target model made up of sets (X, Y) and (W, Z), where (X, Y) is at the same level as (A, B) and (W, Z) is at the same level as (A, C), because there is no link between (X, Y) and (W, Z) through X (or Y). In the input, the two sets (A, B) and (A, C) are related (or linked) through A and thus contain three blobs total. The target with (X, Y) and (W, Z) has no such relation, and is actually made up of four blobs, instead of three. So once the target sets are found, there needs to be a correspondence mapping to ensure the relations between blob sets are correct. We can use simple substitution rules and graph matching techniques to find whether a target model is truly a plausible match by configuration. Once a possible match is found, it needs to be verified by taking a closer look at the data. Remember that the concepts used in generating the structure lattice and preference ordering are binary relation properties. These properties merely say that two blobs may look similar. Nowhere does it retain information on the shape, the motion property (the motion cases contain a few supporting types), or the placement/distance of the blobs (e.g. "a bit above and to the left of the blob"). This can easily be resolved by retaining the actual information for each blob used for the binary concept relations. Therefore the individual blob information (shape, motion, vertical placement) can be retained throughout the indexing process, and then can be used for determining the match by examining the actual internal information.

4.1 Occlusions

There are two types of occlusions that this framework can tolerate. First, there are occlusions that *totally* occlude motion pieces (Type-I occlusion). Here we have no way of knowing what really happened behind the occluder, but we can perform an indexed match with the remaining visible motion regions and

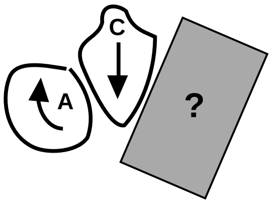

Figure 12: Type-I occlusion on the movement blobs from Figure 11.

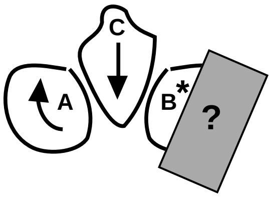

Figure 13: Type-II occlusion on the movement blobs from Figure 11.

check for this data to be a subset of the target model. Knowing the size and position of the occluder can better enable a match because if the input is a subset of some target, we can check to see if the remaining parts of the target set would in fact reside behind the current occluder. Figure 12 shows a Type-I occlusion on the motion patterns from Figure 11. The following single state preference-listing for the input would be used to find possible targets:

$$\begin{array}{c}\{NONE\}\\ \{NONE\}\\ \{NONE\}\\ \{(A,C)\}\end{array}$$

This single set (A, C) would initially register with all targets containing two motion regions also residing in the bottom preference-listing hierarchy. These possible targets are then further checked for compliance using the internal form, motion, and placement specifics. At this point the model from Figure 11 would remain (and possibly others). The model has an additional region, B, which would in fact be located behind the occluder in the test input. We can therefore offer a plausible match for the occluded input movement due to the prior knowledge of the occluder and because the missing region *could* reside behind the occlusion.

The second type of occlusion is when only *part* of the signal is masked (Type-II occlusion). In this situation, we can tag the motion regions that are coincident with the occluder. Then we can match all non-tagged pairs (as in the Type-I case). The compliance check with the target model follows, where it may be possible that the partially occluded region can give some indication of a possible region match (or reject) from its available motion. The occluder's size and position are also used, as in the Type-I example, to offer an explanation of the missing motion, if possible. Figure 13 shows a Type-II occlusion on the motion patterns shown in Figure 11. The following preference-listing would be used to find possible targets:

$$\begin{array}{c}\{NONE\}\\ \{NONE\}\\ \{NONE\}\\ \{(A,C),(A,B^*),(B^*,C)\}\end{array}$$

The un-tagged set (A, C) would find the model from Figure 11 (and possibly others) as was discussed in the Type-I example. The input here has an additional tagged region B^* which announces the presence of a motion region that most likely continues behind the occluder. Since the target (from Figure 11) has a motion region which would be within the occlusion area and also explain the presence of the partial input, we can therefore offer a stronger match possibility as compared to the Type-I version. This match may be rejected though if there is enough discrepancy in the partial motion information (the form and placement information are not helpful for partial information). For example if the input motion region coincident with the occluder shows downward motion and the model contains purely upward motion for the corresponding region, this opposition is enough to remove this target as a possible match.

The basic advantage here is that we have decomposed the total motion signal into coherent pieces. With this we have a chance at indexing and recognition when occlusions occur because not all the data will be corrupted. Solutions can be offered, even when large regions are occluded. The previous Temporal Template method was extremely fragile to occlusions, and now with this framework, should not be as susceptible to such problems.

5 Summary

In this paper we presented an indexing framework to aid in recognizing human movement. The method exploits the coupling between the physical data-generating process and the model representation. An indexing hierarchy is developed which is capable of extending to a wide variety of simple human movements. This approach does not optimize an indexing method based on large sets of "training data", but rather is based on patterns that appear in the sense data due to the underlying generating process and context-sensitive feature/property detectors. The use of a structure lattice and preference ordering enables a compelling means for creating a sensible method for indexing based on categorization rather than using exhaustive search. The approach also suggests a scheme for handling occlusions, which are generally problematic in most systems.

6 Acknowledgements

So far, Whitman...

References

[1] Akita, K. Image sequence analysis of real world human motion. *Pattern Recognition*, 17, 1984.

[2] Allen, J. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

[3] Campbell, L. and A. Bobick. Recognition of human body motion using phase space constraints. In *Proc. Int. Conf. Comp. Vis.*, 1995.

[4] Davis, J. and A. Bobick. The representation and recognition of human movement using temporal templates. In *Proc. Comp. Vis. and Pattern Rec.*, pages 928–934, June 1997.

[5] Feldman, J. Constructing perceptual categories. In *Proc. Comp. Vis. and Pattern Rec.*, pages 244–250, 1992.

[6] Gavrila, D. and L. Davis. Tracking of humans in actions: a 3-d model-based approach. In *ARPA Image Understanding Workshop*, Feb 1996.

[7] Hogg, D. Model-based vision: a paradigm to see a walking person. *Image and Vision Computing*, 1(1), 1983.

[8] Jepson, A. and W. Richards. What is a percept. MIT Center for Cognitive Science Occasional Paper No. 43, MIT, April 1991.

[9] Jepson, A. and W. Richards. A lattice framework for integrating vision modules. *IEEE Trans. Sys., Man and Cyber.*, 22(5):1087–1096, September/October 1992.

[10] Jepson, A., and W. Richards. *What makes a good feature?*, chapter Spatial vision in humans and robots. Cambridge University Press, 1992.

[11] Johansson, G. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973.

[12] Ju, S., Black, M., and Y. Yacoob. Cardboard people: a parameterized model of articulated image motion. In *Submitted to the Second International Conference on Automatic Face and Gesture Recognition*, 1996.

[13] Little, J., and J. Boyd. Describing motion for recognition. In *International Symposium on Computer Vision*, pages 235–240, November 1995.

[14] Mann, R., and A. Jepson, A. *Machine learning in computer vision*, chapter Non-accidental features in learning. AAAI Press, 1994.

[15] Mann, R., Jepson, A., and J. Siskind. The computational perception of scene dynamics. In *Proc. European Conf. Comp. Vis.*, pages 329–342, April 1996.

[16] Minsky, M. *The society of mind.* Simon and Schuster, Inc., 1985.

[17] Polana, R. and R. Nelson. Low level recognition of human motion. In *IEEE Workshop on Non-rigid and Articulated Motion*, 1994.

[18] Richards, W., Jepson, A., and J. Feldman. *Perception as Baysian inference*, chapter Priors, preferences and categorical percepts. Cambridge University Press, 1996.

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

[20] Wren, C., Azarbayejani, A., Darrell, T., and A. Pentland. Pfinder: Real-time tracking of the human body. In *SPIE Conference on Integration Issues in Large Commercial Media Delivery Systems*, 1995.