Research Comprehensive Final Exam: Action Recognition on a Mobile Robotic Platform

Matthew Loper
Computer Science Department
Brown University
Providence, RI 02912

March 7, 2008

Abstract

We present an approach for recognizing human action from a mobile platform. Three components are essential to our action recognition pipeline. First, a new off-the-shelf active sensor is employed to obtain observations. Next, we use our observables to generate a set of pose-inspired features. Finally, we use these features to obtain a history over actions, with the use of a Hidden Markov Model and a Gaussian likelihood function. The proposed approach is insensitive to the changing lighting conditions which can be troublesome for color-based methods. To our knowledge, we are the first to develop a mobile, uninstrumented, color-invariant 3D body-model based approach for action recognition.

Figure 1: iRobot PackBot EOD mobile robot, SwissRanger camera, and Bluetooth headset.

1 Introduction

Many have suggested that the "dull, dirty and dangerous" tasks of life should be automated. Indeed, many good automation examples fulfill some of those tasks, including robotic car assembly, automated teller machines, vending machines, packaging, and electronics assembly.

However, undesirable tasks still remain which are unsuitable for automation today. Keen readers may note that the commercial systems listed above are typically stationary, or (in some cases) move about in highly structured environments. One might also notice that they have limited means of communication with people: the above examples have some combination of keypad operation and manual teleoperation.

Of course, such commercial systems do not represent the state-of-the-art in robotics research. But it is instructive to consider these capabilities (mobility and communication) in terms of both commercial systems and research systems: why are they not more prevalent in commercial systems, and how could they be improved in current research? To help clarify the challenges involved, mobility and communication will be discussed in turn.

Mobility in a heterogeneous environment requires good perceptual capabilities. From sensors, a mobile robotic platform may be required to localize itself, identify objects, detect people, interpret their commands, and recognize their actions. Such capabilities may both aid a robot in its tasks, and (more importantly) help a robot avoid causing dangerous situations.

Many existing techniques are available for both localization [42] and object recognition [30, 45]. However, current techniques for people detection and action recognition from a mobile platform are greatly limited by a number of factors. These include the non-rigid structure of the human form, varied clothing, the variance of shapes among people, and the drawbacks of conventional sensors. Later, we will show how we attempt to address some of these issues.

Next, let us consider the communication abilities of commercially-available automation examples above. They

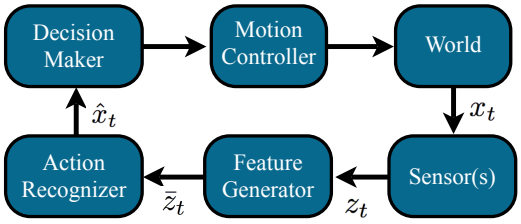


Figure 2: A depiction of the robot control loop. We are primarily concerned with the proper choice of sensor, feature generator, and action recognizer. Sensor output is denoted by z_t, generated features $\bar{z}_t$, state of the human x_t, and our estimated state of the human $\hat{x_t}$

are either operated by keypads, manually teleoperated, or (if working in a very constrained space) do not communicate at all. If robots are to work in tandem with people on "dull, dirty, and dangerous" tasks, humans should be able to communicate with robots in a natural way that does not require the full-time attention of the operator. Therefore, in addition to aiding mobility, we would also like to improve human-robot communication.

In order to work towards both of these objectives, we have developed an approach for mobile action recognition. This approach has been integrated into an experimental system, which is capable of recognizing, following, and identifying a few select gestures of a person, indoors and outdoors. This paper specificially focuses on our action recognition approach, and does not deeply evaluate or describe other parts of the system.

2 Related work

Three components are essential to our action recognition pipeline: a sensor, our feature generator, and our action recognizer. We will cover related work regarding each of these components in turn.

2.1 Sensor(s)

The most historically popular sensor for action recognition, at least in the field of robotics, has been the passive camera (see Appendix A for an overview). This instrument serves as a rich data source, and lends intuitive features because of its similarity to the human visual system. However, constructing invariants sometimes proves difficult due to many factors, including changing lighting conditions, the intricacies of perspective transformation, and the irregular sparsity of features available.

The use of active (time-of-flight) cameras for gesture recognition is a relatively new development. Hand-based gestures are detected with time-of-flight camera by Breuer et al [8] and Liu et al [29]. Pose recognition with these cameras is done by Fujimura et al [13], by using k-means to find clusters of pixels which may correspond to body parts. But these three approaches do not make use of a human body model, whose adjustable parameters have direct correspondence to the real world.

Knoop et al [23] does use an articulated model with a time-of-flight camera, as we do. But their system is intended to recover poses (rather than actions), and they require initial alignment between the person and the articulated body model, which we do not require.

Looking farther afield, speech is a natural medium for robot control, and has been applied in many robotic systems [36, 41, 12, 16, 38]. But speech does not work well with the noisy environments that often accompany dirty, dangerous work. Another source of noise may be the robot itself; a robot's own motors may (as they did in our case) make speech recognition more difficult. Nevertheless, speech is an important medium, and (though it is not described here) has been integrated into our system.

2.2 Feature generation

Historically, the most popular feature for mobile gesture recognition has been skin color [6, 36, 38, 32]. The use of this feature can be hampered by changing lighting conditions, and suffers in the context of skin-like colors in the environment [44], such as wood or some shades of paint. Such overlap between a skin color distribution and that of an environment is especially problematic when users have varying racial characteristics.

Some mobile methods augment skin-tone detection with face detection [12, 46] or shirt color detection [46]. Shirt color detection shares the same problems as skin color detection, and face detection is more helpful for person finding than for gesture recognition (though it can help in skin color initialization).

Our methods generate features from silhouettes, which are segmented from a depth image, thereby avoiding the problems with color detection. Some existing systems, such as those of Waldherr [46] or Liu [29], also use silhouettes for action recognition. But these template-based systems do not use a body model. We have taken the path of evaluating a body-model based recognition system, on the basis that the adjustment parameters (leg length, height) map directly to the real world, and make an allowance for view invariance.

Other existing systems, such as those of Ramanan [34] or Breuer [8], do use silhouettes and body models, but are concerned with pose and do not perform action recognition.

2.3 Action recognition

The two most popular basic methods for temporal gesture recognition are Dynamic Time Warping (DTW) and Hidden Markov Models (HMM's), as reviewed by Wu et al [47]. Other recognition models include image-based templates [5, 46, 29, 14], particle filters [7], and neural networks [46, 6].

Our use of an HMM with a Gaussian likelihood is similar to that of Pentland and Liu [33], though their work was on modeling and prediction of automobile drivers.

We cannot present a comprehensive review of the recognition literature, but a table of many important works is presented in Appendix A.

3 Problem formulation

Given a series of time-varying observations, we wish to infer the action states of a human over that same time period. We denote observations as $Z_{1:t}$ and action states as $X_{1:t}$. We pose this as a latent variable problem, in which a series of observed states depends probabilistically on hidden states. From this perspective, we wish to estimate the most likely history $\hat{X}_{1:t}$:

$$\hat{X}_{1:t} = \arg\max_{X_{1:t}} p(X_{1:t}|Z_{1:t}) \tag{1}$$

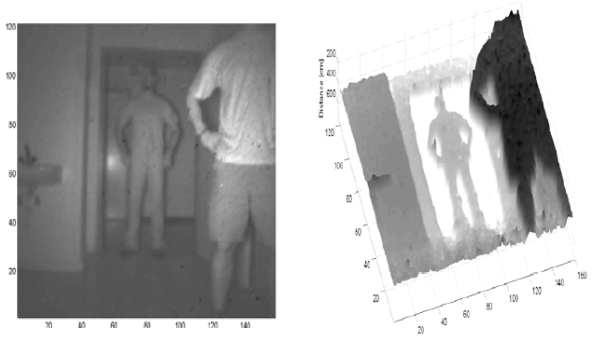
The following filtering equation is instrumental in solving for this history. It may be familiar to readers from both the forward algorithm in HMM's [37], and the state density propagation rule of a particle filter [18].

$$\underbrace{p(x_t|Z_{1:t})}_{\text{posterior}} \propto \underbrace{p\left(z_t|x_t\right)}_{\text{likelihood}} \underbrace{p(x_t|Z_{1:t-1})}_{\text{prior}} \tag{2}$$

$$\underbrace{p(x_t|Z_{1:t-1})}_{\text{prior}} = \int \underbrace{p\left(x_t|x_{t-1}\right)}_{\text{transition model}} \underbrace{p\left(x_{t-1}|Z_{1:t-1}\right)}_{\text{recurrence}} dx_{t-1}$$

Our goal is to appropriately infer an action history with the use of Equations 2 and 1. This requires realizing various aspects of these equations. For example, an observation z_i can take many forms, including color images, depth images, and laser-range scans. And the space of x_i must be defined: the question of whether to incorporate pose into that space is important, as the introduction of a hidden continuous variable changes our problem significantly. We must make these design choices while observing our constraints, which include real-time (at least 5 fps) operation and usability while the robot is mobile.

We make two major assumptions which guide our formulation.



(a) Conventional b/w image from SwissRanger. (b) Depth map of scene from SwissRanger.

Figure 3: Sample data returned from SwissRanger camera (image credit: Oggier et al [31]).

First, we assume that one (and only one) of a finite number of actions is being performed at any given time step.

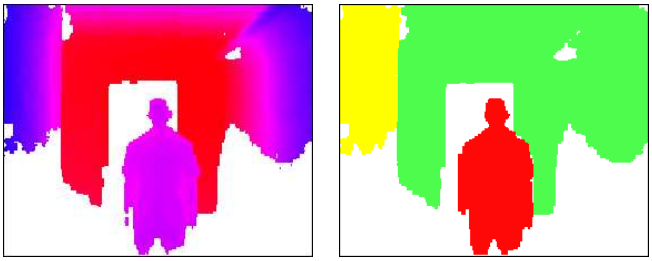
Second, we assume that actions can be detected from knowledge about a person's pose, without reference to external objects. Some actions (such as sitting down or waving) do not require references to external objects; and some (like picking up a ball) intimately require the recognition of surrounding objects. In order to make our problem tractable, we only consider actions that can be characterized by pose and its path over time.

Broadly speaking, the robot control loop in Figure 2 indicates the three subsystems that require description: our sensor, our feature generator, and our action recognizer. We will discuss our approach to each of these components in turn. In doing so, we will define the components of Equation 2: z_t is described in Section 4, the likelihood $p(z_t|x)$ is found in Section 6, and the transition model $p(x_t|x_{t-1})$ is also found in Section 6.

4 Sensor

We use a time-of-flight sensor known as the CSEM SwissRanger. The camera emits non-visible infrared light, and recovers a depth map by measuring the phase of the returned light. The intensity of each pixel indicates the distance to an object along that ray. A sample depth map is shown in Figure 3(b). Additional technical information regarding this camera can be found online [1].

The SwissRanger has many advantages over traditional sensors. Unlike conventional video, this sensor does not require external light to function, and is (for the most part)



(a) Depth image (b) Connected components

Figure 4: An image (a) from the SwissRanger is processed to obtain (b).

unaffected by external lighting conditions. Unlike color stereo, it does not require densely textured regions to recover a depth map. The frame rate is sufficient for many tasks at 13 frames per second, and unlike many laser-based sensors, it reconstructs a full 2D depth map.

The SwissRanger is not without drawbacks. The horizontal field of view is somewhat narrow at 48 degrees. Very reflective objects generate false readings, and may skew surrounding readings. Finally, the resolution is limited to 176x144, which might not be sufficient for all applications.

The physical setup of the camera is as follows. We placed the camera at eye-height, approximately 1.5 meters high. For optimal recognition, a person had to be greater than 1 meter away (as constrained by field of view), and less than 5 meters away (as constrained by camera resolution).

Next, we describe how an image z_i from our sensor is processed into a set of features $\bar{z}_i$.

5 Feature Generator

In order to generate features, we first segment the depth image into contiguous regions. A sample depth image is shown in Figure 4(a), and a resulting connected components image is shown in Figure 4(b). Contiguous regions are classified as either "person" or "non-person." Classification is accomplished by constructing histograms over contiguous regions, and classifying them according to a trained support vector machine. At the end of this process, we have a segmented region which we believe to belong to a person. (Note: this approach to segmentation and classification is not a novel claim of this paper, but is presented in our paper [25]).

Before proceeding further, let us first define some terms. A *pose* is a vector in the J-dimensional space of human joint angles. A *pose trajectory* (or just *trajectory*) is a defined as a bounded, continuous curve through this same space (two such trajectories are shown in Figure 5). A *traversal speed* (or just *speed*) is defined as an instantaneous absolute rate

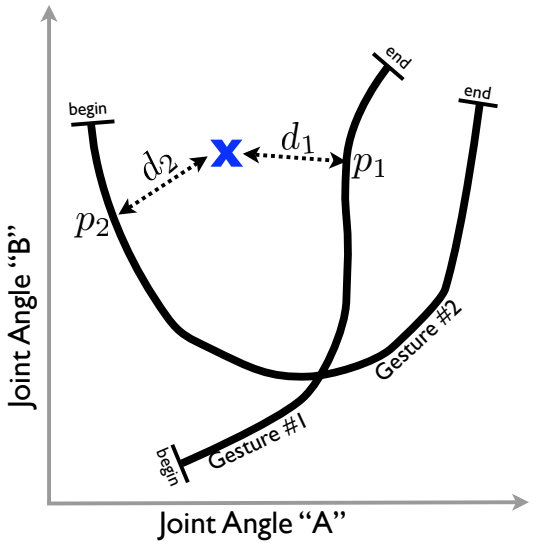


Figure 5: Two action trajectories and one pose (marked "x") are shown. Note that a pose may be projected upon both action trajectories, generating indicators of gesture progress $\{p_1, p_2\}$ and distances $\{d_1, d_2\}$. Speed is not shown, but is just $\{\partial p_1/\partial t, \partial p_2/\partial t\}$.

of movement through pose space.

We wish to recognize actions by characterizing a person's pose trajectories. Unfortunately, the recovery of these trajectories is greatly hindered by the high dimensionality of pose space. Unlike the simplified space of Figure 5, humans have upwards of 70 degrees of articulation. Because a brute-force search over poses is unrealistic, existing pose-recovery approaches make some allowances. These may include instrumenting subjects, using multiple widely spaced cameras, the dimensionality reduction of pose, and the use of a motion prior.

Mobile robots cannot rely on widely spaced, static cameras when following a person through an everyday scenario. Likewise, instrumenting a subject during interaction can be cumbersome, inflexible, costly, and does not allow interaction with the uninstrumented.

For these reasons, we choose to reduce the dimensionality of our pose search space (presented in this section) and the use of a motion prior (presented in the next section). We captured one prototypical trajectory for each action using a Vicon motion capture system. This gives us one trajectory for each action to be recognized. A sample trajectory, which we used for one of our gestures, is in Figure 6.

There are three important ways in which we can consider an arbitrary pose with respect to a trajectory, as shown in Figure 5. First, one might project the pose onto a trajec-

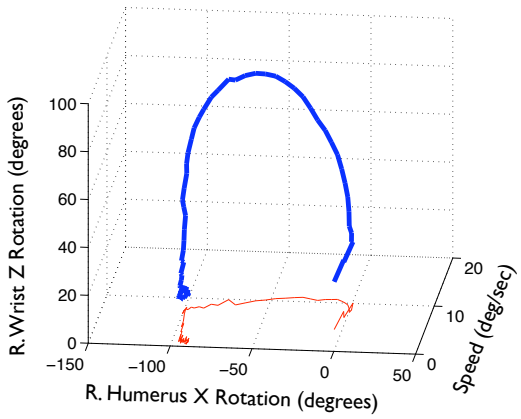


Figure 6: Pose trajectory for one of our trained gestures. Only two joint-angle dimensions can be visualized here, along with a dimension indicating the speed of traversal.

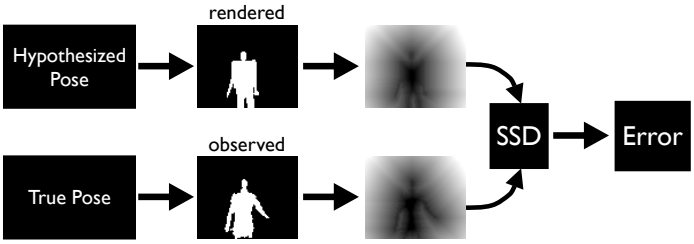


Figure 7: A distance between two poses is computed by measuring the chamfer distance between our body model and an observed silhouette.

tory, by finding the trajectory position closest to that pose. One might also find the distance between poses and trajectories. And finally, we can measure the speed of traversal over projections.

We cannot measure these variables directly. Therefore, for each of these variables (distance, progress, and speed), we introduce a surrogate observable variable which is intended to correlate with the original. These are described as follows.

To model *distance-to-projection*, we compare the silhouette of a cylindrical body model with that of a segmented version of our image, shown in Figure 4(b), by using Chamfer distance [3, 39]. The results of such a distance calculation can be seen in Figure 7. An edge-based likelihood estimator is used to estimate the distance between the hypothesized and the real silhouette.

To model the *projected progress* for one action, we start with a prototypical trajectory as specified by motion capture data. We then minimize the above distance to projection by using a simple search mechanism. This search is only tractable given the one-dimensional search space of the trajectory. Progress on action trajectory i is denoted p_i, and ranges between 0 (beginning of gesture) and 1 (end of gesture).

Finally, the *traversal speed* can be modeled by simply finding the difference between estimated projections in neighboring time steps. Our assumption is that an action's speed is distributed according to a non-uniform probability distribution: static gestures should have very low speed, for example, and dynamic gestures will be more likely to happen at some rates than at others. Speed along action trajectory i is denoted s_i.

The space of $\bar{z}_t$ is therefore that of all the projections, distances, and speeds of all actions; for Figure 5, for example, the set of features is as follows:

$$\bar{z}_t = \{d_1, d_2, p_1, p_2, s_1, s_2\}$$

6 Action Recognizer

Now that our features are described, we describe how they are used to infer state. A Hidden Markov Model (HMM) is an efficient and popular model for the problem of hidden-state estimation, given discrete hidden variables. Using an HMM requires the use of the following filtering equation, such that observation features $\bar{Z}_{1:t}$ are used to infer state x_t.

$$\underbrace{p(x_t|\bar{Z}_{1:t})}_{posterior} \propto \underbrace{p\left(\bar{z}_t|x_t\right)}_{likelihood} \sum_{x_{t-1}} \underbrace{p\left(x_t|x_{t-1}\right)}_{prior} \underbrace{p\left(x_{t-1}|\bar{Z}_{1:t-1}\right)}_{recurrence}$$

Because our observables are not discrete, they require an approximating distribution. The standard algorithms of an HMM (such as viterbi, forward-backward, Baum Welch) can be adapted for many kinds of continuous likelihood densities [21]. We model our likelihood $p(\bar{z}_t|x_t)$ as Gaussian.

As shown in Figure 8, our Markov chain divides each gesture into its static beginning pose, the middle, and the static end pose. We also require a state for "null" gestures, in which no gesture is being performed, and a state for segmentation errors: sometimes the person was improperly segmented from the background.

Our system considers a gesture "recognized" if the transition into the "end state" of a gesture had occurred 5 frames prior to the last frame. The last most likely state may sometimes be unreliable, which is why we use the benefit of hindsight to recognize our gestures.

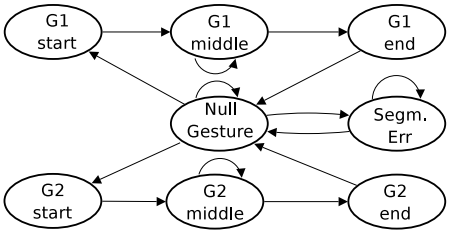


Figure 8: Gesture recognition Markov chain.

7 Results

Our current system runs on an iRobot packbot, shown in Figure 1, and is capable of following a person down hallways or in parking lots to a reasonable degree. It has been programmed to recognize and respond to two gestures: a "one-arm-up" gesture toggles following behavior, and a "two arms up" gesture initiates a door breach. In the accompanying videos, it can be seen that the overall system can function reasonably well, indoors and outdoors.

Eight sequences were collected and annotated for the evaluation of our methods. Dataset A consisted of the first four sequences, had 7 examples of each gesture, and used two male individuals. Dataset B consisted of the remaining sequences, had 16 examples of each gesture, and used a male (who was not in dataset A) and one female.

After training on dataset A, testing on dataset B revealed a precision of 88%, a recall of 93%, and a balanced F-score of 90%. Note that a balanced F-score combines precision and recall into one value, and is computed as follows:

$$F = \frac{(2 \cdot \text{precision} \cdot \text{recall})}{(\text{precision} + \text{recall})}$$

7.1 Comparison to an appearance model

Some may question the advantages of using a body model, given that appearance models do the job more simply. We believe that the use of a body model is advantageous because of the correspondence between model parameters and real-world.

To help substantiate this claim, we evaluated our approach against Temporal Templates (TmT), an appearance-based gesture recognition method by Bobick [5]. Like our method, TmT uses observed silhouettes to train a Gaussian likelihood function. In that method, labeled training sequences are obtained, each being the observed performance of one gesture. Each labeled sequence is then collapsed into a Motion Energy Image and a Motion History Image, as explained in [5]. Finally, Hu moments are measured on these training images, and fit to a multivariate Gaussian, which is in turn used as a likelihood.

	Training Dataset	Testing Dataset	Precision	Recall	F
TmT	A	A	0.76	0.88	0.82
Ours	A	A	0.85	1.00	0.92
TmT	A	B	1.00	0.24	0.39
Ours	A	B	0.88	0.93	0.90

Figure 9: A comparison between our method and that of Bobick, showing performance while varying the use of training and testing datasets.

s	d	p	Precision	Recall	F-measure
√	-	-	.13	.02	.03
-	√	-	.33	.02	.04
-	-	√	.50	.02	.04
-	√	√	.50	.02	.04
√	√	-	.52	.37	.43
√	-	√	.85	.65	.74
√	√	√	.88	.93	.90

Figure 10: Recognition performance is compared while using subsets of our three features (speed, distance, and progress), in order to gauge the relative importance of each feature. Speed is found to be the most discriminating feature overall.

As can be seen in Figure 9, our implementation of TmT performs reasonably well when trained on dataset A, and tested on the same dataset. Our system performs somewhat better in these conditions. But when TmT is trained on dataset A, and tested on dataset B, recall rates dropped significantly. Our method, on the other hand, still performed acceptibly.

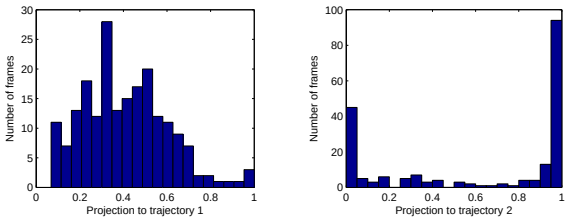
7.2 Feature evaluation

To show that each of our features contributes to recognition rates, we have evaluated performance measures in Figure 10. Note that recall drops significantly if only one feature is removed. Of the three features, distance appears to be the least informative of the three for our datasets.

Another important question is whether the observed distribution of features are Gaussian. An informal inspection of observed distributions suggests that although many of the states exhibit a Gaussian appearance, a few do not. Gesture progress is clearly non-Gaussian in a few cases, as shown in Figure 11. This is discussed in the following section.

8 Discussion

In this section, we will discuss speed constraints, out-of-plane enhancements, the performance of TmT, and the oc-



(a) Distribution of projections to trajectory 1.

(b) Distribution of projections to trajectory 2.

Figure 11: Both of these observed marginal feature distributions correspond to the "middle" state of gesture 1: as projected onto trajectory 1 in (a) and trajectory 2 in (b). Although (a) has a Gaussian appearance, (b) clearly does not.

casionally non-Gaussian nature of our observables.

A limiting factor of our approach, and one which may have steered previous researchers towards appearance-based models, is the expense of offscreen rendering and per-pixel comparison; each frame requires rendering the body model many times in order to find the gesture progress. Therefore, in order to achieve our frame rate of 7 frames-per-second, while allowing the person detection code to run concurrently, we were restricted to having only two gestures. However, we don't see this as a long-term problem; rendering is an easily parallelizable process, and Moore's law will surely help also.

A more significant issue is that of in-plane versus out-of-plane gestures. Our system only works with in-plane gestures, since out-of-plane gestures dampen the effectiveness of our connected-components-based person recognition. To allow for these out-of-plane gestures, a fruitful direction may be to provide feedback between our person recognition and gesture recognition components. Temporal coherence is also a cue that may be used.

One may wonder why Bobick's method performed so poorly on the test data in Figure 9. One possibility is that his method requires a better frame rate than the one available to us: the illustrations in that paper suggest a high degree of temporal continuity. But it is difficult to know exactly why failure occurs, because the underlying parameters of this appearance-based model do not map to real-world parameters; it was even admitted outright in that work that Hu moments may not have an intuitive meaning with real-world correspondence [5].

Finally, there is the issue of the non-Gaussian nature of observed features for some states. Because our gesture progress is truncated at 0 and 1, it is unsurprising that observed gesture progress values sometimes clamp to those values. Although a mixture of Gaussians might model these features acceptibly, it is desirable to deal with this clamping in a more rigorous manner. This is left for future work.

9 Conclusion

An approach was presented for recognizing human action from a mobile platform. Our sensor, feature generation, and recognition components were described. Our method was evaluated against an appearance-based model; proposed features were also evaluated against each other. Our system was shown to have a number of useful attributes, including tolerance to changing lighting, mobility, and reasonable recognition rates.

A Appendix

Author	Sensor(s)	Features	Recognition	Body Model?	Integration?	Gestures?
Starner[a][40]	color	seg+ellipse fitw	HMM	-	-	√
Ju & Black[b] [20]	bw	optical flow	curve recog	√, 2D	-	-
Bobick [5]	mono	silhouettes	templates	-	-	√
Campbell [9]	stereo	Azarbayejani et al, '96	HMM	-	-	√
Kahn[c] [22]	color stereo	edge+disprty+clr+motion	constraints	-	?	-
Kortencamp [27]	bw stereo	texture/depth	thresholds	√	√	√
Bregler [7]	color	xyt/hsv/texture	PF,HMM	-	-	√
Waldherr [46]	color	seg	NN/templates	-	√	√
Black[d] [4]	color	x+y from phicon	PF	-	-	√
Boehme[e] [6]	stereo	skin color	NN	-	√	√
Rittscher [35]	bw	contour-finding	PF	-	-	√
Wu[f] [47]						√
Aggarwal[g] [2]						√
Iba [16]	Cyberglove	Dim reduct and quant	HMM	√	?	√
Deutscher [10]	3 cams	sil+edges	annealed PF	√	-	-
Elgammal [11]	bw	edges	HMM	-	-	√
Rogalla [36]	stereo[h]	skin segm	TFD[i]	-	√	√
Saito[j] [38]	color+dff[k]	skin seg	unspecif	-	√	√
Sminchisescu [39]	color?	level sets, chamfer	none	-	-	-
Stiefelhagen[l] [41]	stereo	k-means on color/dist	HMM,PF	-	√	-
Kwolek [28]	color stereo	skin seg	HMM	-	-	√
Holte [15]	SR2	motion detection	3D histogram	-	-	√
Iba [17]	Cyberglove		HMM	√	√	√
Liu[m] [29]	SR2	seg	templates	-	-	√
Ramanan[n] [34]	mono	edges	see [16]	√, 2D	-	-
Knoop [24]	SR2/stereo		ICP	√	-	-
Fujimura [13]	SR2	k-means seg				-
Urtasun [43]	mono	templ-mtch → joints	SGPLVM	√	-	-
Park [32]	color	skin	HMM	-	√	√
Kojo [26]	stereo	face+sil+hand	CHMM	-	-	√
Hasanuzzaman [14]	color	skin seg	templates	-	-	√
Fransen [12]	stereo	face+seg	SHMM,PF	√	√	√
Breuer [8]	SR2	seg	PCA	√	-	-
Jenkins [19]	color	seg	PF,PCA	√	-	-

[a] Gaussian observation distribution used
[b] Legs and arms only, eight planar patches, 2 for each limb.
[c] Detects pointing only.
[d] Gesture boundaries were hand-segmented
[e] Two cameras on separate pan/tilt, disparity unused.
[f] Review paper
[g] Review paper
[h] depth used only for tracking, not for gest recog
[i] Thresholded fourier distance.
[j] This paper is very vague about its recognition methods.
[k] Depth from focus.
[l] Pointing recog only.
[m] Records trajectory of hand and gesture of hand.
[n] This is really a tracking paper

References

[1] SwissRanger specifications. http://www.swissranger.ch/main.php.

[2] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding: CVIU*, 73(3):428–440, 1999.

[3] Harry G. Barrow, Jay M. Tenenbaum, Robert C. Boles, and Helen C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *IJCAI*, pages 659–663, 1977.

[4] Michael J. Black and Allan D. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *ECCV (1)*, pages 909–924, 1998.

[5] A. Bobick and J. Davis. Real-time recognition of activity using temporal templates, 1996.

[6] Hans-Joachim Boehme, Anja Brakensiek, Ulf-Dietrich Braumann, Markus Krabbes, and Horst-Michael Gross. Neural architecture for gesture-based human-machine-interaction. *Lecture Notes in Computer Science*, 1371:219–??, 1998.

[7] Christoph Bregler. Learning and recognizing human dynamics in video sequences. In *CVPR*, pages 568–, 1997.

[8] Pia Breuer, Christian Eckes, and Stefan Müller. Hand gesture recognition with a novel ir time-of-flight range camera-a pilot study. In *MIRAGE*, pages 247–260, 2007.

[9] L. Campbell, D. Becker, A. Azarbayejani, A. Bobick, and A. Pentland. Invariant features for 3-d gesture recognition, 1996.

[10] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. *cvpr*, 02:2126, 2000.

[11] A. Elgammal, V. Shet, Y. Yacoob, and Larry Davis. Gesture recognition using a probabilistic framework for pose matching. In *ICARCV*, Singapore, December 2002.

[12] Benjamin R. Fransen, Vlad I. Morariu, Eric Martinson, Samuel Blisard, Matthew Marge, Scott Thomas, Alan C. Schultz, and Dennis Perzanowski. Using vision, acoustics, and natural language for disambiguation. In *HRI*, pages 73–80, 2007.

[13] Kikuo Fujimura, Youding Zhu, and Victor Ng-Thow-Hing. Estimating pose from depth image streams. In *Humanoid Robots*, pages 154–160, 2005.

[14] M. Hasanuzzaman, T. Zhang, V. Ampornaramveth, H. Gotoda, Y. Shirai, and H. Ueno. Adaptive visual gesture recognition for human-robot interaction using a knowledge-based software platform. *Robotics Autonomous Systems*, 55(8):643–657, 2007.

[15] M.B. Holte and T.B. Moeslund. Gesture recognition using a range camera. Technical Report 1601-3646, Laboratory of Computer Vision and Media Technology. Aalborg University, Aalborg, Denmark, February 2007.

[16] S. Iba, J. Vande, C. Paredis, and P. Khosla. An architecture for gesture-based control of mobile robots, 1999.

[17] Soshi Iba, Chris Paredis, and Pradeep Khosla. Interactive multimodal robot programming. In *9th International Symposium on Experimental Robotics*, June 2004.

[18] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[19] Odest Chadwicke Jenkins, German González, and Matthew Maverick Loper. Tracking human motion and actions for interactive robots. In *HRI*, pages 365–372, 2007.

[20] Shanon X. Ju, Michael J. Black, and Yaser Yacoob. Cardboard people: A parameterized model of articulated image motion. In *FG*, pages 38–44, 1996.

[21] B H Juang, S E Levinson, and M M Sondhi. Maximum likelihood estimation for multivariate mixture observations of markov chins. *IEEE Trans. Inf. Theor.*, 32(2):307–309, 1986.

[22] Roger E. Kahn, Michael J. Swain, Peter N. Prokopowicz, and R. James Firby. Gesture recognition using the perseus architecture. Technical Report TR-96-04, University of Chicago, 19, 1996.

[23] Steffen Knoop, Stefan Vacek, and Rudiger Dillmann. Sensor fusion for 3d human body tracking with an articulated 3d body model. In *ICRA 2006: Proceedings 2006 IEEE International Conference on Robotics and Automation*, pages 1686–1691, May 2006.

[24] Steffen Knoop, Stefan Vacek, and Rdiger Dillmann. Modeling joint constraints for an articulated 3d human body model with artificial correspondences in ICP. In *Proceedings of the International Conference on Humanoid Robots (Humanoids 2005)*, Tsukuba, Japan, Dec 31 2005.

[25] Nathan Koenig, Matthew Loper, Sonia Chernova, Chad Jenkins, and Chris Jones. Hands-free interaction for human-robot teams. *Workshop on Social Interaction with Intelligent Indoor Robots (ICRA)*, pages 1–2, 2008.

[26] Naoki Kojo, Tetsunari Inamura, Kei Okada, and Masayuki Inaba. Gesture recognition for humanoids using proto-symbol space. In *Humanoid Robots*, pages 76–81, September 2006.

[27] David Kortenkamp, Eric Huber, and R. Peter Bonasso. Recognizing and interpreting gestures on a mobile robot. In *AAAI/IAAI, Vol. 2*, pages 915–921, 1996.

[28] Bogdan Kwolek. Visual system for tracking and interpreting selected human actions. In *WSCG*, 2003.

[29] Xia Liu and Kikuo Fujimura. Hand gesture recognition using depth data. In *Automatic Face and Gesture Recognition*, pages 17–19, 2004.

[30] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1150, Washington, DC, USA, 1999. IEEE Computer Society.

[31] T. Oggier, M. Lehmann, R. Kaufmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, and N. Blanc. An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger). In L. Mazuray, P. J. Rogers, and R. Wartmann, editors, *Optical Design and Engineering. Edited by Mazuray, Laurent; Rogers, Philip J.; Wartmann, Rolf. Proceedings of the SPIE, Volume 5249, pp. 534-545 (2004).*, volume 5249 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 534–545, February 2004.

[32] Hye Sun Park, Eun Yi Kim, Sang Su Jang, Se Hyun Park, Min Ho Park, and Hang Joon Kim. *Pattern Recognition and Image Analysis: HMM-Based Gesture Recognition for Robot Control*. Springer Berlin / Heidelberg, 2005.

[33] Alex Pentland and Andrew Liu. Modeling and prediction of human behavior. *Neural Computation*, 11(1):229–242, 1999.

[34] Deva Ramanan, D. A. Forsyth, and Andrew Zisserman. Strike a pose: Tracking people by finding stylized poses. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 271–278, Washington, DC, USA, 2005. IEEE Computer Society.

[35] Jens Rittscher and Andrew Blake. Classification of human body motion. In *ICCV (1)*, pages 634–639, 1999.

[36] O. Rogalla, M. Ehrenmann, R. Zollner, R. Becher, and R. Dillmann. Using gesture and speech control for commanding a robot assistant. In *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*, pages 454–459, 2002.

[37] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[38] H. Saito, K. Ishimura, M. Hattori, and T. Takamori. Multi-modal human robot interaction for map generation. In *41st SICE Annual Conference*, volume 5, pages 2721 – 2724, 2002.

[39] Cristian Sminchisescu and Alexandru Telea. Human pose estimation from silhouettes - a consistent approach using distance level sets. In *WSCG*, pages 413–420, 2002.

[40] T.E. Starner and A.P. Pentland. Visual recognition of american sign language using hidden markov models. In *International Conference on Automatic Face and Gesture Recognition*, pages XX–YY, 1995.

[41] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *IEEE/RSJ International Conference Intelligent Robots and Systems*, volume 3, pages 2422– 2427, Sendai, Japan, 2004.

[42] Sebastian Thrun. Probabilistic robotics. *Commun. ACM*, 45(3):52–57, 2002.

[43] Raquel Urtasun, David J. Fleet, Aaron Hertzmann, and Pascal Fua. Priors for people tracking from small training sets. In *ICCV*, pages 403–410, 2005.

[44] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques, 2003.

[45] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2002.

[46] S. Waldherr, S. Thrun, and R. Romero. A gesture-based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173, 2000.

[47] Ying Wu and Thomas S. Huang. Vision-based gesture recognition: A review. *Lecture Notes in Computer Science*, 1739:103+, 1999.