Human Activity Recognition with Metric Learning

Du Tran and Alexander Sorokin

University of Illinois at Urbana-Champaign
Urbana, IL, 61801, USA
{dutran2,sorokin2}@uiuc.edu

Abstract. This paper proposes a metric learning based approach for human activity recognition with two main objectives: (1) reject unfamiliar activities and (2) learn with few examples. We show that our approach outperforms all state-of-the-art methods on numerous standard datasets for traditional action classification problem. Furthermore, we demonstrate that our method not only can accurately label activities but also can reject unseen activities and can learn from few examples with high accuracy. We finally show that our approach works well on noisy YouTube videos.

1 Introduction

Human activity recognition is a core unsolved computer vision problem. There are several reasons the problem is difficult. First, the collection of possible activities appears to be very large, and no straightforward vocabulary is known. Second, activities appear to compose both across time and across the body, generating tremendous complexity. Third, the configuration of the body is hard to transduce, and there is little evidence about what needs to be measured to obtain a good description of activity.

Activity can be represented with a range of features. At low spatial resolution when limbs cannot be resolved, flow fields are discriminative for a range of motions [1]. At higher spatial resolutions one can recover body configuration and reason about it [2,3]. There is strong evidence that 3D configuration can be inferred from 2D images (e.g. [4,5,6]; see also discussion in [7]), which suggests building appearance features for body configuration. Such appearance features include: braids [8]; characteristic spatio-temporal volumes [9]; motion energy images [10]; motion history images [10]; spatio-temporal interest points [11,12,13]; nonlinear dimensionality reduced stacks of silhouettes [14]; an extended radon transform [15]; and silhouette histogram of oriented rectangle features [16]. Generally, such features encode (a) what the body looks like and (b) some context of motion. We follow this general pattern with some innovations (Section 2).

An activity recognition process should most likely have the following properties: **Robustness:** features should be relatively straightforward to obtain from sequences with reasonable accuracy, and should demonstrate good noise behaviour. **Discriminative:** at least for the primitives, one would like discriminative

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 549–562, 2008.

rather than generative models, so that methods can focus on what is important about the relations between body configuration and activity and not model irrelevant body behaviour. **Rejection:** activity recognition is going to be working with a set of classes that is not exhaustive for the forseeable future; this means that when a system encounters an unknown activity, it should be labelled unknown.

The whole set of requirements is very demanding. However, there is some evidence that activity data may have the special properties needed to meet them. First, labelling motion capture data with activity labels is straightforward and accurate [17]. Second, clustering multiple-frame runs of motion capture data is quite straightforward, despite the high dimensions involved, and methods using such clusters do not fail (e.g. [18]). Third, motion capture data compresses extremely well [19]. All this suggests that, in an appropriate feature space, motion data is quite easy to classify, because different activities tend to look quite strongly different. Following that intuition, we argue that a metric learning algorithm (e.g. [20,21,22,23]) can learn an affine transformation to a good discriminative feature space even using simple and straightforward-to-compute input features.

1.1 Contributions of the Paper

This paper has the following contributions:

1. Proposes a metric learning based approach for human activity recognition with the abilities to reject unseen activities and to learn with few training examples (Sections 3.4, 5.2).
2. Provides a large body of experimental evidence showing that quite simple appearance features (Section 2) work better than more complex ones (Section 5.1).
3. Demonstrates that our approach achieves strong results on a realistic dataset despite the noise (Section 6).

2 Motion Context Descriptor

Local descriptor. Our frame descriptor is a histogram of the silhouette and of the optic flow inside the normalized bounding box. We scale the bigger side of the bounding box to a fixed size $M (M = 120)$ preserving the aspect ratio. The scaled box is then placed at the center bottom of an $M \times M$ square box padded with zeros. We use this transformation to resample the values of the flow vectors and of the silhouette.

The optic flow measurements are split into horizontal and vertical channels. To reduce the effect of noise, each channel is smoothed using median filter. This gives us two real-valued channels F_x and F_y. The silhouette gives us the third (binary) channel S. Each of the 3 channels is histogrammed using the same technique: The normalized bounding box is divided into 2×2 sub-windows. Each sub-window is then divided into 18 pie slices covering 20 degrees each. The

center of the pie is in the center of the sub-window and the slices do not overlap. The values of each channel are integrated over the domain of every slice. The result is a $72(2\times2\times18)$-dimensional histogram. By concatenating the histograms of all 3 channels we get a 216-dimensional frame descriptor.

In our experiments, we also experimented with 3×3 and 4×4 sub-windows. 3×3 is not different from 2×2, but 4×4 decreases the performance by 5-7%. The radial histograms are meaningless when the sub-windows are getting too small.

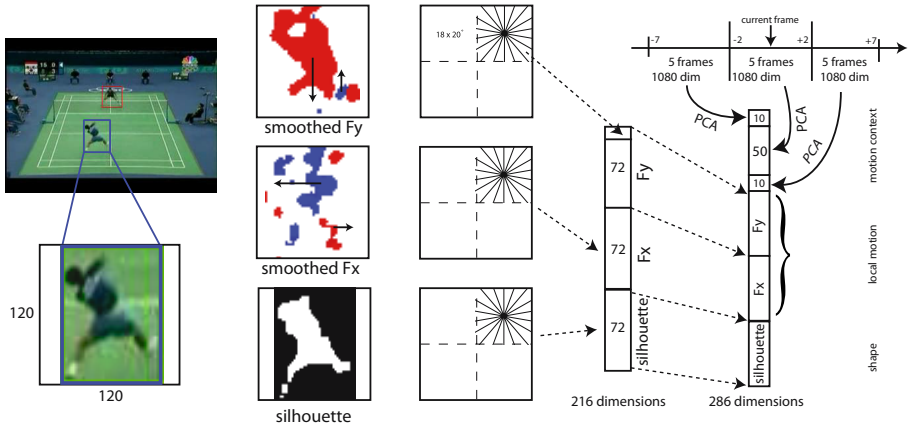

Fig. 1. Feature Extraction: The three information channels are: vertical flow, horizontal flow, silhouette. In each channel, the measurements are resampled to fit into normalized (120×120) box while maintaining aspect ratio. The normalized bounding box is divided into 2x2 grid. Each grid cell is divided into 18-bin radial histogram (20^o per bin). Each of the 3 channels is separately integrated over the domain of each bin. The histograms are concatenated into 216 dimensional frame descriptor. 5-frame blocks are projected via PCA to form medium scale motion summaries. The first 50 dimensions are kept for the immediate neighborhood and the first 10 dimensions are kept for each of the two adjacent neighborhoods. The total 70-dimensional motion summary is added to the frame descriptor to form the motion context.

Motion context. We use 15 frames around the current frame and split them into 3 blocks of 5 frames: past, current and future. We chose a 5-frame window because a triple of them makes a 1-second-long sequence (at 15 fps). The frame descriptors of each block are stacked together into a 1080 dimensional vector. This block descriptor is then projected onto the first N principal components using PCA. We keep the first 50, 10 and 10 dimensions for the current, past and future blocks respectively. We picked 50, 10 and 10 following the intuition that local motion should be represented in better detail than more distant ones. The resulting 70-dimensional context descriptor is appended to the current frame descriptor to form the final 286-dimensional motion context descriptor.

We design our features to capture local appearance and local motions of the person. Our Motion Context descriptor borrows the idea of radial bins from the

Shape Context [24] and of the noisy optic flow measurements from the "30-pixel man" [1]. We append a summary of the motion around the frame to represent medium-scale motion phenomena. We assume that the bounding box of the actor together with the silhouette mask is provided. In this work we use background subtraction to obtain the silhouette and the bounding box. These are often noisy, however our feature representation seems to be tolerant to some level of noise. Our experiments with badminton sequences show that when the noise is too extreme, it starts to affect the accuracy of activity recognition. We compute optic flow using Lucas-Kanade algorithm [25].

3 Action Classification Models

3.1 Naïve Bayes

Naïve Bayes requires the probability $P(frame|l)$ of the frame given the label l. To compute this probability we apply vector quantization via K-Means. After vector quantization the frame is represented by a word w_i and the probability is estimated by counting with Laplace smoothing: $P(w_i|l) = \frac{c(w_i,l)+1}{c(w,l)+K}$ where $c(w_i, l)$ is the numbers of times the word w_i occurred with the label l and $c(w, l)$ is the total number of words with the label l. Assuming uniform prior $P(l)$, ignoring $P(seq)$ and using Bayes rule we get the following prediction rule:

$$l = \operatorname*{argmax}_l P(l|seq) = \operatorname*{argmax}_l \sum_{i=1}^{m} \log P(w_i|l) \tag{1}$$

3.2 1-Nearest Neighbor

1NN classifier assigns a label to every query frame by finding the closest neighbor among training frames and propagating the label from the neighbor to the query frame. Every frame of the query sequence then votes for the label of the sequence. The label of the sequence is determined by the majority. Note that the voting provides us with smoothing and robustness to noise and thus we do not need to use more than one nearest neighbor.

3.3 1-Nearest Neighbor with Rejection

Nearest Neighbors with Rejection work by fixing a radius R and ignoring points further than R. If no neighbor is found within R, the query frame is thus unseen and receives the label "unobserved". The sequence is then classified by the majority vote including the "unobserved" label. We also consider the classifier that does rejection after metric learning. We manually choose the rejection radius to achieve equal accuracy on the discriminative and rejection tasks. The rejection radius can be chosen by cross validation to achieve desired trade-off between the discriminative and rejection tasks.

3.4 1-Nearest Neighbor with Metric Learning

Nearest neighbors crucially depend on the metric of the embedding space. Among metric learning algorithms ([20,21,22,23]), Large Margin Nearest Neighbors (LMNN) [22] are especially tailored to k-NN classifiers. We briefly state LMNN below.

LMNN learns a Mahalanobis distance D:

$$D(x_i, x_j) = (x_i - x_j)^T M(x_i - x_j) = \|L(x_i - x_j)\|^2 \tag{2}$$

LMNN tries to learn a matrix $M = L^T L$ that maximizes the distances between examples with different labels and minimizes the distances between nearby examples with the same label.

Minimize:

$$\sum_{ij} \eta_{ij}(x_i - x_j)^T M(x_i - x_j) + c\sum_{ijl} \eta_{ij}(1 - y_{il})\xi_{ijl}$$

Subject to:

$$\begin{array}{ll} (i) & (x_i - x_l)^T M(x_i - x_l) - (x_i - x_j)^T M(x_i - x_j) \geq 1 - \xi_{ijl} \\ (ii) & \xi_{ijl} \geq 0 \\ (iii) & M \succeq 0 \end{array} \tag{3}$$

where y_{ij} is a binary value indicating whether points x_i and x_j are in the same class and η_{ij} is a binary value indicating whether x_j is a selected nearby neighbor of x_i with the same class, ξ_{ijl} are slack variables. In the objective function, the first term minimizes the distances between all training examples and their selected neighbors. The second term maximizes the margin (relaxed by slack variables) between same-label distances (x_i to x_j) and different-label distances (x_i to x_l) of all training examples. We used the source code kindly provided by the authors of [22].

LMNN learns a global transformation matrix, but its objective is designed to capture the local manifold structure by selecting k nearby neighbors. Normally k is small and in our experiments, we use $k = 3$.

Data subsampling. We note that it is important to subsample training data before applying metric learning. Applying metric learning without subsampling training data will not help in discriminative task and even decreases the performance by 6-8%. This phenomenon is easy to understand. Without subsampling the training examples, the k selected neighbors of every frame are always the neighboring frames from the same sequence. Therefore minimizing the distances between examples with the same label is not helpful. In our experiment, we subsample training examples by the ratio 1:4, choosing 1 from every 4 consecutive frames.

LMNN significantly improves recognition accuracy when it operates in the complete feature space. However it is computationally expensive. We studied the improvements produced by LMNN if the feature space is restricted to be low-dimensional. There are two immediately obvious ways to reduce dimensionality: PCA and random projections (we use [26]). We used a range of dimensions from 5 to 70 with a step of 5. The results are discussed in Section 5.2.

4 Experimental Setup

4.1 Description of the Datasets

For our experiments we used **5 datasets**: 3 datasets presented in the literature and 2 new datasets. The **Weizman dataset** [9] contains 81 isolated sequences of 9 actors performing 9 activities. We use an augmented and more difficult version with 93 isolated sequences of 9 actors and 10 activities with 3 extra sequences. The **UMD dataset** [27] contains 100 sequences of 10 activities performed 10 times each by only one actor. The **IXMAS dataset** [28] contains 36 sequences in which 12 actors perform 13 actions. Each sequence is captured in 5 different views. **Our dataset 1** consists of 532 high resolution sequences of 14 activities performed by 8 actors. **Our dataset 2** consists of 3 badminton sequences downloaded from Youtube. The sequences are 1 single and 2 double matches at the Badminton World Cup 2006.

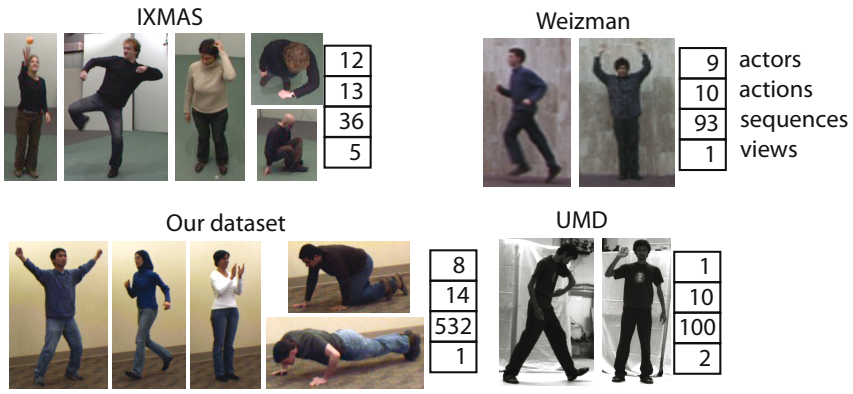

Fig. 2. The variations in the activity dataset design: **Weizman**: multiple actors, single view and only one instance of activity per actor, low resolution (80px). **Our**: multiple actors, multiple actions, extensive repetition, high resolution (400px), single view. **IXMAS**: Multiple actors, multiple synchronized views, very short sequences, medium-low resolution (100,130,150,170,200px). **UMD**: single actor, multiple repetitions, high resolution (300px).

4.2 Evaluation Protocols

We evaluate the accuracy of the activity label prediction for a query sequence. Every sequence in a dataset is used as a query sequence. We define 7 evaluation protocols by specifying the composition of the training set w.r.t. the query sequence. Leave One Actor Out (**L1AO**) excludes all sequences of the same actor from the training set. Leave One Actor-Action (**L1AAO**) excludes all sequences matching both action and actor with the query sequence. Leave One View Out (**L1VO**) excludes all sequences of the same view from the training set. This

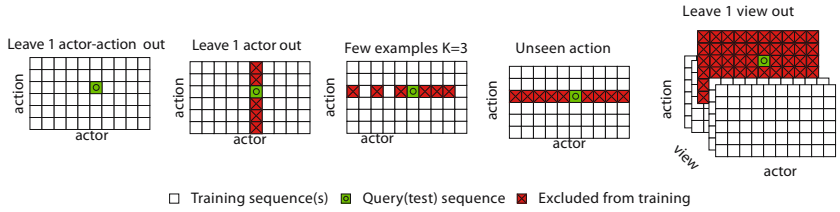

Fig. 3. Evaluation protocols: **Leave 1 Actor Out** removes all sequences of the same actor from the training set and measures prediction accuracy. **Leave 1 Actor-Action Out** removes all examples of the query activity performed by the query actor from the training set and measures prediction accuracy. This is more difficult task than L1AO. **Leave 1 View Out** measures prediction accuracy across views . **Unseen Action** removes all examples of the same action from the training set and measures rejection accuracy. **Few Examples-K** measures average prediction accuracy if only K examples of the query action are present in the training set. Examples from the same actor are excluded.

protocol is only applicable for datasets with more than one view(UMD and IXMAS). Leave One Sequence Out (**L1SO**) removes **only** the query sequence from the training set. If an actor performs every action once this protocol is equivalent to L1AAO, otherwise it appears to be easy. This implies that vision-based interactive video games are easy to build. We add two more protocols varying the number of labeled training sequences. Unseen action (**UAn**) protocol excludes from the training set all sequences that have the same action as the query action. All other actions are included. In this protocol the correct prediction for the sequence is not the sequence label, but a special label "reject". Note that a classifier always predicting "reject" will get 100% accuracy by UAn but 0% in L1AO and L1AAO. On the contrary, a traditional classifier without "reject" will get 0% accuracy in UAn.

Few examples (**FE-K**) protocol allows K examples of the action of the query sequence to be present in the training set. The actors of the query sequences are required to be different from those of training examples. We randomly select K examples and average over 10 runs. We report the accuracy at K=1,2,4,8. Figure 3 shows the example training set masks for the evaluation protocols.

5 Experimental Results

5.1 Simple Feature Outperforms Complex Ones

We demonstrate that our approach achieves state of the art discriminative performance. Table 2 compares our performance with published results. We show that on a large number of standard datasets with closed world assumption we easily achieve state-of-the-art perfect accuracy. Note that there are two versions of Weizman dataset, the original one contains 9 actions while the augmented version has 10. Our model achieves perfect accuracy on both Weizman datasets. For UMD dataset, we find that, it is easy to achieve 100% accuracy with train

Table 1. Experimental Results show that conventional discriminative problems **L1AAO,L1AO,L1SO** are easy to solve. Performance is in the high 90's (consistent with the literature). Learning with few examples **FE-K** is significantly more difficult. Conventional discriminative accuracy is not a good metric to evaluate activity recognition, where one needs to refuse to classify novel activities. Requiring rejection is expensive; the objective **UNa** decreases discriminative performance. In the table bold numbers show the best performance among rejection-capable methods. **N/A** denotes the protocol being inapplicable or not available due to computational limitations.

Dataset	Algorithm	Chance	Protocols								
			Discriminative task				Reject	Few examples			
			L1SO	L1AAO	L1AO	L1VO	UNa	FE-1	FE-2	FE-4	FE-8
Weizman	NB(k=300)	10.00	91.40	93.50	95.70	N/A	0.00	N/A	N/A	N/A	N/A
	1NN	10.00	95.70	95.70	96.77	N/A	0.00	53.00	73.00	89.00	96.00
	1NN-M	10.00	100.00	100.00	100.00	N/A	0.00	72.31	81.77	92.97	100.00
	1NN-R	9.09	83.87	84.95	84.95	N/A	84.95	17.96	42.04	68.92	84.95
	1NN-MR	9.09	**89.66**	**89.66**	**89.66**	N/A	**90.78**	N/A	N/A	N/A	N/A
Our	NB(k=600)	7.14	98.70	98.70	98.70	N/A	0.00	N/A	N/A	N/A	N/A
	1NN	7.14	98.87	97.74	98.12	N/A	0.00	58.70	76.20	90.10	95.00
	1NN-M	7.14	**99.06**	**97.74**	**98.31**	N/A	0.00	88.80	94.84	95.63	98.86
	1NN-R	6.67	95.86	81.40	82.10	N/A	81.20	27.40	37.90	51.00	65.00
	1NN-MR	6.67	**98.68**	**91.73**	**91.92**	N/A	**91.11**	N/A	N/A	N/A	N/A
IXMAS	NB(k=600)	7.69	80.00	78.00	79.90	N/A	0.00	N/A			
	1NN	7.69	81.00	75.80	80.22	N/A	0.00				
	1NN-R	7.14	**65.41**	**57.44**	**57.82**	N/A	**57.48**				
UMD	NB(k=300)	10.00	100.00	N/A	N/A	97.50	0.00	N/A			
	1NN	10.00	100.00	N/A	N/A	97.00	0.00				
	1NN-R	9.09	**100.00**	N/A	N/A	**88.00**	**88.00**				

Table 2. Accuracy Comparison shows that our method achieves state of the art performance on large number of datasets. *-full 3D model (i.e. multiple camera views) is used for recognition.

Dataset	Weizman9						Weizman10			UMD			IXMAS			
Method	[29]	[30]	[31]	[9]	[16]	**Our**	[32]	[14]	**Our**	[27]	[14]	**Our**	[33]	[28]*	**Our**	
Accuracy	72.8	92.6	98.8	99.67	**100**	**100**	82.6	97.78	**100**	**100**	**100**	**100**	80.06	**93.33**	**81**	

and test on the same actor, playing the same action in the same view. In this case even L1VO achieved 97.5% accuracy on this dataset. On IXMAS dataset, [28] report higher (93.33%) accuracy, however they use full 3D model.

5.2 Metric Learning Improves Action Classification

We demonstrate that metric learning significantly improves human activity recognition performance in: (1) discriminative task, (2) rejection task, and (3) few examples. On traditional action recognition problem, 1NN-M achives almost perfect accuracy and outperforms all state-of-the-art methods. For rejection task, 1NN-MR improves the accuracy about 5% on Weizman dataset and 10% on our dataset comparing to 1NN-R. For learning with few examples, 1NN-M significantly improves the accuracy. Specifically, for 1-example, 1NN-M improves

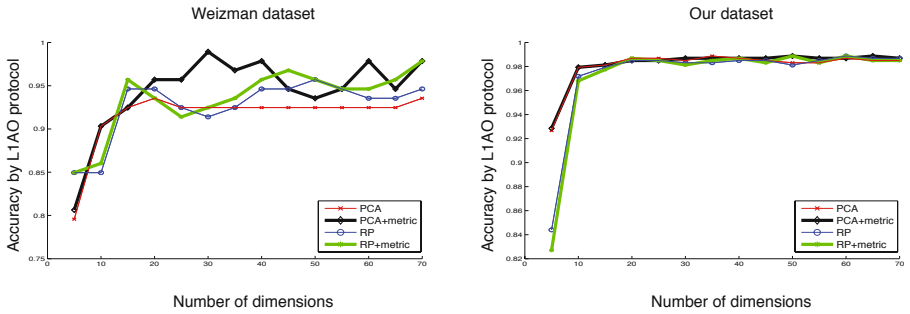

Fig. 4. LMNN with Dimension Reduction: On Weizman dataset, LMNN clearly improves PCA ($2.8 \pm 2.0\%$) and almost improves random projections ($0.8 \pm 1.2\%$). On our dataset, LMNN improvements are not present with few dimensions on the closed world classification task ($0.1 \pm 0.2\%$ from PCA and $0.1 \pm 0.5\%$ from random projection); The improvement is 2% in high dimensions and 3%-10% in rejection task.

about 20% accuracy on Weizman dataset and 30 % accuracy on our dataset. We show that our approach achieves about 72.31% accuracy on Weizman dataset and 88.80% on our dataset for action classification with only one training example. In low dimensions there is not much benefit from LMNN (Fig 4). The only clear improvement appears on Weizman dataset with PCA. In other cases of low dimensionality produce very little improvement if any.

6 Video Labeling with Rejection

How would we spot activities in practice? We would take a video, label some of the example activities and propagate the labels to unseen video. We follow this scenario and apply our algorithm to Youtube videos. We work with 3 badminton match sequences: 1 single and 2 double matches of the Badminton World Cup 2006.

Fig. 5. Our Dataset 2: Example frames from badminton sequences collected from Youtube. The videos are low resolution (60-80px) with heavy compression artifacts. The videos were chosen such that background subtraction produced reasonable results.

Table 3. Label sets for badminton action recognition

Problem	Label set
1. Type of motion	*run, walk, hop, jump, unknown*
2. Type of shot	*forehand, backhand, smash, unknown*
3. Shot detection	*shot, non-shot*

For a badminton match we define 3 human activity recognition problems shown in table 3. Problem 1 is to classify the type of motion of each player. Problem 2 is to classify the shot type. Problem 3 is to predict the moment of the shot. The players closer to the camera are very different from the players in the back. We therefore define two different "views" and evaluate the labeling performance for each view separately as well as for both views combined. One of the sequences was manually labeled for training and quantitative evaluation. The first half of the sequence is used for training, while the second half is used for testing. For problems 1 and 2 we measure prediction accuracy. For problem 3 we measure the distance from the predicted shot instant to the labeled one.

Table 4. Quantitative evaluation of video labeling show the prediction accuracy of **1NN**, **1NN-R**, **1NN-M**, and **1NN-MR** for the video labeling task. One Youtube sequence was manually annotated. The first half was used for training, the second half for evaluation. View 1 shows significantly better results due to higher resolution on the person giving more stable segmentation and less noisy flow computation. **1NN** works well in the closed world. However it performs poorly when it is applied to the open world. The underlined performance (in red) is below chance. **1NN-M** improves 1-6% from **1NN**. **1NN-MR** improves 0.5-4% from **1NN-R** in "view 2" but the other views.

Problem	Algo	View 1	View 2	2 Views	Chance	Assumption
1. Motion	1NN	75.81	63.66	71.30	25.00	close
2. Shot	1NN	88.84	81.50	74.55	33.33	close
1. Motion	1NN-M	**76.46**	**69.25**	**71.89**	25.00	close
2. Shot	1NN-M	**89.52**	**86.23**	**78.82**	33.33	close
1. Motion	1NN	42.72	24.93	34.04	20.00	open
2. Shot	1NN	26.49	23.75	21.98	25.00	open
1. Motion	1NNR	**57.73**	47.95	**53.37**	20.00	open
2. Shot	1NNR	**63.45**	52.29	52.15	25.00	open
1. Motion	1NN-MR	55.29	**48.44**	52.03	20.00	open
2. Shot	1NN-MR	62.72	**56.64**	**54.55**	25.00	open

Labeling with **1NN** achieves very high accuracy in the "view 1". The "view 2" and combined views are more challenging. In "view 1" most of the frames have the figures correctly segmented, while in the "view 2" the segmentation often looses legs and arms of the player. Furthermore as the resolution decreases, the quality of the optic flow degrades. These factors make prediction problem on "view 2" very difficult. The combination of the views presents another challenge. We distinguish forehand and backhand shots, however forehand shot in one view

is similar to the backhand shot in the other view. This further degrades the classifier performance. Consistently with the structured dataset results, **1NN-R** performs worse than **1NN**, because the rejection problem is difficult.

1NN-M improves 1-6% from **1NN** on closed world. **1NN-MR** improves 0.5-4% performance on "view 2" but does not help on "view 1". In "view 1", some unseen activities are quite similar to some observed actions. For example, when the player stands and do nothing, we labelled as "unknown" motion and "unknown" shot. However it looks quite similar to "hop" motion and "backhand" shot because the camera looks from the back of the closer player. In this case, LMNN learns a metric for moving same-label inputs close together. Unfortunately, this transformation also collapses the unseen activities.

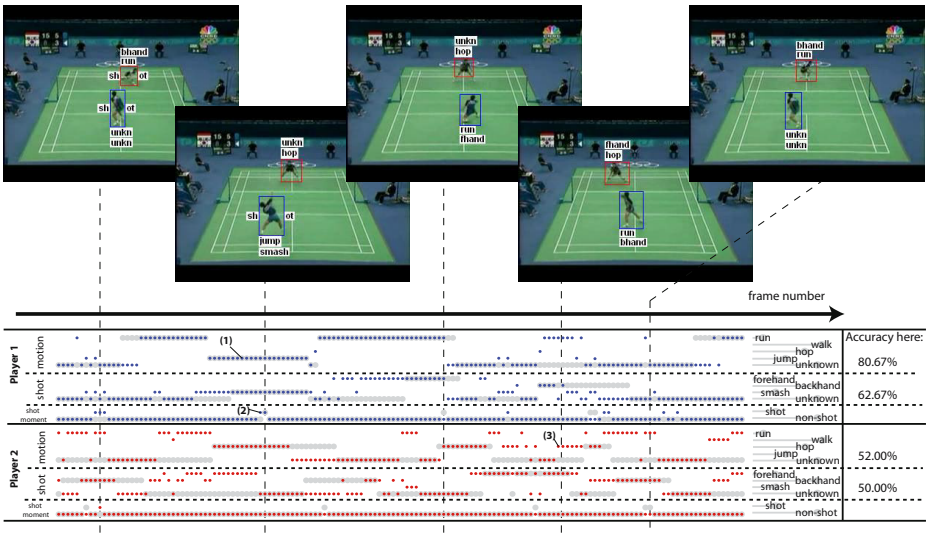

Fig. 6. Video labeling of Youtube sequences demonstrates the **1NN-R** in non-dataset environment. The figure is a snapshot from the video in the supplementary materials. Every frame is associated with one column. Every possible prediction is associated with one row. Gray dots denote the groundtruth which was labeled by hand. Blue dots are predictions for player 1 (close), red dots are predictions for player 2 (far). Predictions are grouped into 3 tasks (see table 3): 5 rows for motion type, 4 rows for the type of shot and 2 rows for shot instant prediction. The point marked (1) shows that at frame 1522 we predicted type of motion **jump** which is also labeled in the ground truth. The point marked (2) shows that at frame 1527 we predict that there is a shot. The ground truth marks the next frame. The point marked (3) shows that at frame 1592, we predict **hop**, while the groundtruth label is **unknown**. The accuracy numbers in the figure are computed for 150 frames shown in the figure.

As can be seen in table 5, our shot instant prediction accuracy works remarkably well: 47.9% of the predictions we make are within a distance of 2 from a labeled shot and 67.6% are within 5 frames. For comparison, a human observer

Table 5. Task 3. Shot prediction accuracy shows the percentage of the predicted shots that fall within the 5,7,9 and 11-frame windows around the groundtruth label shot frame. Note, that it is almost impossible for the annotator to localize the shot better than a 3-frame window (i.e. ± 1).

View	± 2**-shot**	± 3**-shot**	± 4**-shot**	± 5**-shot**
View 1	59.15	69.01	69.01	70.42
View 2	43.08	55.38	63.08	67.69
2 Views	47.97	59.46	63.51	67.57

has uncertainty of 3 frames in localizing a shot instant, because the motions are fast and the contact is difficult to see. The average distance from predicted shots to ground truth shots is 7.3311 while the average distance between two consecutive shots in the ground truth is 51.5938.

For complete presentation of the results we rendered all predictions of our method in an accompanying video. Figure 6 shows a snapshot from this video. The figure has several more frames shown with detected shots. Datasets and source code are available at: http://vision.cs.uiuc.edu/projects/activity/.

7 Discussion

In this paper, we presented a metric learning-based approach for human activity recognition with the abilities to reject unseen actions and to learn with few training examples with high accuracy. The ability to reject unseen actions and to learn with few examples are very crucial when applying human activity recognition to real world applications.

At present we observe that human activity recognition is limited to a few action categories in the closed world assumption. How does activity recognition compare to object recognition in complexity? One hears estimates of $10^4 - 10^5$ of objects to be recognized. We know that the number of even primitive activities that people can name and learn is not limited to a hundred. There are hundreds of sports, martial arts, special skills, dances and rituals. Each of these has dozens of distinct specialized motions known to experts. This puts the number of available motions into tens and hundreds of thousands. The estimate is crude, but it suggests that activity recognition is not well served by datasets which have very small vocabularies. To expand the number we are actively looking at the real-world (e.g. Youtube) data. However the dominant issue seems to be the question of action vocabulary. For just one YouTube sequence we came up with 3 different learnable taxonomies. Building methods that can cope gracefully with activities that have not been seen before is the key to making applications feasible.

Acknowledgments. We would like to thank David Forsyth for giving us insightful discussions and comments. This work was supported in part by Vietnam Education Foundation, in part by the National Science Foundation under IIS - 0534837, and in part by the Office of Naval Research under N00014-01-1-0890

as part of the MURI program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the VEF, the NSF or the Office of Naval Research.

References

1. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV, pp. 726–733 (2003)
2. Ramanan, D., Forsyth, D.: Automatic annotation of everyday movements. In: NIPS (2003)
3. Ikizler, N., Forsyth, D.: Searching video for complex activities with nite state models. In: CVPR (2007)
4. Howe, N.R., Leventon, M.E., Freeman, W.T.: Bayesian reconstruction of 3d human motion from single-camera video. In: Solla, S., Leen, T., Muller, K.R. (eds.) NIPS, pp. 820–826. MIT Press, Cambridge (2000)
5. Barron, C., Kakadiaris, I.: Estimating anthropometry and pose from a single uncalibrated image. CVIU 81(3), 269–284 (2001)
6. Taylor, C.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. CVIU 80(3), 349–363 (2000)
7. Forsyth, D., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D.: Computational aspects of human motion i: tracking and animation. Foundations and Trends in Computer Graphics and Vision 1(2/3), 1–255 (2006)
8. Niyogi, S., Adelson, E.: Analyzing and recognizing walking gures in xyt. In: Media lab vision and modelling tr-223. MIT, Cambridge (1995)
9. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: ICCV, pp. 1395–1402 (2005)
10. Bobick, A., Davis, J.: The recognition of human movement using temporal templates. PAMI 23(3), 257–267 (2001)
11. Laptev, I., Lindeberg, T.: Space-time interest points. In: ICCV, pp. 432–439 (2003)
12. Laptev, I., Prez, P.: Retrieving actions in movies. In: ICCV (2007)
13. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
14. Wang, L., Suter, D.: Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In: CVPR (2007)
15. Wang, Y., Huang, K., Tan, T.: Human activity recognition based on r transform. In: Visual Surveillance (2007)
16. Ikizler, N., Duygulu, P.: Human action recognition using distribution of oriented rectangular patches. In: ICCV Workshops on Human Motion, pp. 271–284 (2007)
17. Arikan, O., Forsyth, D., O'Brien, J.: Motion synthesis from annotations. In: SIGGRAPH (2003)
18. Arikan, O., Forsyth, D.A.: Interactive motion generation from examples. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pp. 483–490. ACM Press, New York (2002)
19. Arikan, O.: Compression of motion capture databases. In: ACM Transactions on Graphics: Proc. SIGGRAPH 2006 (2006)
20. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: NIPS (2002)
21. Yang, L., Jin, R., Sukthankar, R., Liu, Y.: An effcient algorithm for local distance metric learning. AAAI, Menlo Park (2006)

22. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: NIPS (2006)
23. Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I.: Information-theoretic metric learning. In: ICML (2007)
24. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI 24(4), 509–522 (2002)
25. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stero vision. IJCAI, 121–130 (1981)
26. Achlioptas, D.: Database-friendly random projections. In: ACM Symp. on the Principles of Database Systems (2001)
27. Veeraraghavan, A., Chellappa, R., Roy-Chowdhury, A.: The function space of an activity. In: CVPR, pp. 959–968 (2006)
28. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. CVIU (2006)
29. Niebles, J., Fei-Fei, L.: A hierarchical model of shape and appearance for human action classification. In: CVPR, pp. 1–8 (2007)
30. Ali, S., Basharat, A., Shah, M.: Chaotic invariant for human action recognition. In: ICCV (2007)
31. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biological inspired system for human action classification. In: ICCV (2007)
32. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. ACM Multimedia, 357–360 (2007)
33. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and viterbi path searching. In: CVPR (2007)