Feature Selection for tracker-less human activity recognition*

Plinio Moreno, Pedro Ribeiro, and José Santos-Victor

Instituto de Sistemas e Robótica & Instituto Superior Técnico
Portugal
{plinio,pedro,jasv}@isr.ist.utl.pt

Abstract. We address the empirical feature selection for tracker-less recognition of human actions. We rely on the appearance plus motion model over several video frames to model the human movements. We use the L_2Boost algorithm, a versatile boosting algorithm which simplifies the gradient search. We study the following options in the feature computation and learning: (i) full model vs. component-wise model, (ii) sampling strategy of the histogram cells and (iii) number of previous frames to include, amongst others. We select the features' parameters that provide the best compromise between performance and computational efficiency and apply the features in a challenging problem, the tracker-less and detection-less human activity recognition.

1 Introduction

Works on human activity recognition rely on detection and tracking algorithm in order to discriminate the human patterns present in videos [9]. On one hand, the detection algorithms are image-based approaches that segment the region of interest for further processing [6]. On the other hand, tracking algorithms use the detector output and data association techniques to segment video regions where the activity patterns are learnt and matched (e.g. [1]).

The state-of-the-art approaches for people detection and tracking have attained very good performances in challenging data sets (see [1,6]). However, their application on more realistic scenarios does not provide good results yet due to the following challenges: real-time video stream input, outdoor illumination variations, large amounts of clutter, motion blur, moving cameras, amongst others. Since most of the human activity recognition approaches assume flawless detectors and trackers, their application on more challenging scenarios is even more difficult. Considering these constraints for the application of human activity recognition on real scenarios, we address the following questions in this paper:

* This work was supported by FCT (ISR/IST plurianual funding through the PIDDAC Program) and partially funded by EU Project First-MM (FP7-ICT-248258), EU Project HANDLE (FP7-ICT-231640) and by the project CMU-PT/SIA/0023/2009 under the Carnegie Mellon-Portugal Program.

1. Is it possible to remove the tracking algorithm and find features for activity recognition with good performance, assuming a flawless person detector?
2. If (1) is possible, would the found features work properly in a scenario without detector? In other words, would be feasible to detect people and recognize their activities?

In order to address the questions above, we rely on the state-of-the-art model for human activity recognition: the combination of appearance and motion patterns of each activity [9]. The appearance is encoded by the histogram of image gradients and the motion is encoded by the histogram of the optic flow (dense). In order to learn how to discriminate the patterns we use the popular boosting algorithms, which are efficient, versatile and have shown similar recognition results to more elaborate techniques. Our choice is the L_2Boost algorithm [3], which has two main differences with common boosting methods (e.g. AdaBoost): i) the data points do not have weights to adapt because they are basically included in the gradient computation and ii) the weak learners do not have weighting coefficients because L_2boost uses a fixed step size equal to 1.

We choose the Weizmann dataset for the experiments, originally recorder by [2], because it addresses an interesting multiclass problem that has been virtually solved using the detector plus tracker assumption [7,11]. Thus, the common training and testing steps of the previous works use the the location and size of the people over time, provided by the groundtruth.

In order to address question (1) we use the location and size for each frame separately, so the temporal data association is not considered. We build a spatio-temporal cuboid for each detection independently, so the detected region of interest is projected onto the previous frames. This means that the person may not fully visible on the previous regions of interest. Then, the feature selection procedure searches for the parameters of feature computation that provide very good recognition results and low computational requirements.

In order to answer question (2), we use the features obtained in the previous step and add the "background class" (i.e Nobody performing any activity) to the multi-class problem. Thus, we are able to apply the sliding window method in order to detect people and recognize their activities. In the case of video sequences, the sliding window turns into the sliding cuboid for person and activity detection. The results show that the tracker-less activity recognition is plausible, while the tracker-less and detector-less activity recogntion is a very difficult problem.

2 Human activity model

The state-of-the-art action recognition approaches use a combination of appearance and motion-based features in order to extract the activities' patterns from videos [11]. We follow this approach, using the image gradient and optical flow (dense) as the raw features to extract the action patterns. Figure 1-A and 1-B show an example of the video volume (cuboid) for feature computation. Note

that the person's bounding box at frame I^t maintain the same location over the previous $\tau - 1$ frames, so we do not consider the data association provided by a tracking algorithm. Thus, we use only the person's location at the current frame, making the problem even more complicated, but allowing for an easier development toward the use of moving cameras (for instance mounted on moving robots). The most discriminative and efficient features based on gra-

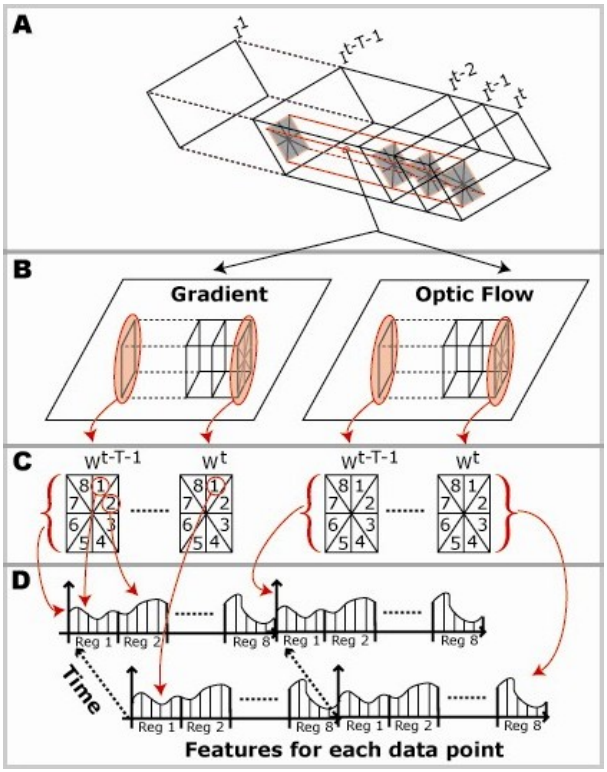

Fig. 1. Feature computation (extracted from [10]): A) example of a volume of video used to compute the features for the person detected in image I^t, B) the two types of raw features used, gradient and flow vectors, computed inside the volume correspondent to the person detected, C) polar sampling used to divide each window into subregions and D) weighted histograms computed for each region, producing a 2D matrix coding the evolution of each bin over a set of T frames.

dients compute weighed histogram of the raw features, such as the histogram of gradients (HOG) [4] and histogram of optic flow [5]. Given a gradient image or optic flow image, the weighed histogram divides the image in subregions (according to a sampling strategy, e.g. Cartesian, polar) and computes the histogram of the gradient (or flow) orientation weighed by its magnitude. Figure 1-C shows the polar sampling strategy. In the case of polar sampling, the his-

togram features are parametrized by the number of subregions (cells) nR and the number of bins nB for each subregion. The correspondent parameters of Cartesian sampling, are the number of intervals the x direction nI_x, the number of intervals in the y direction nI_y and the number of bins nB, which defines $nI_x \times nI_y$ subregions (cells). We denote the gradient histogram as the row vector $g^t \in \mathbb{R}^{nB \cdot nR}$ and $g^t \in \mathbb{R}^{nI_y \cdot nI_x \cdot nB}$ for the polar and Cartesian histograms respectively. Similarly, the flow histograms are denoted as the row vector $o^t \in \mathbb{R}^{nB \cdot nR}$ and $o^t \in \mathbb{R}^{nI_y \cdot nI_x \cdot nB}$, computed at frame t.

At frame I^t and its correspondent rectangular region of interest $R(x_c, y_c, w, h)$ [1], the appearance and motion feature vector for each person detected is

$$h^{t,R} = [g^t o^t] \in \mathbb{R}^{2 \cdot nB \cdot nR} \quad \text{polar sampling} \tag{1}$$

$$h^{t,R} = [g^t o^t] \in \mathbb{R}^{2nI_y \cdot nI_x \cdot nB} \quad \text{cartesian sampling} \tag{2}$$

We consider two ways of modeling the human activity patterns in the spatio-temporal cuboid: (i) the component-wise approach and the (ii) full representation. The component-wise stacks the vector component h_j^t in the previous $t+\tau-1$ frames, so the row feature vector is as follows:

$$X_i^j = \left[h_j^t \ \ldots \ h_j^{t+\tau-1}\right]. \tag{3}$$

The full representation stacks all the h^t vectors in the previous $\tau - 1$ frames,

$$X_i = \left[h^t \ \ldots \ h^{t+\tau-1}\right], \tag{4}$$

where i is the data sample index.

3 L_2Boost with temporal models

The binary L2boost algorithm estimates the function $F : \mathbb{R}^d \to \mathbb{R}$ by minimizing the expected cost $\mathbb{E}\left[C(y, F(X))\right]$ based on the data $(y_i, X_i), i = 1, ..., n$. The cost function is $C(y, f) = (y - f)^2/2$ with $y \in \{-1, 1\}$ and its respective population minimizer is $F(X) = \mathbb{E}\left[y|X = x\right]$. The overall optimization is achieved by means of a sequential stagewise approximation along M rounds, optimizing a so called weak learner in each round, m [3]. The weak learner is the linear combination of the components of the feature vector X_i, so the weak learner of the component-wise model of Eq (3) is $f_m(X_i^j) = X_i^j \beta^m$ and for the full model of Eq. (4) is $f_m(X_i) = X_i \beta^m$.

In order to use matrix notation, we stack all the y_i values into the vector $Y \in \mathbb{R}^N$ and all the X_i data points into the matrix X. In the case of the component-wise model, at each round m, we optimize a temporal model β for each possible feature $j = 1, ..., D$, choosing the one that achieve less error:

$$\widehat{\beta} = \arg\min_{\beta, j} (Y - X^j \beta)^T (Y - X^j \beta). \tag{5}$$

[1] centroid, width and height

The solution is $\widehat{\beta}^m = ({X^{j^m}}^T X^{j^m})^{-1} {X^{j^m}}^T Y$, where j^m is the component that achieves less error. In the case of the full model of Eq. (4), the feature index j is removed from Eq. (5), so $\widehat{\beta} = \arg\min_\beta (Y - X\beta)^T (Y - X\beta)$, whose solution is $\widehat{\beta}^m = (X^T X)^{-1} X^T Y$. The component-wise L_2boosting algorithm with linear temporal models of Eq. (3) is as follows:

1. **Initialization** Chose M and set m=0. Given data (Y, X), fit the first weak learner, $\widehat{F}_0 = X^{j^0} \widehat{\beta}^0$. β^0 and j^0 are computed from Eq. (5).
2. **Projection of gradient to learner** Compute the negative gradient (in this case are the residuals) $u_i^{m+1} = y_i - \hat{F}_m(X_i)(i = 1, ..., n)$. For simplicity, stack all u_i values into the vector $U \in \mathbb{R}^N$.
   Use the residuals U^{m+1} to fit the learner $\widehat{f}_{m+1} = X^{j^{m+1}} \widehat{\beta}^{m+1}$ changing Y for U in Eq. (5).
   Update $\tilde{F}_{m+1} = \widehat{F}_m + \widehat{f}_{m+1}$. Compute $\widehat{F}_{m+1} = sign(\tilde{F}_{m+1}) \min(1, |\tilde{F}_{m+1}|)$.
3. **Iteration** If $m + 1 < M$ increase m by 1 and goto step2.
   If $m + 1 = M$ return $\Theta_j = \{j^0, ..., j^m, ...\}$, and one set of models, $\Theta_\beta = \{\beta^0, ..., \beta^m, ...\}$

The classification of a new point X_i is given by the sign of the strong classifier result, $\operatorname{sgn} \widehat{F}_M(X_i)$. Notice that the last computation of step 2 constraints the strong classifier to be in $[-1\ 1]$, so we apply the L_2Boost with constraints [3], which works better in the classification setup. The algorithm just presented is very similar to the full model one, but removing the feature index j. The strong classifier $F(x)$ relates the class-conditional probabilities,

$$F(x) = 2p(y = 1|x) - 1, |F(x)| = |p(y = 1|x) - p(y = -1|x)|, \tag{6}$$

and its module $|F(X_i)|$ is the classification margin, that is the probability of labeling the new data point given the models estimated. In order to extend the L_2Boost with linear-temporal models to multi-class problems we use the one vs. all approach, which solves C binary problems to discriminate between C classes where $Y \in \{1, \ldots, C\}$. The multi-class version of L_2 starts by computing $\widehat{F}_M^{(c)}$ on the basis of the binary response variables

$$Y_i^{(c)} = \begin{cases} 1 & \text{if } Y_i = c \\ -1 & \text{if } Y_i \neq c \end{cases} i = 1, \ldots, n \tag{7}$$

and then builds the classifier as $\widehat{C}^m(x) = \arg\max_{c \in \{1,...,C\}} \widehat{F}_M^{(c)}(x)$.

4 Feature selection for tracker-less recognition

We address this problem by comparing the recognition rate between different types of features in the Weizmann dataset [2], which contains 9 subjects performing 9 actions: {1 - bending down, 2 - jumping jack, 3 - jumping, 4 - jumping in place, 5 - running, 6 - galloping sideways, 7 - walking, 8 - waving one hand,

9 - waving both hands}. We follow the evaluation protocol proposed by [2] that performs a leave-one-out test with the 9 subjects, so each subject belongs to one of the testing sets. Then, the confusion matrix is averaged over all the leave-one-out test sets and the trace of the averaged matrix is used as the measure of recognition performance.

We consider the following options to select the feature computation method: (i) component-wise vs. full model , (ii) cartesian and polar cell sampling, (iii)number of frames τ of the linear temporal model, (iv)optic flow algorithm, (v) two options for the region of interest in the image (detected bounding box) and (vi) cell overlapping. We observe in Table 1 that the component-wise L_2Boost performs

	Average confusion matrix's trace (%)		
Feature type	component-wise	all features	dims
polar $nR = 8$, $nB = 16$	91,29	89,76	256
polar $nR = 16$, $nB = 16$	**95,42**	93,2	512
cartesian $nI_x = 4$, $nI_y = 8$, $nB = 16$	**95,42**	93,2	512
cartesian $nI_x = 3$, $nI_y = 6$, $nB = 16$	95,46	92,79	576

Table 1. Component-wise vs. full model results, using two sets of parameters for each sampling approach. ($\tau = 10$, no overlapping between cells and using groundtruth detections), Ogale's optic flow [8]

better than the full model one, so in the rest of the experiments we just consider the component-wise approach. In addition,we select the polar and cartesian sampling that attain the top recognition result, $nR = 16$, $nB = 16$ for polar and $nI_x = 4$, $nI_y = 8$ for cartesian. The next step is to compare the effect of the optic flow algorithm in the classification results. Table 2 shows that Ogale's

	Average confusion matrix's trace (%)	
Feature type	Ogale et. al. [8]	Werlberger et. al. [12]
polar $nR = 16$, $nB = 16$	**95,42**	94,11
cartesian $nI_x = 4$, $nI_y = 8$, $nB = 16$	**96,01**	94,9

Table 2. Effect of two optic flow approaches on the recognition rate ($\tau = 10$, no overlapping between cells and using groundtruth detections)

τ	1	3	5	7	10	13	15
polar $nR = 16$, $nB = 16$	86,2	90,36	92,64	93,27	**94,11**	93,55	93,47
cartesian $nI_x = 4$, $nI_y = 8$, $nB = 16$	87,88	91,7	92,35	94,1	**94,9**	94,36	94,11

Table 3. Temporal support comparison. (no overlapping between cells and using groundtruth detections)

et. al. [8] algorithm has a better performance than Werlberger's one. In this case our choice is the Werlberger's algorithm because of the GP/GPU implementation that allows to compute the optic flow (dense) in near real-time for normal cameras. The reason behind this choice is the quicker evaluation of our approach on other datasets (e.g. [10]), and the near real-time plausibility of [12], which facilitates future deployment of the system. The temporal support used in the previous test ($\tau = 10$) was motivated by Schindler et. al. [11]. Table 3 re-validates their choice $\tau = 10$. In the following we compare the groundtruth boxes against a manually set bounding box for all the detections. We define a bounding box with constant width/height ratio in order to select the spatio-temporal cuboids. The rationale of this fixed ratio bounding box is two-folded: (i) facilitate the application of the sliding window method and (ii) allow the search over multiple scales. Table 4 shows that the selected w/h is practically equal to the groundtruth boxes, because the persons of the Weizmann dataset have similar sizes. Finally, we apply the idea of overlapping between cells [4].

	Average confusion matrix's trace (%)	
Feature type	groundtruth ROI [2]	Fixed size ROI $w = 60$, $w/h = 0.779$
polar $nR = 16$, $nB = 16$	94,11	**94,84**
cartesian $nI_x = 4$, $nI_y = 8$, $nB = 16$	94,9	**95,56**

Table 4. Region of interest comparison. Groundtruth boxes vs. manually selected ones. (no overlapping between cells)

In the case of polar sampling, we add more cells to the previous ones in such a way that each new cell overlaps with two of the original neighboring cells in equal proportion. In the case of the cartesian sampling, each new cell overlaps with four of the original neighboring cells in equal proportions. Table 5 shows that cell overlapping and cartesian sampling brings better results, but at the expense of a larger computational load. Since we are interested in features having a lower computational load and good performance, we choose the polar sampling with no overlap. Summarizing, the feature selection options are: (i) component-wise L_2Boost, (ii) Welberger's optic flow [12], (iii) $\tau = 10$, (iv) fixed w/h ratio bounding boxes and (v) no overlap polar sampling cells.

	Average confusion matrix's trace % (dimensions)	
Feature type	half interval overlap	no overlap
polar $nR = 16$, $nB = 16$	95,15 (1024)	**94,84** (512)
cartesian $nI_x = 4$, $nI_y = 8$, $nB = 16$	95,68 (1696)	**95,56** (1024)

Table 5. Comparison between cells with and without overlap.

4.1 Tracker-less and detection-less scenario

The features found above attain very good recognition rates in a tracker-less scenario. In this section we want to evaluate their performance on a tracker-less and detector-less scenario. Thus, we need to add the background activity class (i.e. spatio-temporal cuboids where no person is doing any action) to the activity classes in order to both detect people and recognize their activities. We obtain the background samples by the random selection of video segments in the Weizmann dataset. Then, we compute the features selected in the previous section and re-train the L_2Boost algorithm with the 9 activities plus the "background" activity.

The testing phase comprises the application of the volume-based version of the sliding window algorithm. This image-based algorithm is applied on pedestrian detection, by moving the region of interest (window) along the image grid. For each grid point, the image features are computed inside the window, followed by the binary classification (person or background). We perform the volumetric version of the algorithm, by moving the region of interest (cuboid) along the video (3D) grid. Then, we classify each cuboid as a particular human activity or the "background" activity. We sample the video grid every 5 pixels in each image direction and every 2 frames in the temporal direction. The trace of the confusion matrix for the 10 classes is $94,74\%$. This looks like a good result because of the larger number of background samples compared to the human activity samples. After removing the background samples the trace of the confusion matrix is $30,04\%$. This result is explained by the absence of the perfectly aligned detection results provided by the groundtruth. These misalignments of the cuboids in the video were not learnt during training, so the L_2Boost is not able to discriminate between the human activities.

5 Conclusions

We address the feature selection for human activity recognition in a tracker-less scenario. We construct features that encode appearance and motion by means of the Histogram Of Gradients (HOG) [4] and the Histogram Of Flow (HOF) [5] over several video frames. Our choice of learning approach, the L_2Boost, it finds the linear models for binary problems and we apply the one vs. all approach for the final classification.

In this feature-classifier context, we select experimentally the parameters that: (i) attain very good results and (ii) have low computational requirements. In addition, we evaluate the selected features in a tracker-less and detector-less scenario, a very challenging problem due to the large appearance variation in the background and the reduced amount of motion information contained in it. Future work must study the combination of features from both worlds: human activity recognition and pedestrian detection in order to have features that do not assume flawless person trackers and detectors.

References

1. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: IEEE CVPR 2008. pp. 1–8 (2008)
2. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: IEEE ICCV 2005. vol. 2, pp. 1395–1402 Vol. 2 (Oct 2005)
3. Buhlmann, P., Yu, B.: Boosting with the l2 loss: Regression and classification. Journal of the American Statistical Association 98, 324–339 (January 2003)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the CVPR '05. pp. 886–893. Washington, DC, USA (2005)
5. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: European Conference on Computer Vision (2006)
6. Gerónimo, D., López, A., Sappa, A., Graf, T.: Survey of pedestrian detection for advanced driver assistance systems. IEEE PAMI 32(7), 1239 –1258 (july 2010)
7. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A Biologically Inspired System for Action Recognition. In: Proceedings ICCV. pp. 1–8 (October 2007)
8. Ogale, A.S., Aloimonos, Y.: A roadmap to the integration of early visual modules. International Journal of Computer Vision 72(1), 9–25 (april 2007)
9. Poppe, R.: A survey on vision-based human action recognition. Image and Vision Computing 28(6), 976 – 990 (2010)
10. Ribeiro, P.C., Moreno, P., Santos-Victor, J.: Unsupervised and online update of boosted temporal models: the UAL_2boost. In: Proc. of ICMLA (December 2010)
11. Schindler, K., van Gool, L.: Action snippets: How many frames does human action recognition require? In: IEEE CVPR 2008. pp. 1–8 (June 2008)
12. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. In: Proc. of BMVC (September 2009)