A Dynamic Approach and a New Dataset for Hand-detection in First Person Vision

Alejandro Betancourt[1,2(✉)], Pietro Morerio[1], Emilia I. Barakova[2], Lucio Marcenaro[1], Matthias Rauterberg[2], and Carlo S. Regazzoni[1]

[1] Department of Naval, Electric, Electronic and Telecommunications Engineering, University of Genoa, Genoa, Italy
a.betancourt@tue.nl
[2] Designed Intelligence Group, Department of Industrial Design, Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract. Hand detection and segmentation methods stand as two of the most most prominent objectives in First Person Vision. Their popularity is mainly explained by the importance of a reliable detection and location of the hands to develop human-machine interfaces for emergent wearable cameras. Current developments have been focused on hand segmentation problems, implicitly assuming that hands are always in the field of view of the user. Existing methods are commonly presented with new datasets. However, given their implicit assumption, none of them ensure a proper composition of frames with and without hands, as the *hand-detection* problem requires. This paper presents a new dataset for hand-detection, carefully designed to guarantee a good balance between positive and negative frames, as well as challenging conditions such as illumination changes, hand occlusions and realistic locations. Additionally, this paper extends a state-of-the-art method using a dynamic filter to improve its detection rate. The improved performance is proposed as a baseline to be used with the dataset.

1 Introduction

Videos recorded from head-mounted cameras are becoming popular due to the increasing availability of wearable devices such as smart glasses and action cameras. The idea of a wearable computer recording what the user is looking at, and giving him relevant feedback and assistance is nowadays technically possible. As expected, this emerging technology is increasingly capturing the interest of computer scientists and software developers to create methods to process videos recorded with head or chest mounted cameras. This video perspective is commonly referred as First Person Vision (FPV) or Egocentric-vision [9]. In fact, FPV video analysis is not a new research field. It is possible to state that modern devices are highly influenced by the academic research of the late 1990s [29].

This work was supported in part by the Erasmus Mundus joint Doctorate in Interactive and Cognitive Environments, which is funded by the EACEA, Agency of the European Commission under EMJD ICE.

© Springer International Publishing Switzerland 2015
G. Azzopardi and N. Petkov (Eds.): CAIP 2015, Part I, LNCS 9256, pp. 274–287, 2015.
DOI: 10.1007/978-3-319-23192-1_23

Existing literature points out several promising applications of this video perspective. Among them, hand-based methods stand as the most explored ones, aiming to exploit the conscious or unconscious hands movements for performing higher inference about the user [7] as in activity recognition [14,23] and user-machine interaction [27]. A common practice in FPV is to assume that hands are always recorded by the camera and, as a consequence, they can be located and tracked to infer more complex information. As it can be concluded after a quick scan of uncontrolled datasets like Disney [13] or UTE [15], this assumption is not entirely true. In fact, the predominance of one or the other type of frames (with/without hands) in a video sequence is not a consequence of the advantageous camera location but also of the activity performed e.g. hands are more frequent when the user is cooking than when he is walking in the street.

Despite the practical advantages of assuming full time hands presence, this fact introduces important issues when the proposed methods are applied on uncontrolled videos, for example wasted computational resources or noisy signals in the hand-segmentation stage, that could be propagated to other levels of the system. The authors in [6] propose a characterization of the two distinct problems, namely *hand-detection* and *hand-segmentation*, and combine them in a sequential structure to improve the overall system performance. Following the definition of [6], the *hand-detection* level answers the yes-or-no question of the hands' presence in the frame using global features and classifiers, while the *hand-segmentation* level locates and outlines the hands' region in a positive frame using low level features like color under an exhaustive pixel by pixel classification framework [19,21,27].

Regarding data availability, there are several FPV datasets available for research purposes. In general the technical characteristics of these datasets are similar and the videos are carefully recorded to guarantee the basic requirements identified by Schiele in 1999 [26]: i) Scale and texture variations, ii) Frame resolution, iii) Motion blur and iv) Hand occlusions. Undoubtedly, these requirements are important, but, under the light of the recent technological trends, some extra characteristics must be taken into account. An example is the necessity of balanced datasets in terms of hands presence as described by [6] and [8], to face the *hand-detection* problem under a classification framework. A balanced dataset is a realistic assumption for wearable devices and could lead to important improvements in the battery life, as well to the performance of higher-level methods like hand-based activity-recognition[12] and user-machine interaction [27]. It is worth to mentions that, as shown in section 2, existing datasets does not guarantee this condition, which makes them inappropriate to face the classification problem of the *hand-detection* level.

This work focuses indeed on *hand-detection*, and its contributions are threefolded: i) It presents the UNIGE-HANDS dataset for *hand-detection*, which guarantees a balanced number of frames with and without hands in 5 realistic locations, as well as changes in illumination, camera motion and hands occlusions. [1] ii) Multiple *hand-detectors* (feature-classifier) are evaluated over the dataset,

[1] [Dataset:] http://www.isip40.it/resources/UNIGEhands

following [6], without considering the temporal dimension of the data. iii) The best *hand-detector* (HOG-SVM) is extended using a Dynamic Bayesian Network (DBN), which is tuned to smooth the decision process. The presented method improves the performance of [6], taking advantage of the temporal dimension of the video, and of [8], tuning the parameters through an heuristic optimization. The computational complexity of the proposed approach is taken into account by filtering the classification certainty of the SVM directly, instead of a generic *multidimensional* array of features. Namely, we perform the filtering step at a higher hierarchical level in the estimation process as depicted in Figure 1.

The remainder of this paper is organized as follows: Section 2 summarizes the evolution of *hand-detection and segmentation* methods and shows why the existent datasets are not suitable to solve the *hand-detection* problem. Section 3, presents the UNIGE-HANDS dataset and evaluates multiple frame by frame *hand-detectors* (combinations of image features and classifiers). Later, section 4 extends the state-of-the-art method using a DBN and briefly describes each of its components. Section 5 tunes the DBN using a classic Genetic Algorithm (GA) and the Nelder-Mead simplex (NM) algorithm in a cooperative fashion. Subsequently, the performance of the DBN is evaluated, and under the light of the results, the challenges offered by the UNIGE-HANDS dataset are presented. Finally, in section 6 conclusions are drawn and some lines for future research are proposed.

2 State of the Art

In the recent years, thanks to the growing availability of FPV recording devices, the number of methods to process related videos, as well as datasets, has increased quickly. To the best of our knowledge a total of 16 datasets have been published between 2005 and 2014, each of them especially designed to face a particular objective, i.e. Object recognition and tracking, activity recognition, computer machine interaction, video summarization, physical scene reconstruction, and interaction detection. Table 1 summarizes the existent datasets and their basic characteristics. The table also highlights the evolution of the camera location, moving from shoulder, to

Table 1. Current datasets and sensors availability [9].

				# Objects			C. Location		
		Year	Objective	Activities	Objects	Num. of People	Shoulder	Chest	Head
Mayol05	[20]	2005	O1		5	1	✓		
Intel	[23]	2009	O1		42	2	✓		
Kitchen.	[28]	2009	O2	3		18			✓
GTEA11	[12]	2011	O2	7		4			✓
VINST	[2]	2011	O2			1		✓	
UEC Dataset	[16]	2011	O2	29		1			✓
ADL	[24]	2012	O2	18		20		✓	
UTE	[15]	2012	O4		4			✓	
Disney	[13]	2012	O6			8			✓
GTEA gaze	[14]	2012	O2	7		10			✓
EDSH	[18]	2013	O1	-	-	-			✓
JPL	[25]	2013	O6	7		1			✓
Virtual Museum	[27]	2013	O3	5		1			✓
BEOID	[11]	2014	O2	6		5			✓
EGO-GROUP	[4]	2014	O6			19			✓
EGO-HPE	[3]	2014	O1			4			✓

* **Objectives:** [O1] Object Recognition and Tracking. [O2] Activity Recognition. [O3] User-Machine Interaction. [O4] Video Summarization. [O5] Phisical Scene Reconstruction. [O6] Interaction Detection.

head-mounted. This trend can be explained by the interest of technology companies to develop smart glasses and action cameras.

Existing datasets can be divided in two main groups: datasets where hands are almost always present, and datasets where hands barely appear. The first group has been used for object recognition (Mayol05, Intel), activity recognition (Kitchen, GTEA11, GTEA12) and user-machine interaction (Virtual-Museum). These datasets are usually recorded in fixed locations, like a kitchen or the office, while the user performs different tasks. Regarding the *hand-detection* problem, these datasets are not suitable because it is not possible to extract a set of negative samples in the same location and light conditions as the positive ones to train binary classifiers. The second group of datasets are frequently used for activity recognition (VINST, UEC, ADL), video segmentation (UTE, BEOID), Interaction Detection (Disney, JPL, Bristol, EGO-GROUP, EGO-HPE). In general these datasets are large and contain sequences of the user moving through several realistic locations. The number of frames with hands is low compared with the length of the videos, and the locations with frames with hands are sparse, making impossible to extract a large enough balanced training set with similar locations. It is worth to highlight the importance of having frames with and without hands in the same location. This would lead the classifiers to learn patterns related with the hands presence and not from the changes in the location.

According to [20], known for being the first public dataset in FPV for object recognition, *hand-detection/segmentation* methods can be grouped in two: model-driven and data-driven. The former uses a computerized model of the hands to recreate the image of the videos [30], while the latter exploit image features to infer about hand location, shape and position [19,21,27].

Regarding *hand-detection*, a data-driven sequential classifier is proposed in [6], which in a first stage detects hands, and in a second stage finds the hands silhouette at a pixel level *only for positive frames*. In their experiments, the authors report the performance of multiple classifiers and image features, to finally conclude that the best-performing combination is HOG plus SVM achieving 90% of true-positives and 93% of true-negatives. The authors in [31] follow a color-based approach in the same line of [19] which, as is shown in [6], could introduce noise in the results under large illumination changes. To conclude the overview, [17] proposes a probabilistic approach to detect if the hands in the video belongs to the user or to another person.

3 UNIGE-HANDS: Hand-detection Dataset

The UNIGE-HANDS dataset for *hand-detection* is a set of FPV videos, carefully recorded to guarantee a good balance between frames with hands and without hands, and offers challenging characteristics such as changes in illumination, camera motion and hand occlusions. The UNIGE-HANDS dataset, videos and ground truth, is distributed for public use. The dataset contains videos recorded in 5 uncontrolled locations (1. Office, 2. Coffee Bar, 3. Kitchen, 4. Bench, 5. Street). Each location in the dataset is in turn divided in training and testing videos. Table 2 shows some examples of the frames in each location.

Table 2. Examples of the dataset frames.

		Office	Street	Bench	Kitchen	Coffe Bar
Training	Hands					
	No Hands					
Testing	Hands					
	No Hands					

To record the dataset we used a *GoPro hero3+* head mounted camera with a resolution of 1280×720 *pixels* and 50 *fps*. The whole dataset, including training and testing videos, contains one-hour and thirty eight minutes of video. In total, the training videos have 37.21 and 37.63 minutes of positives and negative sequences, respectively. The training videos for each location are formed by 2 positives and 2 negatives videos approximately 3.34 minute-long each (10020 frames). Regarding the testing videos, they comprise 12.6 minutes of positive and 12.7 minutes of negative segments. The testing video of each location lasts approximately 4 minutes (12000 frames), changing from positive to negative in intervals of about one minute.

Following the procedure described in [6], multiple combinations of classifiers and video features are evaluated over the new dataset. The classifiers are: Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF). The video features are: Histogram of Oriented Gradients (HOG), the global scene descriptor GIST, three color spaces (RGB, HSV, LAB) and its concatenation (RHL). The SVM uses a linear kernel with a regularization parameter $C = 1$. To compute the features, each frame is compressed to 200×112 px. The HOG extractor uses a block size of $16px$, a cell size of $8px$, and 9 directional bins, while color features are estimated over a grid of 25×14 cells (which are indeed $8 \times 8px$ cells).

Table 3. Performance of the proposed *hand-detectors*.

		True Positives			True Negatives		
		SVM	DT	RF	SVM	DT	RF
10-fold	HOG	**0.89**	0.77	0.81	**0.90**	0.76	0.88
	GIST	0.78	0.75	0.72	0.79	0.74	0.88
	RGB	0.77	0.72	0.73	0.77	0.73	0.86
	HSV	0.72	0.76	0.78	0.72	0.78	0.88
	LAB	0.75	0.85	**0.89**	0.75	0.85	**0.90**
	RHL[1]	0.78	0.85	0.86	0.77	0.85	0.91
Training	HOG	**0.93**	0.80	0.83	**0.91**	0.80	0.91
	GIST	0.83	0.81	0.80	0.82	0.80	0.91
	RGB	0.82	0.76	0.78	0.82	0.78	0.90
	HSV	0.77	0.80	0.83	0.78	0.82	0.92
	LAB	0.80	0.88	**0.92**	0.79	0.88	**0.93**
	RHL[1]	0.81	0.87	0.88	0.81	0.87	0.93
Testing	HOG	**0.76**	0.72	0.70	**0.84**	0.75	0.83
	GIST	0.51	0.51	0.43	0.67	0.58	0.70
	RGB	0.57	0.60	0.57	0.72	0.64	0.68
	HSV	0.60	0.65	0.65	0.66	0.67	0.75
	LAB	0.56	0.75	**0.74**	0.69	0.73	**0.77**
	RHL[1]	0.57	0.74	0.71	0.68	0.71	0.78

1 RHL is the concatenation of RGB, HSV and LAB.

Table 3 reports the performance of each feature-classifier combination under three different evaluation strategies: i) *Cross-validation:* 10-fold validation performed using the training frames as described in [6]. This procedure requires to train each classifier 10 times using 90% of the sampled frames for training and 10% for testing. The reported performances are computed using as training data 2203 frames with hands and 2233 without hands. These frames are gathered by sampling the training videos once every second. ii) *Frame by frame in the training videos:* The classifier is trained using the sampled frames, and tested in the remaining frames of the training videos. This approach only requires to train the classifiers once, which is particularly useful for the tuning procedure explained in section 4. iii) *Frame by frame in the testing videos:* The classifier is trained in the sampled frames but tested in the testing videos. This approach is the more realistic to test the classifier because, despite being recorded in the same locations, the testing videos are completely independent of the training stage.

The first finding in the table is that the performance reported in the 10-fold is slightly lower that the reported by the authors in the original paper. This reduction is explained by the challenges intentionally introduced in the dataset, namely the illumination changes and the number of locations. The 10-fold performance validates the conclusion of [6], where HOG-SVM stands as the best performing combination, although here the LAB-RF achieve a similar performance. In general the first (10-fold) and second group (Training) of performances are similar, which validates the use of the second strategy to tune the DBN in a computationally efficient way. To evaluate the performances in a dynamic perspective (video sequences), each frame of the testing videos is classified using the already trained *hand-detectors*. In general, these performances are lower than the first and second group, showing the importance of the testing videos. The optimistic performance reported by the cross-validation method is extensively explained in the literature and is known as the bias in the cross validation procedure [5].

It is worth to note that HOG-SVM is the best performing combination in all the evaluation strategies, particularly in the third one (*testing videos*), where it achieves 76% of true-positives and 84% of true-negatives. Noteworthy is also the performance of LAB-RF, which despite of being lower than HOG-SVM in the testing case, could offer important cues for to improve computational efficiency of the hand-detector. In addition to the outstanding classification rate, the HOG-SVM combination shows an extra advantage, given by its theoretical formulation, which naturally provides could provide a real valued confidence measurement of hands presence. The latter is particularly important in the dynamic approach as explained in the next section. The remainder of this paper is focused on the HOG-SVM detector and the dynamic strategy to improve its results.

4 Hand-detection DBN

In this section, a SVM-based detector is extended with dynamic information using the DBN proposed in Figure 1. The figure sketches a multilevel Bayesian filter for state estimation where the bottom level contains the raw images and the upper level the filtered decision. In general, the measurement (z_k) is a real valued representation of the SVM classifier applied to set of features F_k extracted from the k^{th} frame I_k. The state $x_k \in R^2$ is the filtered SVM confidence enriched with its speed: $x_k = [f(F_k), \dot{f}(F_k)]$. Finally, h_k is the binary decision based on the filtered value of the state: $h_k = sign(x_k[0] + t_h)$. The latter allows t_h to take values different from 0, in order to capture the effects of the dynamic filter to the decision threshold of the SVM. The dotted line of Figure 1 is drawn to illustrate the possible filtering at features level, as discussed in section 1. However, in our case only the state of the system is filtered. The remaining part of this section briefly introduces the SVM notation, the dynamic filtering, and the heuristic tuning of the DBN parameters. See [8] for extra details about the mathematical formulation of the SVM and the dynamic filter.

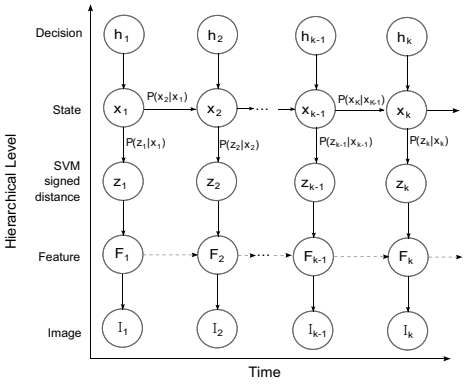

Fig. 1. Dynamic Bayesian Network for smoothing the decision process.

i) Support Vector Machine: Let's assume a dataset composed by N pairs of training data: $(F_1, y_1), (F_2, y_2), \ldots, (F_N, y_N)$, with $F_i \in R^p$ and $y_i \in \{-1, 1\}$. Equation (1) defines a classification hyperplane and equation (2) its induced classification rule, where β is a unit vector. Assuming that the classes are not separable then the values of β and β_0 are the solution of the optimization problem given by (3), where $\xi = (\xi_1, \xi_2, \ldots, \xi_N)$ are referred to as the slack variables, and K is constant.

$$\{F : f(F) = F^T\beta + \beta_0 = 0\} \tag{1}$$

$$G(F) = sign(f(F)) = sign(F^T\beta + \beta_0) \tag{2}$$

$$\min_{\beta,\beta_0} ||\beta|| \text{ subject to: } y_i(F_i^T\beta + \beta_0) \geq 1 - \xi_i, \forall i, \quad \xi_i \geq 0, \sum \xi_i \leq K \tag{3}$$

For the *hand-detection* problem we use the signed distance to the classification hyperplane, $f(F_k)$, as the measurement ($f(F_k)$ is denoted as z_k in the DBN diagram, using the common notation for measurements in Bayesian filtering), where F_k is a global feature extracted from the k-th frame. It is important to note that the signed distance to the decision boundary $f(F)$ gives both a

description of the result $G(F)$ of the classification (i.e. $sign(f(F))$) as well as its level of certainty. In addition, augmenting the state with the speed ($\dot{f}(F)$) would allows us to control sudden variations of such confidence. In some sense the DBN is thus self-aware of how good the classification is evolving, and can introduce some feedback mechanism to compensate for poor classification.

ii) Kalman Filter: Once the certainty level from the SVM is extracted, we address the problem of transferring and stabilizing that measurement from time to time. This strategy aims to reduce the number of wrong decisions caused by little variations in the features between frames. For this purpose we use a discrete linear Kalman filter. In general notation, the process and measurement model is given by (4), where $x_k \in \mathbb{R}^n$ is the state and $z_k \in \mathbb{R}^m$ is the measurement. The matrix $A_{n\times n}$ relates the state at previous step, x_{k-1}, with the state at current step, x_k. The matrix $H_{m\times n}$ relates the state with the measurement. Finally, w and v are the process and measurement noise respectively, which are assumed Gaussian with zero mean and covariances $Q_{n\times n}$ and $R_{m\times m}$ respectively. In our case $n = 2$ and $m = 1$, x_k is then a two dimensional vector, whose first component contains the decision certainty and the second its changing speed. At this point the binary decision, h_k, is calculated using $sign(x_0 + t_h)$, which as already mentioned, is equivalent to allow changes in the original SVM decision threshold.

$$x_k = Ax_{k-1} + w_k, \qquad z_k = Hx_k + v_k \tag{4}$$

iii) Tuning the DBN: Within the general framework presented above, there are two sets of parameters to be estimated. The first set are the parameters defining the classification hyperplane of the SVM, namely β and β_0. These parameters are estimated using the training dataset and the SVM implementation of sklearn library [22] for python . The second set are the Kalman filter parameters and the decision threshold, namely Q, R and t_h. The tuning of the parameters of a dynamic filter is a widely explored field, and different approaches are usually followed according to the requirements of the system, restrictions in the measurements, and the ground truth availability.

Following the work of [1] the main idea behind the tuning procedure is to decompose the joint distribution of the system $p(z_{0:T}, x_{0:T}, h_{0:T})$, using the Bayesian notation, and, given the data availability and characteristics of the marginal distributions, find the optimal values of the parameters. In our case the more appropriated approach, taking advantage of the ground truth, and given the non-differentiability the binary decision boundary, is to minimize the residual prediction error in an heuristic way. With this in mind we look to minimize the squared error of the DBN decisions, defining the optimization problem as (5).

$$< Q, R, t_h >= \underset{Q,R,t_h}{\arg\min} \sum_{k=0}^{T} (h_k - \hat{h}_k)^2 \tag{5}$$

This optimization problem is usually faced using a method like the Nelder-Mead simplex (NM) algorithm to find a optimal solution close to an initial

solution. Under the absence of intuition about the initial point, the authors in [10] suggest to use a combination of a basic Genetic Algorithm (GA), to find some initial points, and later improve them using NM. In our case we design a classical GA where each genome is an instance of the parameters to be optimized, and each generation contains 100 genomes. The algorithm starts with an initial population of 100 random genomes to select the best 4, named parents. The subsequent generation is then composed by two parts. The first 64 genomes are crossovers: combinations of the parents, and the remaining 36 genomes are mutations: random modifications of the parents. In the mutation stage, the parents are selected randomly, and each element is modified with a probability of 0.5. Once the algorithm achieved an acceptable decaying rate of the objective function, the 4 best genomes among all the generations are used as initial points in NM. The best of the NM results is selected as the optimal combination.

5 Results

The results presented in this section are two-fold. First, we introduce two different optimization cases for the proposed filter. Second, we show how the DBN approach considerably improves the performance of the naive HOG-SVM detector (detailed results are presented for the best optimization problem only, but they enhancement is significant even in the worst case).

The Kalman filter is formulated as a kinematic model of the "position" (distance to the separation hyperplane) enriched with the speed, and a sampling rate Δ_t. Equation (6) shows the process and measurement model, where $w_k \sim \mathcal{N}(0, Q)$ and $v_k \sim \mathcal{N}(0, r)$. There is not exact knowledge of the differential equation regulating the dynamic process, thus it is not possible to precisely state the law that moves the decision back and forth the decision boundary. Actually, it is not known if such differential equation exists or can be solved in closed form. For this reason, we borrow from physics a constant force model, which we think is a good starting point. This is equivalent to suppose there is some constant (oscillating) force that keeps the features away from the decision hyper-surface or make them cross it, with a constant acceleration a.

$$\begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{bmatrix} + w_k, \text{ and } z_k = [1, 0] \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + v_k \tag{6}$$

More in detail, the first equation in (6) models an exact constant acceleration, where a is the *effect* of a control input which generates exactly the time-dependent noise term. On the other hand, employing a state augmented with the second derivative as well, would allow small variations of a, accounted for in the noise term w_k. In our optimization framework, this is equivalent to parametrize each of the elements of Q. In this case the genomes are given by instances of $[Q_{1,1}, Q_{1,2}, Q_{2,1}, Q_{2,2}, r, t_h]$, and the elements of each crossover are selected randomly from one of the current parents. In the second optimization case, we suppose instead that the acceleration is constant, and the matrix Q is factorized isolating the sampling rate as in (7). In this case the genomes are

of the form $[q, r, t_h]$ and the crossovers are all the possible combinations of the current parents. To keep control of the search space we bound the elements of Q as well as q and r to move between 0 and 1000. The decision criteria t_h is bounded between -0.5 and 0.5. The number of iterations is set to 20. To evaluate the objective function for each combination we merge the testing videos and calculate the overall accuracy under the second strategy of Table 3. We point out that the second strategy is used because of computational advantages and to keep the training and tuning process independent of the testing videos.

$$Q = q * \begin{bmatrix} \frac{\Delta_t^4}{4} & \frac{\Delta_t^3}{2} \\ \frac{\Delta_t^3}{2} & \Delta_t^2 \end{bmatrix} \tag{7}$$

From the tuning process of the two cases presented above we found that the best accuracy is achieved for the genome $[+1.15e^{-9}, +1.39e^{-7}, +8.72e^{-8}, +2.07e^{-5}, +60.78, \ -7.63e^{-2}]$ and $[+0.039, +32.54, -0.151]$ for the general and factorized case respectively. The final number of frames misclassified by each case are 3505 and 3391 over a total of 220610. As a comparison, the total of misclassified frames using naive HOG-SVM is 18211. It is remarkable the fact that both optimization scenarios reach a similar value in the objective function, validating the use of the constant acceleration model to reduce the flickering in the decision. The remaining of this section present more in detail the results achieved by the factorized case over the testing videos. Figure 2 shows, in red line, the measurement z_k and, in blue line, the filtered state x_k. The horizontal axis is the decision threshold. Taking the value of 4, 5, 6 (-4, -5, -6) the figure shows the ground truth, the decision of the HOG-SVM method and DBN, respectively. These decisions takes positive values if there are hands and negative if not. The noisy movements of z_k confirm the dependence of the measurement to little changes between frames. As it is intended, the Kalman filter reduces the noise and preserve the trend of z_k.

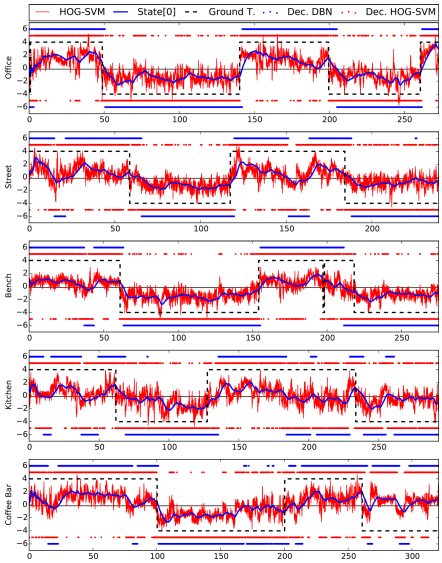

Fig. 2. Performance of the DBN in each of the locations in the UNIGE-HANDS . dataset.

It can be noted from the pointwise decisions of HOG-SVM (Dec. HOG-SVM) that it is difficult to obtain continuous segments of the video with or without hands. This effect is the consequence of the measurement noise changing frequently the sign of z_k. Once the noise is reduced using the DBN, the decisions stabilizes and continuous segments appear. It is particularly remarkable the

performance of the DBN in the *Office* and the *Bench* sequences. However, because of the poor performance of the HOG-SVM, the DBN misclassifies long segments in the *Kitchen* and the *Coffee bar* sequences. The poor performance of the HOG-SVM in these sequences can be explained by the 3D perspective created by the table, which creates lines in the same positions and directions of those created by the hands.

Table 4 summarizes the performance for each location of the dataset. In total the DBN improves the number of true-positives by 5.6 percentage points, moving from 76.4% to 82.0%. The number of true-negatives is improved by 2.7 percentage points, changing from 83.7% to 86.4%. The only performance which suffer a reduction is the true-positives of the *Kitchen*. This reduction is explained by a long segment (Figure 2 between second 150 and 250) in which the measurements are switching between positive and negative values with no trend. An extra analysis of the corresponding video validates the hypothesis of the 3D perspective created by the used table, and points out an interesting research idea regarding the fusion of color and shape features to deal with this kind of scenarios. A similar case is found in the last segment of the *Coffee Bar* location, which despite showing an improvement of 0.7 percentage points in the true-negatives, is one of the worst performing. In all the other scenarios the improvement is remarkable. Particularly, the true-positives of the *Bench* location is the one with the largest improvement (11.7 percentage points). The improvement in the true-positives of the *Office* (7.2 percentage points) and the true-negatives of the *Kitchen* (7.1 percentage points) are also noteworthy. Based on these improvements we validate the *Kitchen* and *Coffee Bar* locations as the more challenging in the UNIGE-HANDS dataset.

Table 4. Comparsion of the performance of the HOG-SVM and the proposed DBN.

	True positives		True negatives	
	HOG-SVM	DBN	HOG-SVM	DBN
Office	0.893	0.965	0.929	0.952
Street	0.756	0.834	0.867	0.898
Bench	0.765	0.882	0.965	0.979
Kitchen	0.627	0.606	0.777	0.848
Coffee bar	0.817	0.874	0.653	0.660
Total	0.764	0.820	0.837	0.864

6 Conclusions and Future Research

This paper presents the UNIGE-HANDS dataset for *hand-detection* and extends a state-of-the-art method proposed in [6] incorporating a dynamic perspective. The dataset is recorded in 5 different locations and guarantees realistic conditions like, changes in the illumination, occlusions and fast camera movements. Additionally, the dataset is divided in training and testing videos to guarantee fair comparisons of coming methods.

To validate the consistence of the dataset with previous studies we evaluate the state-of-the-art method using cross validation, as suggested in [6,8], and using the testing videos of the dataset. Three conclusions arises from the results: i) The dataset is challenging enough, and the testing videos are a good

approach to avoid the bias in the cross validation results, ii) Little variations between frames highly affects the performance of the existing frame-by-frame *hand-detectors*, iii) The performances reported validates the results of previous studies on which SVM-HOG is the best combination for *hand-detection*.

The HOG-SVM frame by frame approach is extended using a Dynamic Bayesian Network where the dynamic part is carried by a Kalman filter with a constant acceleration model. The parameters of the KF, as well as the decision threshold, are tuned using a genetic algorithms and the Nelder-Mead simplex algorithm. The DBN is evaluated in each of the dataset locations and its performance is presented as the baseline to be used with the UNIGE-HANDS dataset. We highlight the model selection as an interesting research line that could lead to further improvements in the performance of the classifier.

References

1. Abbeel, P., Coates, A.: Discriminative training of Kalman filters. In: Robotics: Science and Systems, pp. 1–8. Cambridge, MA, USA (2005)
2. Aghazadeh, O., Sullivan, J., Carlsson, S.: Novelty detection from an ego-centric perspective. In: Computer Vision and Pattern Recognition, pp. 3297–3304. IEEE, Pittsburgh, June 2011
3. Alletto, S., Serra, G., Calderara, S., Cucchiara, R.: Head pose estimation in first-person camera views. In: International Conference on Pattern Recognition, p. 4188. IEEE Computer Society (2014)
4. Alletto, S., Serra, G., Calderara, S., Solera, F., Cucchiara, R.: From ego to nos-vision: detecting social relationships in first-person views. In: Computer Vision and Pattern Recognition, pp. 594–599. IEEE, June 2014
5. Bengio, Y., Grandvalet, Y.: No Unbiased Estimator of the Variance of k-fold Cross-Validation. The Journal of Machine Learning Research **5**, 1089–1105 (2004)
6. Betancourt, A.M.L., Rauterberg, M., Regazzoni, C.: A sequential classifier for hand detection in the framework of egocentric vision. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, vol. 1, pp. 600–605. IEEE, Columbus, June 2014
7. Betancourt, A., Morerio, P., Marcenaro, L., Barakova, E., Rauterberg, M., Regazzoni, C.: Towards a unified framework for hand-based methods in first person vision. In: IEEE International Conference on Multimedia and Expo (Workshops). IEEE, Turin (2015)
8. Betancourt, A., Morerio, P., Marcenaro, L., Rauterberg, M., Regazzoni, C.: Filtering SVM frame-by-frame binary classification in a detection framework. In: International Conference on Image Processing. IEEE, Quebec (2015)
9. Betancourt, A., Morerio, P., Regazzoni, C., Rauterberg, M.: The Evolution of First Person Vision Methods: A Survey. IEEE Transactions on Circuits and Systems for Video Technology **25**(5), 744–760 (2015)
10. Chelouah, R., Siarry, P.: Genetic and NelderMead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. European Journal of Operational Research **148**(2), 335–348 (2003)

11. Damen, D., Haines, O.: Multi-user egocentric online system for unsupervised assistance on object usage. In: European Conference on Computer Vision (2014)
12. Fathi, A., Farhadi, A., Rehg, J.: Understanding egocentric activities. In: International Conference on Computer Vision, pp. 407–414. IEEE, November 2011
13. Fathi, A., Hodgins, J., Rehg, J.: Social interactions: a first-person perspective. In: Computer Vision and Pattern Recognition, pp. 1226–1233. IEEE, Providence, June 2012
14. Fathi, A., Li, Y., Rehg, J.: Learning to recognize daily actions using gaze. In: European Conference on Computer Vision, pp. 314–327. Georgia Institute of Technology, Florence (2012)
15. Ghosh, J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: Computer Vision and Pattern Recognition, pp. 1346–1353. IEEE, June 2012
16. Kitani, K., Okabe, T.: Fast unsupervised ego-action learning for first-person sports videos. In: Computer Vision and Pattern Recognition, pp. 3241–3248. IEEE, Providence, June 2011
17. Lee, S., Bambach, S., Crandall, D., Franchak, J., Yu, C.: This hand is my hand: a probabilistic approach to hand disambiguation in egocentric video. In: Computer Vision and Pattern Recognition, pp. 1–8. IEEE Computer Society, Columbus (2014)
18. Li, C., Kitani, K.: Pixel-level hand detection in ego-centric videos. In: Computer Vision and Pattern Recognition, pp. 3570–3577. IEEE, June 2013
19. Li, Y., Fathi, A., Rehg, J.: Learning to predict gaze in egocentric video. In: International Conference on Computer Vision, pp. 1–8. IEEE (2013)
20. Mayol, W., Murray, D.: Wearable hand activity recognition for event summarization. In: International Symposium on Wearable Computers, pp. 1–8. IEEE (2005)
21. Morerio, P., Marcenaro, L., Regazzoni, C.: Hand detection in first person vision. In: Information Fusion, pp. 1502–1507. University of Genoa, Istanbul (2013)
22. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python. Research, Journal of Machine Learning **12**, 2825–2830 (2011)
23. Philipose, M.: Egocentric recognition of handled objects: benchmark and analysis. In: Computer Vision and Pattern Recognition, pp. 1–8. IEEE, Miami, June 2009
24. Pirsiavash, H., Ramanan, D.: Detecting activities of daily living in first-person camera views. In: Computer Vision and Pattern Recognition, pp. 2847–2854. IEEE, June 2012
25. Ryoo, M., Matthies, L.: First-person activity recognition: what are they doing to me? In: Conference on Computer Vision and Pattern Recognition, pp. 2730–2737. IEEE Comput. Soc, Portland (2013)
26. Schiele, B., Oliver, N., Jebara, T., Pentland, A.: An interactive computer vision system DyPERS: dynamic personal enhanced reality system. In: Christensen, H.I. (ed.) ICVS 1999. LNCS, vol. 1542, pp. 51–65. Springer, Heidelberg (1998)
27. Serra, G., Camurri, M., Baraldi, L.: Hand segmentation for gesture recognition in ego-vision. In: Workshop on Interactive Multimedia on Mobile & Portable Devices, pp. 31–36. ACM Press, New York (2013)

28. Spriggs, E., De La Torre, F., Hebert, M.: Temporal segmentation and activity classification from first-person sensing. In: Computer Vision and Pattern Recognition Workshops, pp. 17–24. IEEE, June 2009
29. Starner, T., Schiele, B., Pentland, A.: Visual contextual awareness in wearable computing. In: International Symposium on Wearable Computers, pp. 50–57. IEEE Computer Society (1998)
30. Sun, L., Klank, U., Beetz, M.: Eyewatchme3d hand and object tracking for inside out activity analysis. In: Computer Vision and Pattern Recognition, pp. 9–16 (2009)
31. Zariffa, J., Popovic, M.: Hand Contour Detection in Wearable Camera Video Using an Adaptive Histogram Region of Interest. Journal of NeuroEngineering and Rehabilitation **10**(114), 1–10 (2013)

George Azzopardi · Nicolai Petkov (Eds.)

Computer Analysis of Images and Patterns

16th International Conference, CAIP 2015
Valletta, Malta, September 2–4, 2015
Proceedings, Part I

Springer

Editors
George Azzopardi
University of Malta
Msida
Malta

Nicolai Petkov
University of Groningen
Groningen
The Netherlands

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-23191-4 ISBN 978-3-319-23192-1 (eBook)
DOI 10.1007/978-3-319-23192-1

Library of Congress Control Number: 2015946746

LNCS Sublibrary: SL6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

Springer Cham Heidelberg New York Dordrecht London
© Springer International Publishing Switzerland 2015
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)