Gesture Recognition in Ego-Centric Videos using Dense Trajectories and Hand Segmentation

Lorenzo Baraldi[1], Francesco Paci[2], Giuseppe Serra[1], Luca Benini[2,3], Rita Cucchiara[1]

[1]Dipartimento di Ingegneria "Enzo Ferrari"
Università di Modena e Reggio Emilia, Italy
baraldi.lorenzo@gmail.com
{name.surname}@unimore.it

[2]DEI
Univeristà di Bologna, Italy
{f.paci, luca.benini}@unibo.it

[3]Integrated Systems Laboratory
ETH Zürich, Switzerland
lbenini@iss.ee.ethz.ch

Abstract

We present a novel method for monocular hand gesture recognition in ego-vision scenarios that deals with static and dynamic gestures and can achieve high accuracy results using a few positive samples. Specifically, we use and extend the dense trajectories approach that has been successfully introduced for action recognition. Dense features are extracted around regions selected by a new hand segmentation technique that integrates superpixel classification, temporal and spatial coherence. We extensively test our gesture recognition and segmentation algorithms on public datasets and propose a new dataset shot with a wearable camera. In addition, we demonstrate that our solution can work in near real-time on a wearable device.

1. Introduction

Ego-centric vision is a paradigm that joins in the same loop humans and wearable devices to augment the subject vision capabilities by automatically processing videos captured with a first-person camera. We are interested in investigating the usage of ego-vision algorithms and devices to enhance new human-machine interfaces that could integrate information from the local environment with web and social media. These interfaces could help users to generate and share content in real-time, and could offer a customized experience, more suited for the user's specific cognitive needs and interests. For instance, ego-vision wearable systems could help understand what visitors of a museum are observing or doing, and determine their degree of interest, collecting data to enhance and customize visitors' experience.

Moreover, the recent growth of computational capability of embedded devices has made possible to exploit wearable and low-power devices as target platforms for ego-centric real-time applications. For this reason, applications and algorithms designed for ego-vision must be suited for portable input and elaboration devices, that often present a more constrained scenario, with different power needs and performance capabilities.

In this paper, we propose a hand gesture recognition approach that could be used in future human-machine interfaces. We take into account both static gestures, in which the meaning of the gesture is conveyed by the hand pose, and dynamic gestures, in which the meaning is given by motion too. It should be noted that gestures are somehow personal. In fact, they can vary from individual to individual and even for the same individual between different instances. Our method uses a monocular camera placed on the user's body to recognizes his gestures. The video stream processing is achieved with an ARM based embedded device that can be worn by users.

Our main contributions are:

- A novel gesture recognition algorithm for ego-vision applications that uses trajectories, appearance features and hand segmentation to classify static and dynamic hand movements, and that can achieve high accuacy results even when trained with a few positive samples.
- A performance analysis of the proposed method on an x86 based workstation and an ARM based embedded device that demonstrates that our algorithm can work in near real-time on a wearable device.

2. Related Work

The ego-vision scenario has been addressed only recently by the research community and mainly to understand human activities and to recognize hand regions. Pirsiavash *et al*. [15] detected activities of daily living using an approach that involves temporal pyramids and object detectors tuned for objects appearance during interactions and spatial

reasoning. Sundaram *et al*. [17] proposed instead to use Dynamic Bayesian Networks to recognize activities from low resolution videos, without performing hand detection and preferring computational inexpensive methods. Fathi *et al*. [5] used a bottom-up segmentation approach to extract hand held objects and trained object-level classifier to recognize objects; furthermore they also proposed an activity detection algorithm based on object state changes [4].

Regarding hand detection, Khan *et al*. in [8] studied color classification for skin segmentation. They pointed out how color-based skin detection has many advantages and potentially high processing speed, and demonstrated that Random Forest is one of the best classifiers for skin segmentation. Fathi *et al*. [5] proposed a different approach to hand detection, based on the assumption that background is static in the world coordinate frame, thus foreground objects are detected as to be the moving regions respect to the background. This approach is shown to be a robust tool for skin detection and hand segmentation in indoor environments, even if it performs poorly with more unconstrained scenarios. Li *et al*. [11] proposed a method with sparse feature selection which was shown to be an illumination-dependent strategy. To solve this issue, they trained a set of Random Forests indexed by a global color histogram, each one reflecting a different illumination condition.

To our knowledge, the study of gesture recognition in the ego-centric paradigm has not yet been addressed. Even though not related to ego-vision domain, several approaches to gesture and human action recognition have been proposed. Kim *et al*. [9] extended Canonical Correlation Analysis to measure video-to-video similarity in order to represent and detect actions in video. Lui *et al*. [13, 12] used tensors and tangent bundle on Grassmann manifolds to classify human actions and hand gestures. Sanin *et al*. [16] developed a new and more effective spatio-temporal covariance descriptor to classify gestures in conjunction with a boost classifier. However, all these approaches are not appropriate for the ego-centric perspective, as they do not take into account any of the specific characteristics of this domain, such as fast camera motion, hand presence and background cluttering, as well as the limited computational power of wearable platforms.

3. Proposed Method

Gesture recognition systems should recognize both static and dynamic hand movements. Therefore, we propose to describe each gesture as a collection of dense trajectories extracted around hand regions. Feature points are sampled inside and around the user's hands and tracked during the gesture; then several descriptors are computed inside a spatio-temporal volume aligned with each trajectory, in order to capture its shape, appearance and movement at each frame. These descriptors are coded, using the Bag of Words approach and power normalization, in order to obtain the final feature vectors, which are then classified using a linear SVM classifier. A summary of our approach is presented in Figure 1.

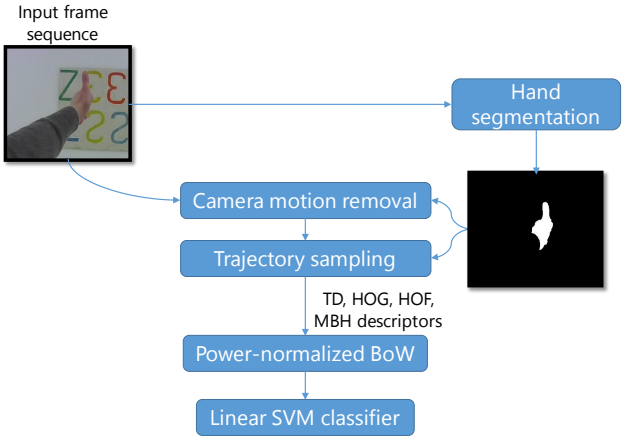

Figure 1: Outline of the proposed Gesture Recognition method.

3.1. Camera motion removal

To describe shape, appearance and movement of each trajectory we use the Trajectory descriptor, histograms of oriented gradients, of optical flow, and motion boundary histograms, following [18]. The Trajectory descriptor captures trajectory shape, HOG are based on the orientation of image gradient and encode the static appearance of the region surrounding the trajectory, HOF and MBH are based on optical flow and capture motion information.

In order to remove camera motion, the homography between two consecutive frames is estimated running the RANSAC algorithm on densely sampled features points. SURF features and sample motion vector are extracted from the optical flow to get dense matches between frames.

However, in first-person camera views hands movement is not consistent with camera motion and this generates wrong matches between the two frames. For this reason we introduce a segmentation mask that disregards feature matches belonging to hands. In fact, without the hand segmentation mask, many feature points from the user's hands would become inliers, degrading the homography estimation. As a consequence, the trajectories extracted from the video would be incorrect. Instead, computing an homography using feature points from non-hand regions allows us remove all the camera movements.

3.2. Gesture Description

Having removed camera motion between two adjacent frames, trajectories can be extracted. The second frame is warped with the estimated homography, the optical flow be-

tween the first and the second frame is recomputed, and then feature points around the hands of the user are sampled and tracked following what [18] does for human action recognition. Feature points are densely sampled at several spatial scales and tracked using median filtering in a dense optical flow field. In contrast to [18], trajectories are restricted to lie inside and around the user's hands: at each frame the hand mask is dilated, and all the feature points outside the computed mask are discarded.

Then, the spatio-temporal volume aligned with each trajectory is considered, and Trajectory descriptor, HOG, HOF and MBH are computed around it. While HOF and MBH are averaged on five consecutive frames, a single HOG descriptor is computed for each frame. In this way we can better describe how the hand pose changes in time. After this step, we get a variable number of trajectories for each gesture. In order to obtain a fixed size descriptor, the Bag of Words approach is exploited: we train four separate codebooks, one for each descriptor. Each codebook contains 500 visual words and is obtained running the k-means algorithm in the feature space.

Since BoW histograms in our domain tend to be sparse, they are power normalized to unsparsify the representation, while still allowing for linear classification. To perform power-normalization [14], the following function is applied to each bin h_i:

$$f(h_i) = \text{sign}(h_i) \cdot |h_i|^{\frac{1}{2}} \tag{1}$$

The final feature vector is then obtained by the concatenation of its four power-normalized histograms. Eventually, gestures are recognized using a linear SVM 1-vs-1 classifier.

3.3. Hand Segmentation

The proposed gesture recognition approach uses a hand segmentation mask to distinguish between camera and hand motions, and to prune away all the trajectories that do not belong to the user hand. In this way, our descriptor captures hands movement and shape as if the camera was fixed, and disregards the noise coming from other moving regions that could be in the scene.

For computing hand segmentation masks, at each frame we extract superpixels using the SLIC algorithm [3], that performs a k-means-based local clustering of pixels in a 5-dimensional space, where color and pixel coordinates are used. Superpixels are represented with several features: histograms in the HSV and LAB color spaces (that have been proved to be good features for skin representation [8]), Gabor filters and a simple histogram of gradients, to discriminate between objects with a similar color distribution.

In order to deal with different illumination conditions we also train a collection of Random Forest classifiers indexed by a global HSV histogram, instead of using a single classifier. Hence, training images are distributed among the classifiers by a k-means clustering on the feature space. At test time, the predictions from the five nearest classifier are averaged to make the final prediction.

Furthermore, semantic coherence in time and space is taken into account. Since past frames should affect the prediction for the current frame, a smoothing filter is applied, so that the prediction for each frame is replaced with a combination of the classifier results from past frames. Then, to remove small and isolated pixel groups and also to aggregate bigger connected pixel groups, the GrabCut algorithm is applied to exploit spatial consistency.

4. Experimental Results

To compare the performance of the proposed gesture recognition algorithm with existing approaches, we test it on the Cambridge-Gesture database [10], which includes nine hand gesture types performed on a table, under different illumination conditions. To better investigate the effectiveness of the proposed approach in videos taken from the ego-centric perspective and in a museum setting, we also propose a far more realistic and challenging dataset which contains seven gesture classes, performed by five subjects in an interactive exhibition room which functions as a virtual museum. Furthermore, to evaluate the hand segmentation approach, we test it on the publicly available CMU EDSH dataset [11] which consists of three ego-centric videos with indoor and outdoor scenes and large variations of illuminations. We implemented two different versions of our approach, one targeted for x86 based workstations and a lightweight version for ARM based embedded devices. We present performance evaluations on these two implementations and evaluate the accuracy-performance tradeoff of the embedded version.

4.1. Gesture Recognition

The Cambridge Hand Gesture dataset contains 900 sequences of nine hand gesture classes. Although this dataset does not contain ego-vision videos it is useful to compare our results to recent gesture recognition techniques. In particular, each sequence is recorded with a fixed camera, placed over one hand, and hands perform leftward and rightward movements on a table, with different poses. The whole dataset is divided in five sets, each of them containing image sequences taken under different illumination conditions. The common test protocol, proposed in [10], requires to use the set with normal illumination for training and the remaining sets for testing, thus we use the sequences taken in normal illumination to generate the BoW codebooks and to train the SVM classifier. Then, we perform the test using the remaining sequences.

Method	Set1	Set2	Set3	Set4	Overall
TCCA [9]	0.81	0.81	0.78	0.86	0.82
PM [13]	0.89	0.86	0.89	0.87	0.88
TB [12]	**0.93**	0.88	0.90	0.91	0.91
Cov3D [16]	0.92	**0.94**	0.94	0.93	0.93
Our method	0.92	0.93	**0.97**	**0.95**	**0.94**

Table 1: Recognition rates on the Cambridge dataset.

(a) *Dislike* gesture

(b) *Point* gesture

Figure 2: Sample gestures from the Interactive Museum dataset.

Table 1 shows the recognition rates obtained with our gesture recognition approach, compared with the ones of tensor canonical correlation analysis (TCCA) [9], product manifolds (PM) [13], tangent bundles (TB) [12] and spatio-temporal covariance descriptors (Cov3D) [16]. Results show that proposed method outperforms the existing state-of-the-art approaches.

We then propose the Interactive Museum dataset, a gesture recognition dataset taken from the ego-centric perspective in a virtual museum environment. It consists of 700 video sequences, all shot with a wearable camera, in an interactive exhibition room, in which paintings and artworks are projected over a wall, in a virtual museum fashion (see figure 2). The camera is placed on the user's head and captures a 800×450, 25 frames per second 24-bit RGB image sequence. In this setting, five different users perform seven hand gestures: *like*, *dislike*, *point*, *ok*, *slide left to right*, *slide right to left* and *take a picture*. Some of them (like the *point*, *ok*, *like* and *dislike* gestures) are statical, others (like the two *slide* gestures) are dynamical. This dataset is very challenging since there is fast camera motion and users have not been trained before recording their gestures, so that each user performs the gestures in a slightly different way, as would happen in a realistic context. We have publicly released our dataset[1].

Since Ego Vision applications are highly interactive, their setup step must be fast (i.e. few positive examples can be acquired). Therefore, to evaluate the proposed gesture recognition approach, we train a 1-vs-1 linear classifier for each user using only two randomly chosen gestures per class as training set. The reported results are the average over 100 independent runs.

[1] http://imagelab.ing.unimore.it/files/ego_virtualmuseum.zip

User	No segmentation	With segmentation
Subject 1	0.91	**0.95**
Subject 2	0.87	**0.87**
Subject 3	0.92	**0.95**
Subject 4	**0.96**	0.94
Subject 5	0.91	**0.96**
Average	0.91	**0.93**

Table 2: Gesture recognition accuracy on the Interactive Museum dataset with and without hand segmentation.

In Table 2 we show the gesture recognition accuracy for each of the five subjects, and we also compare with the ones obtained without the use of the hand segmentation mask for camera motion removal and trajectories pruning. Results show that our approach is well suited to recognize hand gestures in the ego-centric domain, even using only two positive samples per gesture, and that the use of the segmentation mask can improve recognition accuracy.

4.2. Hand Segmentation

The CMU EDSH dataset consists of three ego-centric videos (EDSH1, EDSH2, EDSHK) containing indoor and outdoor scenes where hands are purposefully extended outwards to capture the change in skin color. As this dataset does not contain any gesture annotation, we use it to evaluate only the hand segmentation part.

We validate the techniques that we have proposed for temporal and spatial consistency. In Table 3 we compare the performance of the hand segmentation algorithm in terms of F1-measure, firstly using a single Random Forest classifier, and then incrementally adding illumination invariance, the temporal smoothing filter and the spatial consistency technique via the GrabCut algorithm application. Results shows that there is a significant improvement in performance when all the three techniques are used together: illumination invariance increases the performance with respect to the results obtained using only a single random forest classifier, while temporal smoothing and spatial consistency correct incongruities between adjacent frames, prune away small and isolated pixel groups and merge spatially nearby regions, increasing the overall performance.

Then, in Table 4 we compare our segmentation method with different techniques: a video stabilization approach

Features	EDSH2	EDSHK
Single RF classifier	0.761	0.829
II	0.789	0.831
II + TS	0.791	0.834
II + TS + SC	**0.852**	**0.901**

Table 3: Performance comparison considering Illumination Invariance (II), Temporal Smoothing (TS) and Spatial Consistency (SC).

Method	EDSH2	EDSHK
Hayman and Eklundh [6]	0.211	0.213
Jones and Rehg [7]	0.708	0.787
Li and Kitani [11]	0.835	0.840
Our method	**0.852**	**0.901**

Table 4: Hand segmentation comparison with the state-of-the-art.

based on background modeling [6], a single-pixel color method inspired by [7] and the approach proposed in [11] by Li *et al*., based on a collection of Random Forest classifiers. As can be seen, the single-pixel approach, which basically uses a random regressor trained only using the single pixel LAB values, is still quite effective, even if conceptually simple. Moreover, we observe that the video stabilization approach performs poorly on this dataset, probably because of the large ego-motions these video present. The method proposed by Li *et al*. is the most similar to our approach, nevertheless exploiting temporal and spatial coherence we are able to outperform their results.

4.3. Performance Evaluations

We have first implemented and tested our algorithm on an Intel based workstation, with a i7-2600 CPU that runs at 3.40 GHz, and then developed a lightweight version that reaches good performance even on low-power devices. On the workstation implementation we did not perform any code optimization whereas the embedded implementation has been optimized using OpenMP and tested on a Odroid-XU developer board [1]. This board embeds the ARM Exynos 5 SoC, hosting a Quad big.LITTLE ARM processor (Cortex A15 and A7), codename 5410 [2].

To evaluate execution times on both architectures, we divide our algorithm in four modules: the *Trajectory sampling* module, which includes trajectories extraction and description, the *Power-normalized BoW* module, that exploits the Bag of Words approach and power normalization to build the final feature vectors, the *Classification* module, that performs linear SVM classification, and the *Hand Segmentation* module, that runs our segmentation algorithm.

The embedded version has been implemented in C++ and each described module has been optimized, using OpenMP parallel regions. This allows to exploit the computational power of the Exynos processor, that embeds two clusters A15 and A7, that runs from 250 MhZ to 1.6 GhZ. The performance evaluation has been done at maximum CPU frequency using the A15 cluster. Figure 3 shows the execution time for each module on both devices.

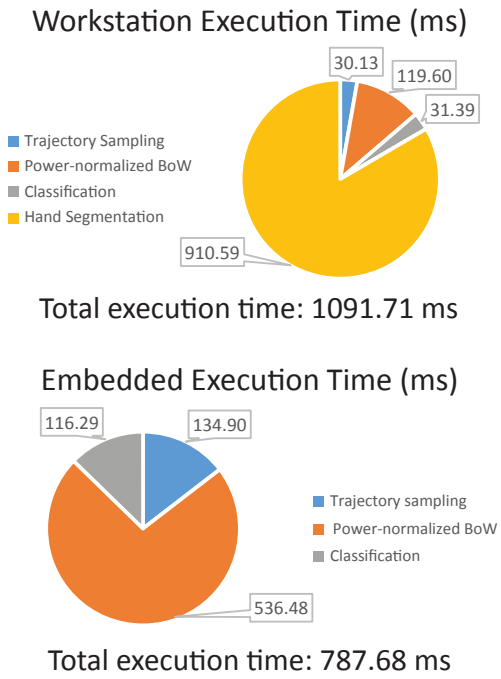

Figure 3: Performance comparison between Workstation and Embedded implementations on 15 frames trajectories.

As can be seen, *Trajectory Sampling*, *Power-normalized BoW* and *Classification* modules tested on the workstation reach around 4x to 5x speedup, compared to the embedded ones. Thus for the embedded implementation we removed the hand segmentation module that has the worst accuracy/performance contribution to the whole algorithm. Result shows that the workstation implementation can elaborate almost 15 frames per second, while the embedded one reaches around 19 fps, when the *Hand segmentation* module is disabled. This is a good result that means that we reach a near real-time frame rate, for both the two versions. Moreover comparing these results with Figure 3 and Table 2 it is possible to correlate the accuracy loss, that is around 2%. Hence we trade off a modest accuracy loss for being able to reach near real time performance.

5. Conclusion

We described a novel approach to hand gesture recognition in ego-centric videos. Our work is motivated by the increasing interest in ego-centric human-machine interfaces and by the growth of computational capabilities of wearable devices, which encourages the development of real-time computer vision algorithms. We presented a model that can deal with static and dynamic gestures and can achieve high accuracy results even when trained with a few positive samples. Our gesture recognition and hand segmentation results outperform the state-of-the-art approaches on Cambridge Hand Gesture and CMU EDSH datasets. Finally, we demonstrated that our algorithm can work in near real-time on a wearable Odroid board.

References

[1] Odroid-XU developement board by Hardkernel. http://www.hardkernel.com. 5

[2] Samsung Exynos5 5410 ARM SoC. http://www.samsung.com/global/business/semiconductor/minisite/Exynos/products5octa_5410.html. 5

[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012. 3

[4] A. Fathi and J. M. Rehg. Modeling actions through state changes. In *Proc. of CVPR*, 2013. 2

[5] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In *Proc. of CVPR*, 2011. 2

[6] E. Hayman and J.-O. Eklundh. Statistical background subtraction for a mobile observer. In *Proc. of ICCV*, 2003. 5

[7] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. 1999. 5

[8] R. Khan, A. Hanbury, and J. Stoettinger. Skin detection: A random forest approach. In *Proc. of ICIP*, 2010. 2, 3

[9] T.-K. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(8):1415–1428, 2009. 2, 4

[10] T.-K. Kim, K.-Y. K. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *Proc. of CVPR*, 2007. 3

[11] C. Li and K. M. Kitani. Pixel-level hand detection in ego-centric videos. In *Proc. of CVPR*, 2013. 2, 3, 5

[12] Y. M. Lui and J. R. Beveridge. Tangent bundle for human action recognition. In *In proc. of Automatic Face & Gesture Recognition and Workshops*, 2011. 2, 4

[13] Y. M. Lui, J. R. Beveridge, and M. Kirby. Action classification on product manifolds. In *Proc. of CVPR*, 2010. 2, 4

[14] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. of ECCV*, 2010. 3

[15] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Proc. of CVPR*, 2012. 1

[16] A. Sanin, C. Sanderson, M. T. Harandi, and B. C. Lovell. Spatio-temporal covariance descriptors for action and gesture recognition. In *Proc. of Workshop on Applications of Computer Vision*, 2013. 2, 4

[17] S. Sundaram and W. W. M. Cuevas. High level activity recognition using low resolution wearable vision. In *Proc. of CVPR*, 2009. 2

[18] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *Proc. of CVPR*, 2011. 2, 3