Personalizing Vision-based Gestural Interfaces for HRI with UAVs: a Transfer Learning Approach

Gabriele Costante[1], Enrico Bellocchio[1], Paolo Valigi[1] and Elisa Ricci[1,2]

***Abstract*— Following recent works on HRI for UAVs, we present a gesture recognition system which operates on the video stream recorded from a passive monocular camera installed on a quadcopter. While many challenges must be addressed for building a real-time vision-based gestural interface, in this paper we specifically focus on the problem of user personalization. Different users tend to perform the same gesture with different styles and speed. Thus, a system trained on visual sequences depicting some users may work poorly when data from other people are available. On the other hand, collecting and annotating many user-specific data is time consuming. To avoid these issues, in this paper we propose a personalized gestural interface. We introduce a novel transfer learning algorithm which, exploiting both data downloaded from the web and gestures collected from other users, permits to learn a set of person-specific classifiers. We integrate the proposed gesture recognition module into a HRI system with a flying quadrotor robot. In our system first the UAV localizes a person and individuates her identity. Then, when a user performs a specific gesture, the system recognizes it adopting the associated user-specific classifier and the quadcopter executes the corresponding task. Our experimental evaluation demonstrates that the proposed personalized gesture recognition solution is advantageous with respect to generic ones.**

I. Introduction

While many efforts in the past have been devoted to develop Human Robot Interaction (HRI) systems for commanding both single and multiple robots, only recently, with the advent on the market of low cost Unmanned Aerial Vehicles (UAVs), researchers have concentrated their efforts toward devising HRI solutions with flying quadrotor robots. Aiming at emulating the way humans interact only through eye contact and simple gestures, non-verbal communication methods specifically targeted to HRI with UAVs have been proposed recently [1], [2], [3]. These approaches have the advantages of avoiding the need of instrumenting the human as the only adopted sensors are those on-board of the robot. Thus, they represent a practical solution easy to deploy.

Previous works [1], [2] have considered the use of low cost cameras as on board sensors and visual based gesture interaction systems for UAVs have been proposed. Many challenges arise in this context. First, devising a vision-mediated HRI with flying quadrotor robots which are continuously moving is much more challenging with respect to considering stationary wheeled mobile robots. This requires to develop a gesture recognition module which correctly separates motion patterns due to robot movements from those ascribable to the user's gesture. Furthermore, robustness to point-of-view variations of the target with respect to the camera is required. Second, varying illumination conditions are difficult to handle with traditional cameras, while solutions based on modern RGB-D sensors works well only if the target is close enough to the robot. Finally, visual-based gesture recognition is a difficult problem since people tend to perform the same gestures with different styles and speed.

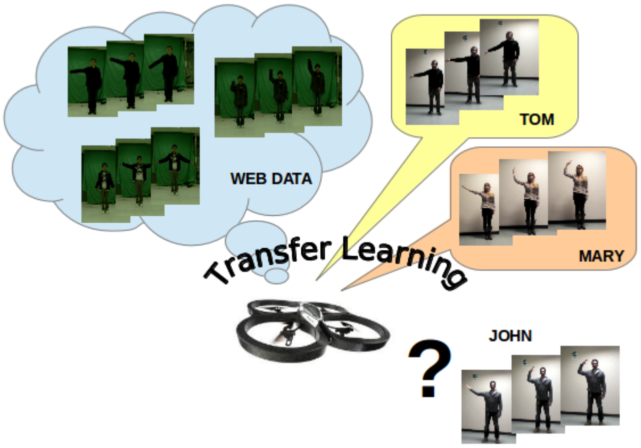


Fig. 1. Overview of the proposed problem. An uninstrumented person commands an UAV using face engagement and hand gestures. As different users tend to perform the same gesture with different styles and speed, we propose a personalized gestural interface. Having at disposal only few sample gesture sequences for a specific user, the system performs *knowledge transfer* reusing visual data gathered from the web or previously collected by other people.

In this paper we specifically address the latter and consider a visual-based HRI system where an uninstrumented user commands a single UAV by performing a set of specific gestures. Differently from previous works, we focus on developing a user-specific gesture recognition module. We consider the scenario (Fig.1) where only few gesture samples (the *target* samples) are available from a specific user, collected during a very short configuration phase. As a classifier trained only on these data will perform poorly, we propose to exploit the knowledge available in form of video sequences (the *source* data) downloaded from publicly available datasets on the web or collected by other users previously interacting with the robot. To maximally benefit from these informations we propose a novel transfer learning approach. Transfer learning operates by selecting useful source data in order to improve classification performance on the target set. As demonstrated in our experimental results, the obtained user-specific classifier is much more

G. Costante, E. Bellocchio, P. Valigi and E. Ricci are with the Department of Engineering, University of Perugia, Via G. Duranti, 93, 06125, Perugia, Italy. E.Ricci is with Fondazione Bruno Kessler, Via Sommarive, 18, 38123, Trento, Italy {gabriele.costante, enrico.bellocchio}@gmail.com, {valigi,ricci}@unipg.it

accurate then a classifier trained only on few sequences of the same user and than a global classifier constructed from data collected from many users. We integrate the proposed gesture recognition method into a HRI system by devising a user identification module such that the appropriate gesture recognition classifier can be invoked at runtime.

To summarize the main contributions of this work are: (i) Our paper is one of the first attempts to address the challenging problem of vision-mediated HRI with flying quadrotor robots. In our scenario a single robot is engaged in a one-on-one interaction with the user. Differently from recent works, the system recognizes the user's identity and uses this information in the gesture recognition module by invoking the appropriate classifier. (ii) To our knowledge, no previous works have proposed to improve the performance of a gesture recognition system within a transfer learning framework, *i.e.* by reusing information from other users or downloading data from the web. (iii) The proposed transfer learning algorithm is also novel and oppositely to many previous methods can effectively handle the situation where source and target data have different gesture categories.

The rest of the paper is organized as follows. Related works are reviewed in Section II. In Section III the proposed gesture recognition system is described together with our novel transfer learning algorithm. In Section IV its integration into a HRI system for UAV is illustrated. Section V shows the experimental evaluation, while in Section VI conclusions are drawn.

II. Related Works

A. Human Robot Interaction with UAVs

Recent works on HRI have focused on solutions where the human is not provided any instrumentation as the aim is to achieve interactions as much as natural as possible [4]. Therefore detecting gaze and body movements is considered a central problem. In particular, vision-mediated solutions for recognizing gestures have been proposed in several previous works [1], [2], [3] and few of them have considered specifically HRI with UAVs. In [1] a system where a teams of UAVs receive a series of commands from an uninstrumented human is presented. Similarly, we propose a gesture recognition approach operating on the video stream of a low cost camera installed on-board of the UAV. However, while in [1] the authors focused on a multi-robot scenario, we specifically address the problem of recognizing user-specific hand movements. In [2] a single robot/single user setup is considered and a vision based gestural interface is realized. However a depth sensor is used instead of a traditional RGB camera. A multi-robot system employing RGB-D cameras is also presented in [3]. In [5] different technologies for the development of a ground control station for UAVs are studied and multimodal solutions (not only visual) are investigated. Song *et al.*[6] introduced a database of body and hand gestures indicating aircraft handling signals but the videos are recorded from stationary cameras.

B. Visual based Gesture Recognition

In the vision community, the problems of gesture and action recognition have received considerable attention in the past [7]. In general visual based gesture interaction systems suffer from lack of robustness due to point of view changes, intra-class variations and self-occlusions. To overcome the issue of point of view variations many works have focused on using view-invariant features [8], [9] and more recently on transferring visual features across views [10], [11], [12], [13]. However, to our knowledge, transferring visual information for personalization (*i.e.* for improving the performance of a gesture recognition system in HRI by learning user specific classifiers) has not been proposed before.

C. Transfer Learning in Robotics

As the novel generation of autonomous systems is intended to operate in challenging unconstrained scenarios, researchers are struggling to develop robots emulating as much as possible the human abilities. For example people can naturally transfer knowledge obtained from their past experience or reuse informations gathered from the web by properly selecting the relevant ones. Constructing robots with similar abilities is one of the main goals of the research activities involving the RobotEarth platform [14]. Following [14] many other works have been recently proposed. In [15] the authors introduced an approach for processing web resources to help a robot to perform everyday manipulation tasks. Instead of building specific models for each platform, the robot can process information gathered from some websites and acquire the knowledge needed to perform specific tasks, *e.g.* cleaning a room or repairing a machinery. Similarly Samadi *et al.* [16] proposed a method where the web is searched to learn a model reflecting the probability of finding objects in specific rooms. In [17] a transfer learning approach for addressing the problem of semantic place recognition is presented where visual data obtained from the web is reused, if relevant, to construct an environment-specific classifier. In the context of place categorization the domain adaptation problem is also considered in [18], [19]. However, none of these previous works proposed a transfer learning framework for personalized gesture recognition.

III. Personalized Gesture Recognition

Our gesture recognition module is made of two parts. First, features are extracted from a video sequence by computing Histograms of Optical Flows (HOF) at every frame and using a Fisher Kernel representation [20] to calculate the sequence descriptor. Then the proposed transfer learning approach is used to build a set of user-specific classifiers.

A. Feature Extraction

Our method is based on extracting motion features from image sequences using dense optical flow. We adopt the Farneback [21] algorithm available in the OpenCV library, which ensures both speed and accuracy. Once the optical flow vectors are computed, the region of interest is divided in 3×3 blocks and for each block an HOF is extracted. We consider

12 bin for each block histogram descriptor. HOFs are then normalized to sum up to 1 and concatenated, obtaining a global histogram of 12x9=108 bins. Once HOF are calculated for each frame of a sequence, then the Fisher Kernel is used to compute the descriptor of an entire sequence. The Fisher Kernel has been introduced in [20] in the context of image classification as a way to improve the construction of a visual vocabulary in the bag-of-words framework. Recently, in [22] Mironica *et al.*proposed to employ the Fisher Kernel to capture variation in time in a video sequence and showed how this can be successfully used for action recognition. We follow [22] and we consider a Gaussian Mixture Model with diagonal covariance matrices and $k = 8$ clusters as generative distribution. Given a set of d-dimensional feature vectors x_t computed at frame t ($d = 108$), the Fisher vector is obtained concatenating the two terms $\mathcal{F}_{\mu,i}$ and $\mathcal{F}_{\sigma,i}$, $\forall i = 1, \dots, k$ defined as:

$$\begin{aligned}\mathcal{F}_{\mu,i} &= \frac{1}{T\sqrt{\omega_i}} \sum_t \beta(i) \frac{x_t - \mu_i}{\sigma_i} \\ \mathcal{F}_{\sigma,i} &= \frac{1}{T\sqrt{2\omega_i}} \sum_t \beta(i) \left[\frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right]\end{aligned}$$

where T is the number of frames in a video, μ_i and σ_i are the mean and standard deviation of the i-th Gaussian centroid, $\beta(i)$ represents the soft assignment to the i-th Gaussian of x_t. The final descriptor for the j-th sequence $\mathbf{x}_j$ has a dimensionality of $2kd = 1728$. Intuitively, the Fisher vector descriptor captures the temporal dynamics as the term $\mathcal{F}_{\mu,i}$ averages over time the features associated to the same component, while $\mathcal{F}_{\sigma,i}$ models the variation of features corresponding to the same component, capturing more subtle visual changes. In our experiments we use L_2 norm for the normalization of the Fisher vectors. The computed features are then provided to the classifier described in the following.

B. Transfer Learning for Gesture Recognition

In this subsection we present our transfer learning method for gesture recognition. The aim of the proposed approach is to learn a set of user-specific distance functions to effectively classify a set of gestures, the *target set* $\mathcal{T}$, taking advantage from previous knowledge, *i.e.* a set of video sequences (the *source set* $\mathcal{S}$) depicting gestures collected by other users or downloaded from the web. In general, we can expect that sequences gathered from websites may have different gestures types than the one the current user will perform, *i.e.* the source and the target set may contain different classes. In our scenario the system is asked to recognize person-specific gestures. A very short configuration phase is needed where the user perform only two sample sequences for each gesture type (*i.e.* the set $\mathcal{T}$ is small). Having at disposal only very few training samples in $\mathcal{T}$, it is very challenging to learn an accurate recognition model. Thus, intuitively, if we can take advantage from past knowledge, even if the novel categories are different, we should expect improved performance.

Algorithm 1 Algorithm to solve (3)

```
**Input:** The sets $\mathcal{T}$ and $\mathcal{S}$, the regularization parameter $\lambda$, the number of iteration $T$, the threshold $\zeta$.

Compute $MMD$ with (1).
Initialize $\mathbf{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_{K_t}] = \mathbf{0}$.
for $t = 1, \dots, T$ do
  Initialize $\mathbf{D} \in I\!R^{dK_t}$, $\mathbf{D} = \mathbf{0}$.
  $c = 0$.
  for $i = 1, \dots, N_t$ do
    if $(1 - \boldsymbol{\theta}_{l_i}^T(\mathbf{d}_{ik} - \mathbf{d}_{ij}) \geq 0, \forall \mathbf{x}_j, \mathbf{x}_k, l_k \neq l_i, l_j = l_i)$
      $\mathbf{D}[(l_i - 1)d + 1 : l_i d] = \mathbf{D}[(l_i - 1)d + 1 : l_i d] + \mathbf{d}_{ik} - \mathbf{d}_{ij}$
      $c = c + 1$.
    endif
  endfor
  for $q = 1, \dots, N_s$ do
    Compute $p = \arg\min_{z \in \mathcal{C}^t} MMD(y_q, z)$
    if $(1 - \boldsymbol{\theta}_p^T(\mathbf{d}_{qn} - \mathbf{d}_{qm}) \geq 0, \forall \mathbf{x}_n, \mathbf{x}_m, y_n \neq y_q, y_m = y_q)$
      Compute $r = \arg\min_{a \in \mathcal{C}^t, p \neq a} MMD(y_n, a)$, $y_n \neq y_q$
        if $(MMD(y_q, p) \leq \zeta \wedge MMD(y_n, r) \leq \zeta)$
        $\mathbf{D}[(p-1)d+1 : pd] = \mathbf{D}[(p-1)d+1 : pd] + \mathbf{d}_{qn} - \mathbf{d}_{qm}$
          $c = c + 1$.
        endif
      endif
  endfor
  $\mathbf{\Theta}_{t+\frac{1}{3}} = \left(1 - \frac{1}{t}\right)\mathbf{\Theta}_t + \frac{1}{c\lambda t}\mathbf{D}$
  $\mathbf{\Theta}_{t+\frac{2}{3}} = \max\{0, \mathbf{\Theta}_{t+\frac{1}{3}}\}$
  $\mathbf{\Theta}_{t+1} = \min\{1, \frac{1}{\sqrt{\lambda}\|\mathbf{\Theta}_{t+\frac{2}{3}}\|}\}\mathbf{\Theta}_{t+\frac{2}{3}}$
endfor
 **Output:** $\mathbf{\Theta}$
```

Formally, we are given the source set $\mathcal{S} = \{(\mathbf{x}_1^s, y_1), (\mathbf{x}_2^s, y_2), \dots, (\mathbf{x}_{N_s}^s, y_{N_s})\}$ and the target set $\mathcal{T} = \{(\mathbf{x}_1^t, l_1), (\mathbf{x}_2^t, l_2), \dots, (\mathbf{x}_{N_t}^t, l_{N_t})\}$. The vectors $\mathbf{x}_i \in I\!R^d$ contain the Fisher Kernel descriptor computed on the i-th video sequence, $y_i \in \mathcal{C}^s = \{C_1^s, \dots C_{K_s}^s\}$, $l_i \in \mathcal{C}^t = \{C_1^t, \dots C_{K_t}^t\}$ are the source and the target labels and N_s and N_t are respectively the number of source and target data. As stated above, the source and the target sets may contain different classes. To perform knowledge transfer it is crucial to automatically understand what to transfer from the source, *i.e.* which source data are useful in the target domain. In this paper we propose to use the Maximum Mean Discrepancy (MMD) [23] to compute the distance between the distributions of the data of one class of the source and one class of the target. The MMD between two classes C_i^s and C_j^t is:

$$MMD^2[C_i^s, C_j^t] = \left\| \frac{1}{m_i^s} \sum_{i=1}^{m_i^s} \phi(\mathbf{x}_i) - \frac{1}{m_j^t} \sum_{j=1}^{m_j^t} \phi(\mathbf{x}_j) \right\|_{\mathcal{H}}^2 \qquad (1)$$

$$= \frac{1}{m_i^{s2}} \|K_{C_i^s C_i^s}\|_1 - \frac{2}{m_i^s m_j^t} \|K_{C_i^s C_j^t}\|_1 + \frac{1}{m_j^{t2}} \|K_{C_j^t C_j^t}\|_1$$

where $\phi(\cdot)$ is a feature mapping function, K is the associated kernel (in this paper we simply consider a linear kernel) and m_i^s and m_j^t are the number of samples of classes C_i^s and C_j^t. We first compute the MMD of every pair of source and target classes using (1) and then we learn a distance function to be used to classify novel gestures according to a nearest neighbor scheme. More specifically, we propose to learn a set of K_t distance function $\delta_c(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\theta}_c^T \mathbf{d}(\mathbf{x}_i, \mathbf{x}_j) =$

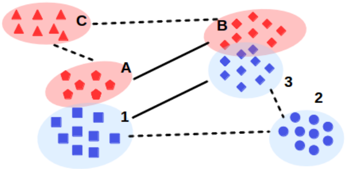


Fig. 2. A visual representation of our constraints selection strategy. $\mathcal{C}^s = \{$A,B,C$\}$: source classes. $\mathcal{C}^t = \{1,2,3\}$: target classes. The constraints between A and B are preserved as the data distributions of A and B are similar to those of 1 and 3. The constrains between A and C and between B and C are discarded as C is not similar to any of the target classes.

$\sum_d \theta_c^d |x_i^d - x_j^d| = \boldsymbol{\theta}_c^T \mathbf{d}_{i,j}$, one for each target class. We learn them imposing that, when considering target data, gestures of the same class should be close while gestures of different categories should be separated at least by a margin of 1. In formulas:

$$\boldsymbol{\theta}_{l_i}^T(\mathbf{d}_{i,k} - \mathbf{d}_{i,j}) \geq 1 \quad \forall \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \, \forall l_i \neq l_k, \, l_i = l_j$$

where $\boldsymbol{\theta}_{l_i}$ is the weight vector relative to the target class l_i. As we also want to exploit information from the source data, we formulate similar constraints on the set $\mathcal{S}$, *i.e.* :

$$\boldsymbol{\theta}_p^T(\mathbf{d}_{q,n} - \mathbf{d}_{q,m}) \geq 1 \quad \forall \mathbf{x}_q, \mathbf{x}_m, \mathbf{x}_n, \, \forall y_q \neq y_n, \, y_q = y_m$$

where $p = \arg\min_{z \in \mathcal{C}^t} MMD(y_q, z)$. Defining $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{K_t}]$ and:

$$\gamma_{qmn} = \begin{cases} 1 & if \quad \min_{z \in \mathcal{C}^t} MMD(y_q, z) \leq \zeta \;\; \wedge \\ & \quad \min_{a \in \mathcal{C}^t, z \neq a} MMD(y_n, a) \leq \zeta \\ 0 & otherwise \end{cases} \tag{2}$$

we consider the following optimization problem:

$$\begin{aligned} \min \quad & \frac{\lambda}{2}||\boldsymbol{\Theta}||^2 + \frac{1}{N_1}\sum \xi_{ijk} + \frac{1}{N_2}\sum \gamma_{qmn}\xi_{qmn} \qquad (3) \\ \text{s.t.} \quad & \boldsymbol{\theta}_{l_i}^T(\mathbf{d}_{ik} - \mathbf{d}_{ij}) \geq 1 - \xi_{ijk} \; \forall i,j,k \quad l_i = l_j, l_i \neq l_k \\ & \boldsymbol{\theta}_p^T(\mathbf{d}_{qn} - \mathbf{d}_{qm}) \geq 1 - \xi_{qmn} \; \forall q,m,n \quad y_q = y_m, y_q \neq y_n \\ & \boldsymbol{\Theta}, \xi_{ijk}, \xi_{qmn} \geq 0 \end{aligned}$$

where the slack variables ξ_{ijk} and ξ_{qmn} are introduced to handle not satisfied constraints. In a nutshell, we are interested in finding the weight vector $\boldsymbol{\Theta}$ such that all the constraints on the target set are satisfied. Moreover we also impose some constraints on the source data that are similar to target classes in the MMD sense. Here ζ is a threshold parameter that controls the amount of information transferred from the source data. In practice we add a source constraint only if the discrepancy between the associated source classes and another one in the target data is low. The intuition behind the proposed approach is illustrated in Fig. 2. To solve the optimization problem in (3) we use an online learning approach [24]. The resulting algorithm is reported in Algorithm 1. In a nutshell, the algorithm performs stochastic gradient descent on the regularized version of the instantaneous loss while using a learning rate of $1/\lambda t$. Then the current weights vector is projected back to a feasible region $B = \{\boldsymbol{\Theta} : \boldsymbol{\Theta} \geq 0, \boldsymbol{\Theta} \leq 1/\lambda\}$.

Once $\boldsymbol{\Theta}$ is learned, a novel test sample $\mathbf{x}$ is classified according to a nearest neighbor rule, *i.e.* computing:

$$l = \arg \min_{l_i : (\mathbf{x}_i, l_i) \in \mathcal{T}} \boldsymbol{\theta}_{l_i}^T \mathbf{d}(\mathbf{x}, \mathbf{x}_i) \tag{4}$$

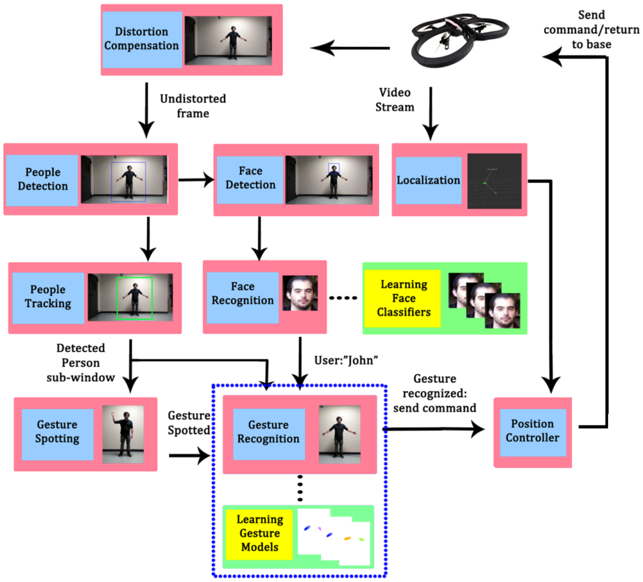


Fig. 3. System overview: the pink blocks correspond to processes running online while the green blocks are associated to operations performed offline. The core of our HRI system, *i.e.* the personalized gesture recognition module, is highlighted.

IV. HRI System with Parrot AR Drone 2.0

In this section we describe the integration of the proposed gesture recognition module into a HRI system. This system is made of several components that need to be able to communicate easily with each other. Therefore, we use ROS (Robot Operating System) [25] as it provides a communication framework that hides the complexities of transferring data between processes. In this work we employ a Parrot AR Drone 2.0 quadrocopter equipped with a 720p HD camera and a built-in attitude controller. Similarly to previous works [1], most of the computations are performed on an external computer which receives video streams along with control data via wireless network. This setup has the advantage of being easy to deploy and guarantees sufficient computational resources to perform the required tasks in real-time. On the other hand, it makes the system more vulnerable to network delays.

Figure 3 shows a block diagram representing the main components of our system. In the offline phase (green boxes), the face and the gesture recognition classifiers are learned. For gesture recognition we adopt the transfer learning algorithm described in the previous section. We consider five different gesture categories (Fig.4) and use as target training set two video sequences for each type of gesture corresponding to the same user. In our experiments four people are involved, thus four user-specific classifiers are learned. For each user the source training data are represented by video sequences of the other individuals as well as by web data obtained from the Keck Gesture Dataset [26] (further details are provided in Sec.5). During the offline phase for each user we record not only sample gestures but we employ the Viola-Jones face detector [27] to automatically collect a set of face images. Then we compute Local Binary Patterns Histograms (LPBH)

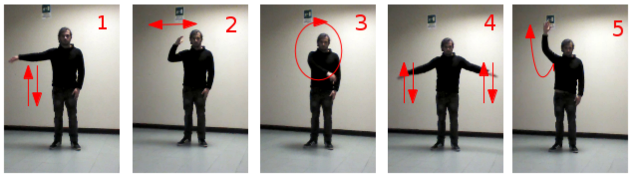


Fig. 4. Sample frames illustrating the 5 gestures categories of our dataset

[28] and use a Support Vector Machine (SVM) to implement the face recognition module.

The remaining system's blocks correspond to online operations. The AR Drone driver controls the UAV behavior. The visual streams collected from the front camera are sent to the computer and the distortion compensation node corrects the nonlinearities of incoming images using the intrinsic camera parameters. The resulting frames are made available to both the localization and the people detector modules. The localization is achieved through the well-known Parallel Tracking and Mapping (PTAM) system [29]. The vision-based estimates are fused with the inertial measurements from the drone using an Extended Kalman Filter to further improve pose estimation. To recover the scale factor we follow the approach in [30]. The UAV position is then sent to the drone controller node that consists of four different PID, one for each degree of freedom. When a new position commandis received, *i.e.* a new gesture is recognized, each PID computes and sends the controls to the UAV driver.

For people detection, the implementation in OpenCV based on Histograms of Oriented Gradient and SVM [31] is employed. The resulting detection windows are used by two subsequent modules: the people tracking and the face detection components. The first is realized with a simple Kalman Filter that uses the detector output in the measurement step. In this way the disturbance caused by the quadcopter movements are alleviated and an image window centered as much as possible around the user is obtained, improving the performance of the subsequent gesture recognition step. The face detection block employs the OpenCV Viola-Jones detector [27] and the output windows are provided to the face recognition module for user identification. In this block LBPH features are extracted and the previously trained SVM classifier is employed. The information about user identity is provided to the gesture recognition module which then select the corresponding user-specific model.

The cropped frames resulting from the people tracking module are then processed by the gesture spotting component. This module operates on the top two-thirds of the people tracking window and computes the dense optical flow vectors [21]. However, the UAV's continuous motion caused by the stabilization commands and the external turbulences, adds noise to flow estimation. Thus, similarly to [1], we compensate for the camera motion subtracting the median value of the computed optical flows (assuming it represents a good approximation of the camera ego-motion). After the motion compensation phase, the region of interest is divided into 2×3 blocks and for each block the average optical flow magnitude is computed. If in some blocks it exceeds a certain threshold a start signal is sent to the gesture recognition module.

The gesture recognition component is the core of our system. When a start signal is received, the system takes the optical flow vectors for a time-window of $T = 50$ frames. HOF are then computed at each frame and the Fisher Kernel descriptor for the entire sequence is calculated. Using the output of the face recognizer node, the appropriate gesture classifier is invoked and the gesture category is computed using (4). Once the gesture is recognized, a command is sent back to the AR Drone and the associated movement is executed.

V. Experimental Results

In this section we first describe the datasets used in our experiments, then we evaluate the proposed transfer learning approach for personalized gesture recognition and finally show our HRI system at work.

A. Datasets

To evaluate the performance of our gesture recognition approach we consider two datasets. The Keck gesture dataset [26] is used to build the source set $\mathcal{S}$. The dataset is composed by 14 different gesture classes (Fig. 5), collected using a 640 x 480 camera. Each gesture type is performed six times by three different people. Among the six repetitions, three are recorded with a static camera, while in the others the camera is moving and some background clutter and other moving objects are present. We also collect a new dataset with the parrot AR Drone, recording sequences of 4 different users while the quadcopter is flying. Five different gesture types are considered which represent a subset of the 14 classes of the Keck dataset: *turn left, attention left, start, flap, speed up*. Each user performs 10 repetitions for each gesture category. The previously described people detector and tracking modules have been used to collect the sequences of our dataset. To train the classifier of the face recognition module we also gather about 200 images per user employing the face detector in OpenCV.

B. Quantitative Evaluation

To demonstrate the effectiveness of our gesture recognition approach we conduct three series of experiments: two aiming to demonstrate the advantages of considering a transfer learning algorithm, the other showing that the chosen Fisher Kernel representation is a simple and effective approach to capture temporal variations in a video stream.

In the first series of experiments we compare six different methods: (i) a SVM classifier trained only with data corresponding to the 5 chosen gestures in the Keck datasets (*Keck-5*), (ii) a SVM classifier trained with sequences of our dataset but not containing samples of gestures of the current user (*Parrot (no User)*), (iii) a SVM classifier using both data from the 5 gesture types of Keck and data recorded from our AR Drone but without user's samples (*Keck-5 + Parrot (no User)*), (iv) a SVM classifier trained only with two

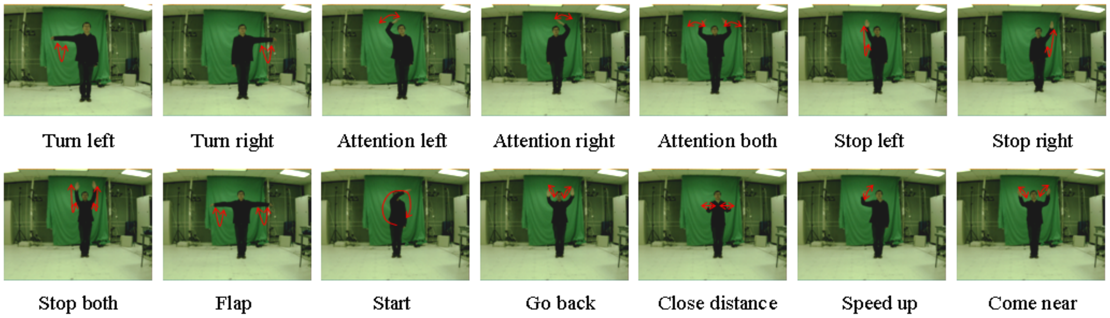


Fig. 5. Sample frames illustrating the 14 gesture categories of the Keck dataset

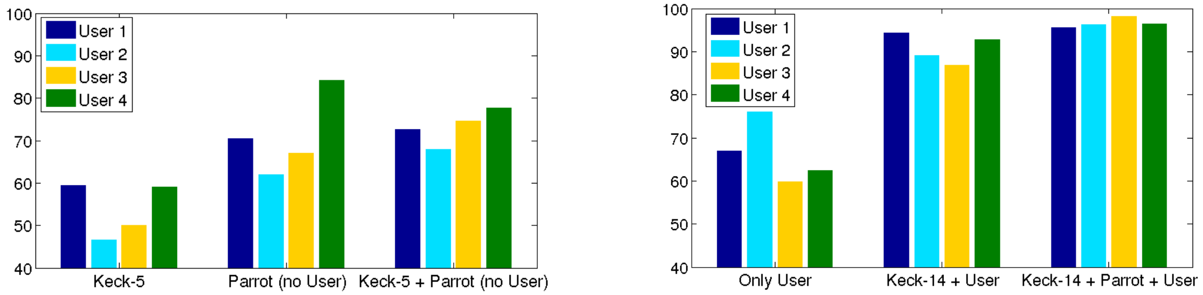


Fig. 6. Comparison of gesture recognition accuracy obtained with six different classifiers: the two right-most bar plots correspond to our transfer learning algorithm

sample gestures from the user currently interacting with the robot (*only User*), (v) the proposed transfer learning classifier using all the Keck dataset (14 gestures) plus two sequences of the chosen user (*Keck-14 +User*) and (vi) our approach when web data (14 gestures from the Keck dataset), two sequences of the chosen user and all the sequences from other users are employed for training (*Keck-14 +Parrot+User*). Both the regularization parameters of our approach and of the SVM classifiers are set with cross-validation. Linear SVM is considered in our tests. This ensures a fair comparison with the proposed approach which is also based on a linear classifier. We perform experiments for all the 4 users in our dataset, considering two randomly chosen samples per gesture as training data and the remaining eight for test. The average performance of twenty different runs are reported. Figure 6 clearly demonstrates the benefits of our method. A gesture recognition approach based only on data downloaded from the web (*Keck-5*) or on few sequences from the specific user (*only User*) lead to poor performance. Similarly using only data from other people (*Parrot (no User)*) also leads to unsatisfactory results as typically different users perform the same gesture in different ways. On the other hand our approach always guarantees recognition accuracies around 95%. For these experiments we also report the confusion matrices associated to user 3 and corresponding to the scenarios: *only User* (Fig. 7.a), *Keck-14 +User* (Fig. 7.b) and *Keck-14+Parrot+User* (Fig. 7.c).

A second series of experiments demonstrates the advantages of another important component of our gesture recognition method: the Fisher Kernel representation. We consider the six classifiers discussed above and we compare the proposed features representation with a simple accumulation scheme, *i.e.* constructing video sequence descriptors by simply summing up HOF features extracted from single frames. As shown in Table I the proposed feature representation ensures higher accuracy in all the tests. The performance are averaged over twenty different runs and four users.

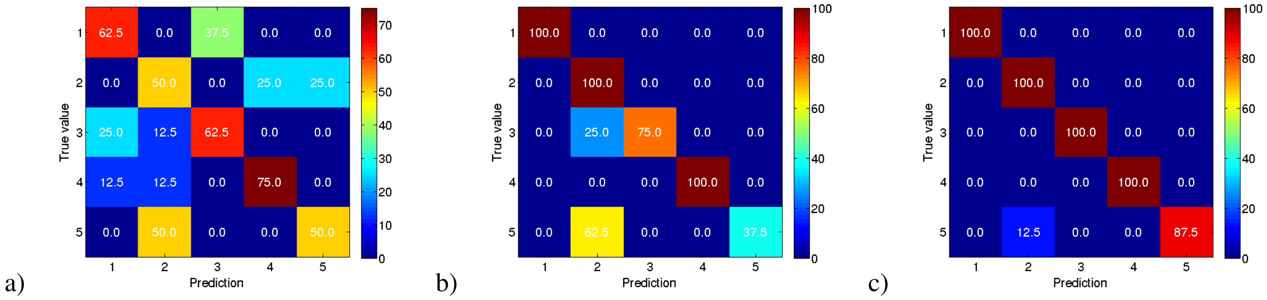
TABLE I
COMPARISON BETWEEN RECOGNITION ACCURACY OBTAINED WITH FEATURE ACCUMULATION AND FISHER KERNEL REPRESENTATION

	accumulation	Fisher Vector
SVM: Keck-5	50.31	53.78
SVM: Parrot (no User)	68.61	70.91
SVM: Keck-5 + Parrot (no User)	70.72	73.17
SVM: Only User	62.11	66.28
our approach: Keck-14 + User	86.31	90.74
our approach: Keck-14 + Parrot + User	93.2	96.56

TABLE II
COMPARISON BETWEEN RECOGNITION ACCURACY OBTAINED WITH OUR APPROACH AND DIFFERENT SOURCE SETS

	SVM	Our Approach
Keck-5 + User	88.2	89.6
Keck-5 + Parrot + User	93.2	94.87
Keck-14 + User	-	90.74
Keck-14 + Parrot + User	-	96.56
Keck-9 + User	-	68.22

Finally we discuss the advantages of our transfer framework which permits to deal with the general situation when source and target data may have different categories. This is a desirable characteristic for a system who wants to exploit data from the web and therefore must be able to select only relevant information. The results are reported in Table II and are averaged over twenty different runs and four users. We observe two interesting effects: first, with our approach a higher recognition accuracy is obtained when using not only 5 but all 14 gesture types of the Keck dataset. Through MMD



	1	2	3	4	5
1	62.5	0.0	37.5	0.0	0.0
2	0.0	50.0	0.0	25.0	25.0
3	25.0	12.5	62.5	0.0	0.0
4	12.5	12.5	0.0	75.0	0.0
5	0.0	50.0	0.0	0.0	50.0

	1	2	3	4	5
1	100.0	0.0	0.0	0.0	0.0
2	0.0	100.0	0.0	0.0	0.0
3	0.0	25.0	75.0	0.0	0.0
4	0.0	0.0	0.0	100.0	0.0
5	0.0	62.5	0.0	0.0	37.5

	1	2	3	4	5
1	100.0	0.0	0.0	0.0	0.0
2	0.0	100.0	0.0	0.0	0.0
3	0.0	0.0	100.0	0.0	0.0
4	0.0	0.0	0.0	100.0	0.0
5	0.0	12.5	0.0	0.0	87.5

Fig. 7. Confusion matrices associated to user 3 and corresponding to the scenarios (a) *only User*, (b) *Keck-14+User* and (c) *Keck-14+Parrot+User*

computation not only useless informations are discarded but motion patterns from other gestures categories help to increase classification accuracy. This is due to the fact that among the 14 gesture classes some are rather similar. Moreover, even in the *Keck-5* tests, we get higher results than a SVM classifier. Some users perform gestures with very different motion patterns with respect to others. Hence, even within the same categories, knowledge transfer is not the best choice. While we can handle this particular situation with our transfer learning framework, a SVM is not able to understand how much information should be transferred. Finally we apply our approach in the situation where the source set is made of sequences from the Keck dataset excluding data from the five selected categories (*Keck-9*). This represents the situation where source and target sets have completely different classes. Still in this case with our approach 68.22% of accuracy is achieved compared to 66.28% obtained with SVM *Only User* (Table I). We did not report SVM results for the *Keck-14* and *Keck-9* scenarios as source and target sets have different categories.

C. Demonstrations

Finally we show some video sequences demonstrating the integration of the proposed personalized gesture recognition module into a HRI system. Figure 8 depicts a user engaged into a one-on-one interaction with the AR Drone and performing a simple gesture, *i.e.* a circle. The system correctly recognizes it and the quadcopter executes the associated command. In this case it simply flies performing a square trajectory in the air and goes back to the original position. In our implementation running on a 2.4 Quad-Core processor, the system requires about 2 seconds for recognizing a single gesture and the main bottleneck is represented by the optical flow computation. A video showing the complete system at works is provided as supplementary material.

VI. CONCLUSIONS

We present one of the first attempts toward the realization of a visual based HRI system for flying robots. Differently from previous works, in this paper we specifically focus on the development of a personalized gesture recognition system and we demonstrate that adopting a transfer learning approach is greatly beneficial in terms of recognition accuracy. We also show that the chosen Fisher Kernel descriptors are suitable to describe gesture sequences representing the variations over time of HOF features. Future works will be devoted to improve our system still facing the many challenges of realizing HRI solutions tailored to UAVs, *e.g.* copying with the problem of network delays, improving system localization accuracy, the user tracking module [32], moving to outdoor scenarios, addressing the problem of gesture recognition with cluttered background and surrounding moving objects. Moreover the gestures dataset will be improved with more users and different gesture categories to further evaluate the effectiveness of our framework.

ACKNOWLEDGMENT

This research has been partly funded by the MIUR Smart Cities and Communities and Social Innovation program, under the grant SEAL.

REFERENCES

[1] V. Monajjemi, J. Wawerla, R. Vaughan, and G. Mori, “Hri in the sky: Creating and commanding teams of uavs with a vision-mediated gestural interface,” in *Intelligent Robots and Systems (IROS)*, 2013.

[2] T. Naseer, J. Sturm, and D. Cremers, “Followme: Person following and gesture recognition with a quadrocopter,” in *Intelligent Robots and Systems (IROS)*, 2013.

[3] M. Lichtenstern, M. Frassl, B. Perun, and M. Angermann, “A prototyping environment for interaction between a human and a robotic multi-agent system,” in *Human-Robot Interaction (HRI)*, 2012.

[4] B. R. Duffy, “Anthropomorphism and the social robot,” *Robotics and autonomous systems*, vol. 42, no. 3, pp. 177–190, 2003.

[5] I. Maza, F. Caballero, R. Molina, N. Peña, and A. Ollero, “Multimodal interface technologies for uav ground control stations,” *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 371–391, 2010.

[6] Y. Song, D. Demirdjian, and R. Davis, “Tracking body and hands for gesture recognition: Natops aircraft handling signals database,” in *Automatic Face & Gesture Recognition and Workshops (FG)*, 2011.

[7] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, “A survey of video datasets for human action and activity recognition,” *Comput. Vis. Image Underst.*, vol. 117, no. 6, pp. 633–659, 2013.

[8] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez, “View-independent action recognition from temporal self-similarities,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 172–185, 2011.

[9] C. R. A. Yilmaz and M. Shah, “View-invariant representation and recognition of actions.” *International Journal of Computer Vision*, vol. 50, no. 2, pp. 203–226, 2002.

[10] A. Farhadi and M. K. Tabrizi, “Learning to recognize activities from the wrong view point,” in *European Conference on Computer Vision (ECCV)*, 2008.

[11] R. Li, “Discriminative virtual views for cross-view action recognition,” in *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[12] J. Liu, M. Shah, B. Kuipers, and S. Savarese, “Cross-view action recognition via view knowledge transfer,” in *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[13] Y. Yan, E. Ricci, G. Liu, R. Subramanian, and N. Sebe., “Clustered multi-task linear discriminant analysis for view invariant color-depth action recognition,” in *International Conference on Pattern Recognition (ICPR)*, 2014.

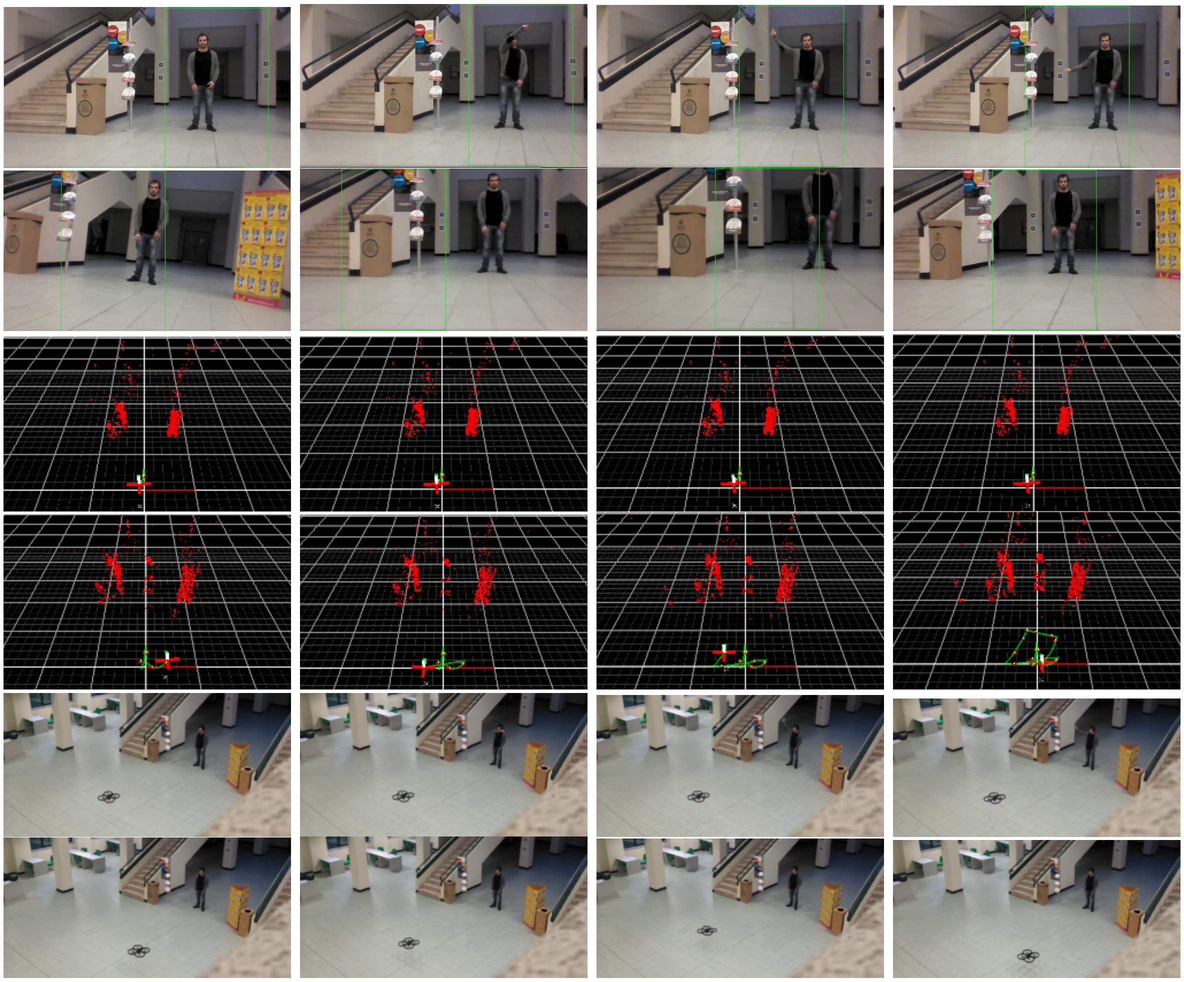


Fig. 8. Sample frames illustrating the integration of our gesture recognition module into a HRI system: the user performs the gesture *start* (circle), the AR drone receives a command and flies performing a square trajectory in the air. The first two rows show the output of the people detector module which processes the images from the front camera of the drone. The two central rows show the output of the PTAM algorithm. In the two bottom rows an external camera captures the interaction of the user with the AR drone.

[14] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. Montiel, A. Perzylo, B. Schiessle, M. Tenorth, O. Zweigle, and R. van de Molengraft, "Roboearth," *IEEE Robotics Automation Magazine*, vol. 18, no. 2, pp. 69–82, 2011.

[15] T. Moritz, K. Ulrich, P. Dejan, and B. Michael, "Web-enabled robots – robots that use the web as an information resource," *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 58–68, 2011.

[16] M. Samadi, T. Kollar, and M. Veloso, "Using the web to interactively learn to find objects." in *Artificial Intelligence (AAAI)*, 2012.

[17] G. Costante, T. A. Ciarfuglia, P. Valigi, and E. Ricci, "A transfer learning approach for multi-cue semantic place recognition," in *Intelligent Robots and Systems (IROS)*, 2013.

[18] J. Luo, A. Pronobis, and B. Caputo, "Svm-based transfer of visual knowledge across robotic platforms," in *International Conference on Computer Vision Systems (ICVS)*, 2007.

[19] S. P. Elango, T. Tommasi, and B. Caputo, "Transfer learning of visual concepts across robots: a discriminative approach," Idiap, Tech. Rep. Idiap-RR-06-2012, 2012.

[20] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European Conference on Computer Vision (ECCV)*, 2010.

[21] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian Conference on Image Analysis (SCIA)*, 2003.

[22] I. Mironica, J. Uijlings, N. Rostamzadeh, B. Ionescu, and N. Sebe, "Time matters!: capturing variation in time in video using fisher kernels," in *ACM International Conference on Multimedia*, 2013.

[23] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schlkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," in *Intelligent Systems for Molecular Biology (ISMB)*, 2006.

[24] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for svm," *Mathematical Programming*, vol. 127, no. 1, pp. 3–30, 2011.

[25] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," 2009.

[26] Z. Lin, Z. Jiang, and L. S. Davis, "Recognizing actions by shape-motion prototype trees." in *Internation Conference on Computer Vision (ICCV)*, 2009, pp. 444–451.

[27] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition (CVPR)*, 2001.

[28] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *European Conference on Computer Vision (ECCV)*, 2004, pp. 469–481.

[29] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[30] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *Intelligent Robots and Systems (IROS)*, 2012.

[31] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[32] S. Duffner, J.-M. Odobez, and E. Ricci, "Dynamic partitioned sampling for tracking with discriminative features," 2009.