2nd IFAC Workshop on Research, Education and Development of
Unmanned Aerial Systems
November 20-22, 2013. Compiegne, France

Real-time Adaptive Multi-Classifier Multi-Resolution Visual Tracking Framework for Unmanned Aerial Vehicles

Changhong Fu*, **Ramon Suarez-Fernandez***
Miguel A. Olivares-Mendez**, **Pascual Campoy***

* *Computer Vision Group(CVG), Centro de Automática y Robótica(CAR), UPM-CSIC, Universidad Politécnica de Madrid, José Gutiérrez Abascal 2, 28006 Madrid, Spain, (E-mail: fu.changhong@upm.es).*
** *Automation Research Group, Interdisciplinary Centre for Security, Reliability and Trust, SnT-University of Luxembourg, 4 rue Alphonse Weicker, 2721 Luxembourg.*

Abstract: This paper presents a novel robust visual tracking framework, based on discriminative method, for Unmanned Aerial Vehicles (UAVs) to track an arbitrary 2D/3D target at real-time frame rates, that is called the Adaptive Multi-Classifier Multi-Resolution (AMCMR) framework. In this framework, adaptive Multiple Classifiers (MC) are updated in the (k-1)th frame-based Multiple Resolutions (MR) structure with compressed positive and negative samples, and then applied them in the kth frame-based Multiple Resolutions (MR) structure to detect the current target. The sample importance has been integrated into this framework to improve the tracking stability and accuracy. The performance of this framework was evaluated with the Ground Truth (GT) in different types of public image databases and real flight-based aerial image datasets firstly, then the framework has been applied in the UAV to inspect the Offshore Floating Platform (OFP). The evaluation and application results show that this framework is more robust, efficient and accurate against the existing state-of-art trackers, overcoming the problems generated by the challenging situations such as obvious appearance change, variant illumination, partial/full target occlusion, blur motion, rapid pose variation and onboard mechanical vibration, among others. To our best knowledge, this is the first work to present this framework for solving the online learning and tracking freewill 2D/3D target problems, and applied it in the UAVs.

Keywords: Unmanned Aerial Vehicles(UAVs), Discriminative Visual Tracking(DVT), Hierarchical Tracking Strategy(HTS), Online Appearance Learning(OAL), Compressive Visual Sensing(CVS), Adaptive Algorithm, Robot Navigation.

1. INTRODUCTION

Visual object tracking has been researched and developed fruitfully in the robot community recently. However, real-time robust visual tracking for arbitrary 2D/3D targets (also referred to visual *model-free* tracking), especially in Unmanned Aerial Vehicle (UAV) control and navigation applications, remains a challenging task due to significant appearance change, variant illumination, partial/full target occlusion, blur motion, rapid pose variation, and onboard mechanical vibration, among others.

The typical visual tracking system/framework consists of three components (Babenko et al. (2011), Yilmaz et al. (2006)): (I) the appearance model, which can evaluate the likelihood that the target is at some particular locations; (II) the motion model, which relates the locations of the target over time; (III) the search strategy, which is applied for finding the most likely location in the current frame. This paper proposes a new system/framework in order to provide the stable and accurate visual information to control and navigate the UAVs in the civil applications, e.g. autolanding, power tower and offshore floating platform inspection, volcano monitoring, people and car following et al, as shown in the Figure 1.

In the literatures, many visual trackers have obtained the promising tracking performances for specific/arbitrary objects. Considering on their appearance model representation schemes, they can be generally categorized as either *generative* or *discriminative*-based visual trackers. Generative tracking method (Felzenszwalb et al. (2010), Ramanan et al. (2007), Ross et al. (2008), Balan and Black (2006)) typically adopts a static model that is manually selected or trained only on the first frame to represent the target, and then obtains the target location based on the minimal reconstruction error in the current frame region. Martinez et al. (2013) utilized the direct method (i.e. directly represent the target using all the pixels in the selected and fixed image template) to track target. Mejias et al. (2006) selected the target model on the first frame using Lucas-Kanade tracker. Mei and Ling (2011)

978-3-902823-57-1/2013 © IFAC

10.3182/20131120-3-FR-4045.00010

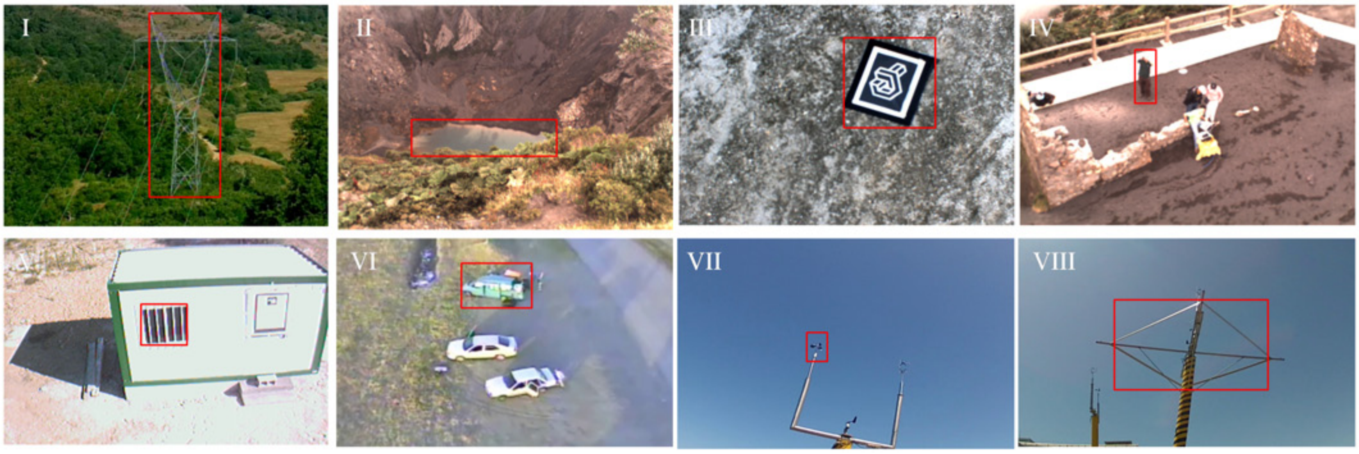

Fig. 1. Visual Object Tracking using AMCMR-CT Framework, where, each image represents the different inspection task: (**I**) Power Tower; (**II**) Volcano; (**III**) Helipad; (**IV**) Pedestrian; (**V**) Window; (**VI**) Car; (**VII**) Sensor; (**VIII**) Mast.

modeled the target as a sparse linear combination of target and trival templates, i.e. l_1- tracker. But these feature and direct method-based algorithms are unstable to track the targets with obvious appearance changes and other challenging factors mentioned above, and they did not use the valuable background information to improve the tracking performances (Wang et al. (2011)).

While discriminative algorithms (also called visual *tracking-by-detection* methods (Avidan (2004))) employ an adaptive binary classifier to separate the target from background during the frame-to-frame tracking (Zhang et al. (2013), Collins et al. (2005), Grabner and Bischof (2006), Kalal et al. (2012)). The important stage for discriminative visual tracking is *classifier update* using online selected features, many works update their classifiers using only one positive sample and some surrounding negative samples, but this method often causes the tracking drift (failure) problem becasue of noisy and misaligned samples. Recently, both multiple positive samples and negative samples are used to update classifier, the location of sample with maximum classifier score (i.e. most *correct/important* sample) is the new target location at current frame, this method can even solve significant appearance changes in cluttered background. However, these discriminative algorithms did not take into account the information about the importance of the sample, i.e. classifier score for each sample, to improve the tracking stability and accuracy.

Although many discriminative approaches often achieve superior tracking results, and tolerate the motions in the range of search radius, but in the tracking on-board UAV for control and navigation, we have observed that discriminative visual tracking algorithms are sensitive to strong motions (e.g. onboard mechanical vibration) or large displacements over time. Therefore, we adopt Multi-Resolution (MR) strategy to cope with these problems. Additionlly, this strategy can help to deal with the problems that are the onboard low computational capacity and information communication delays between UAV with Ground Station. Thus, this paper proposes to address the UAV tracking problem using discriminative visual tracker named Compressive Tracking (CT), and extending it in a hierarchy-based framework with different image resolutions and adaptive classifiers applied to estimate the motion models in these resolutions: the Adaptive Multi-Classifier Multi-Resolution framework (AMCMR). Using this strategy, especially in the Adaptive Multi-Classifier structure, the importance of the sample has been used to reject samples, i.e. the lower resolution features are initially applied in rejecting the majority of samples (called Rejected Samples (RS)) at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions, thereby ensuring the real-time performance and higher accuracy.

To the author's best knowledge, this framework has not been presented for solving the online learning and tracking freewill 2D/3D target problems in the UAV. The proposed AMCMR-CT framework runs at real-time and performs favorably on challenging public and aerial image sequences in terms of efficiency, accuracy and robustness. For this reason, the intention of this paper is also to expand this discriminative method-based framework in more real-time UAV control and navigation applications.

The outline of the paper is organized as follows: In Section 2, we introduced the Discriminative Visual Tracking (DVT) concept and the Compressive Tracking (CT) algorithm. Section 3 proposed the details of AMCMR-CT visual tracking framework and its configurations. The evaluation performance results are presented in Section 4 using public image datasets and aerial image databases. Then, a UAV application using this proposed framework is shown, and its performance results are given and discussed in Section 5. Finally, the concluding remarks and future work are presented in the Section 6.

2. DISCRIMINATIVE VISUAL TRACKING

2.1 Preliminaries

Discriminative Visual Tracking (DVT) takes the tracking problem as a binary classification task to separate target from its surrounding background, as shown in Figure 2.

It trains a classifier in an online method using positive and negative samples extracted from the current frame. When the next frame is coming, the samples around the

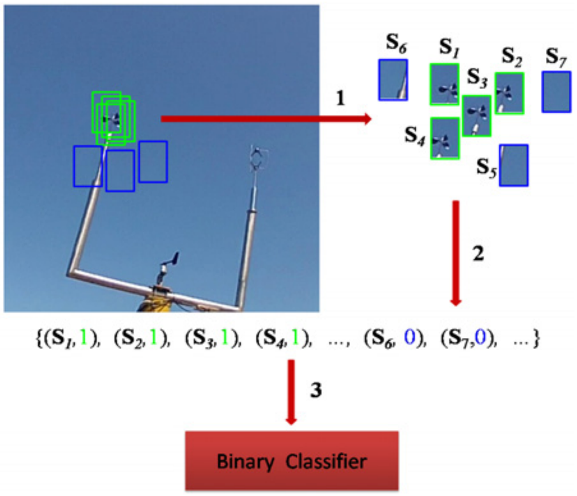
S_6
S_1
S_2
S_7
S_3
S_4
S_5
1
2
3
$\{(S_1, 1), (S_2, 1), (S_3, 1), (S_4, 1), \ldots, (S_6, 0), (S_7, 0), \ldots\}$
Binary Classifier

Fig. 2. Discriminative visual tracking, where, tracking onboard UAV for the sensor at real-time, the image with green rectangle represents the positive sample, while the one with blue rectangle is the negative sample.

old target location are extracted at this frame, and then the afore-trained classifier is applied to these samples. The location of the sample with the maximum classifier score is the new target location at this frame. A generic process of the DVT is presented in the Algorithm 1.

Algorithm 1 Discriminative Visual Tracking.

Input: the k-th frame

1. Extract a set of image samples: $\mathcal{S}^{\alpha}=\{\mathbf{S}|\|\mathbf{l}(\mathbf{S})-\mathbf{l}_{k-1}\| < \alpha\}$, where, $\mathbf{l}_{k-1}$ is the target location at (k-1)th frame, and online select feature vectors.
2. Use classifier trained in the (k-1)th frame to these feature vectors and find the target location $\mathbf{l}_k$ with the maximum classifier score.
3. Extract two sets of image samples: $\mathcal{S}^{\beta}=\{\mathbf{S}|\|\mathbf{l}(\mathbf{S})-\mathbf{l}_{k}\| < \beta\}$ and $\mathcal{S}^{\gamma,\delta}=\{\mathbf{S}|\gamma < \|\mathbf{l}(\mathbf{S})-\mathbf{l}_{k}\| < \delta\}$, where, $\beta < \gamma < \delta$.
4. Online select the feature using these two sets of samples, and update the classifier.

Output: (1) Target location $\mathbf{l}_k$
(2) Classifier trained in the kth frame

In the Algorithm 1, the parameter α is called search radius, which is used to extract the test samples in the k-th frame, the parameter β is the radius applied for extracting the positive samples, while the parameter γ and δ are the inner and outer radii, which are used to extract the negative samples.

2.2 Compressive Tracking (CT)

As introduced in above sections, updating classifier depends on online selecting features, Collins et al. (2005) demonstrated that the most discriminative features can be learned online. In this paper, we adopt an effective and efficient DVT-based Compressive Tracking (CT) algorithm proposed by Zhang et al. (2012), which selects the features in the compressed domain, as shown in the Figure 3. The CT runs at real-time frame rates and outperforms the existing state-of-art visual trackers. In addition, the positive and negative samples are compressed with the same data-independent sparse measurement matrix discriminated by a simple naive Bayes classifier.

The CT mainly contributes three aspects, as shown in Figure 3: (I)*Selecting*: select image feature vector from each positive/negative sample; (II) *Compressing*: compress these feature vectors to the low-dimensional feature vectors; (III) *Updating*: update classifier with these low-dimensional feature vectors.

1. Selecting As the Part 1 and 2 in the dash line rectangle, for each sample $\mathbf{S} \in \mathbb{R}^{\underline{w} \times \underline{h}}$, it is processed with a set of rectangle filters at multiple scales, $\{h_{1,1}, ..., h_{\underline{w},\underline{h}}\}$ defines as

$$h_{i,j}(x,y) = \begin{cases} 1, & 1 \leqslant x \leqslant i, 1 \leqslant y \leqslant j \\ 0, & otherwise \end{cases} \quad (1)$$

where i and j are the width and height of a rectangle filter, respectively. Then, each filtered image is represented as a colum vector in $\mathbb{R}^{\underline{w} \times \underline{h}}$, and then these vectors are concatenated as a very high-dimensional multi-scale image feature vector $\mathbf{x} = (x_1, ..., x_m)^T \in \mathbb{R}^m$, where $m = (\underline{w} \times \underline{h})^2$. The dimensionality m is typically in the order of 10^6 to 10^{10}.

2. Compressing After obtained the high-dimensional multi-scale image feature vector $\mathbf{x} \in \mathbb{R}^m$, as shown in Part 3 and 4, the random matrix $R \in \mathbb{R}^{n \times m}$ is used to compress it to a lower-dimensional vector $\mathbf{v} \in \mathbb{R}^n$

$$\mathbf{x} \xrightarrow{\times R} \mathbf{v}, \qquad i.e. \mathbf{v} = R\mathbf{x} \quad (2)$$

where $n \ll m$. In the Equation 2, a very sparse random Gaussian matrix $R \in \mathbb{R}^{n \times m}$, where $r_{ij} \sim N(0,1)$, with entries is defined as

$$r_{ij} = \sqrt{s} \times \begin{cases} 1 & with\ possibility \ \frac{1}{2s} \\ 0 & with\ possibility \ 1 - \frac{1}{s} \\ -1 & with\ possibility \ \frac{1}{2s} \end{cases} \quad (3)$$

where, when $s = 3$, it is very sparse where two thirds of the computation can be avoided.

From Equation 3, it is only necessary to store the nonzero entries in R and the positions of rectangle filters in an input image corresponding to the nonzero entries in each row of R. Then, $\mathbf{v}$ can be efficiently computed by using R to sparsely measure the rectangle features which can be efficiently computed using the integral image method, where, the compressive features compute the relative intensity difference in a way similar to the generalized Haar-like features. The compressive sensing theories ensure that the extracted features of CT algorithm preserve almost all the information of the original image.

3. Updating For each sample $\mathbf{S} \in \mathbb{R}^{\underline{w} \times \underline{h}}$, its low-dimensional representation is $\mathbf{v} = \{v_1, ..., v_n\} \in \mathbb{R}^n$ with $m \gg n$. All elements in $\mathbf{v}$ are assumed to be independently distributed and modeled with a naive Bayes classifier,

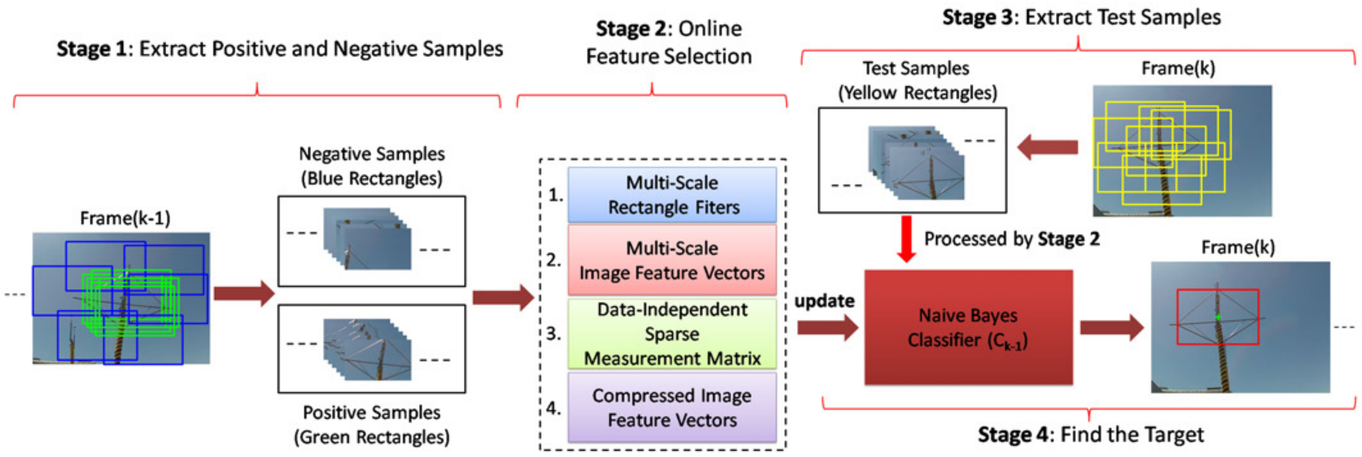
Stage 1: Extract Positive and Negative Samples
Stage 2: Online Feature Selection
Stage 3: Extract Test Samples
Frame(k-1)
Negative Samples (Blue Rectangles)
Positive Samples (Green Rectangles)
1. Multi-Scale Rectangle Fiters
2. Multi-Scale Image Feature Vectors
3. Data-Independent Sparse Measurement Matrix
4. Compressed Image Feature Vectors
Test Samples (Yellow Rectangles)
Frame(k)
Processed by Stage 2
update
Naive Bayes Classifier (C$_{k-1}$)
Frame(k)
Stage 4: Find the Target

Fig. 3. Compressive Tracking (CT). The adaptive classifier is updated with compressed image feature vectors in the (k-1)th frame, and then applied to estimate the target location in the kth frame.

$$H(\mathbf{v}) = \log\left(\frac{\prod_{i=1}^{n} p(v_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(v_i|y=0)p(y=0)}\right) = \sum_{i=1}^{n} \log\left(\frac{p(v_i|y=1)}{p(v_i|y=0)}\right) \quad (4)$$

where, uniform prior are assumed: $p(y=1) = p(y=0)$, and $y \in \{0,1\}$ is a binary variable which represents the sample label.

The conditional distributions $p(v_i|y=1)$ and $p(v_i|y=0)$ in the classifier $H(\mathbf{v})$ are assumed to be Gaussian distributed with four parameters $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$ where

$$p(v_i|y=1) \sim N(\mu_i^1, \sigma_i^1), \quad p(v_i|y=0) \sim N(\mu_i^0, \sigma_i^0) \quad (5)$$

The scalar parameters in Equation (5) are incrementally updated

$$\mu_i^1 \leftarrow \eta\mu_i^1 + (1-\eta)\mu^1$$
$$\sigma_i^1 \leftarrow \sqrt{\eta(\sigma_i^1)^2 + (1-\eta)(\sigma^1)^2 + \eta(1-\eta)(\mu_i^1 - \mu^1)^2} \quad (6)$$

where, $0 < \eta < 1$ is a learning parameter,

$\sigma^1 = \sqrt{\frac{1}{n}\sum_{k=0|y=1}^{n-1}(v_i(k) - \mu^1)^2}$ and

$\mu^1 = \frac{1}{n}\sum_{k=0|y=1}^{n-1} v_i(k)$.

The update schemes for μ_i^0 and σ_i^0 have similar formations.

3. HIERARCHY-BASED TRACKING STRATEGY

3.1 Hierarchy-based Tracking

In the UAV tracking applications, as mentioned in the Section 1, compressive tracking is also sensitive to the strong motions or large displacements. Although the search radius for extracting test samples can be set to be larger, as shown in Algorithm 1, to get more tolerance for these problems, however, more test samples will be generated, which influence the real-time and accuracy performances. Therefore, Multi-Resolution (MR) approach was proposed to deal with these problems, as shown in Figure 4. Nevertheless, there must be a compromise between the number of levels required to overcome the large inter-frame motion and the amount of visual information required to update the adaptive multiple classifiers for estimating the motions.

3.2 Configurations

1. Number of Pyramid Levels (N_{PL}) Considering the images are downsampled by a ratio factor 2, the Pyramid Levels of the MR structure are defined as a function below:

$$N_{PL} = \lfloor log_2 \frac{min\{\mathbf{T}_W, \mathbf{T}_H\}}{minSizes} \rfloor \quad (7)$$

where, $\lfloor * \rfloor$ is the largest integer not greater than value $*$, $\mathbf{T}_W$, $\mathbf{T}_H$ represent the width and height of target $\mathbf{T}$ in the highest resolution image (i.e. the lowest-level of pyramid: 0 level), respectively. And $minSizes$ is the minimum size of target in the lowest resolution image (i.e. the highest-level of pyramid: p_{max} level, $p_{max} = N_{PL}$-1), in order to have enough information to estimate the motion model in that level. Thus, if the $minSizes$ is set in advanced, the N_{PL} directly depends on the width/height of tracking target $\mathbf{T}$.

2. Motion Model ($\mathbf{l}$) Propagation Taking into account that the motion model estimated by CT in each level is used as the initial estimation of motion for the next higher resolution image, therefore, the motion model propagation is defined as follows:

$$\mathbf{l}_k^{p-1} = 2\mathbf{l}_k^p \quad (8)$$

where, p represents the pth level of the pyramid, $p = \{p_{max}, p_{max}-1, ..., 0\} = \{N_{PL}-1, N_{PL}-2, ..., 0\}$, and k is the kth frame.

3. Number of Rejected Sample (N_{RS}) In the Adaptive Multi-Classifier structure, since the MR approach provides the computational advantage to analyze features and update classifiers in low resolution images, the majority of samples will be rejected based on their classifier scores (i.e. sample importances) in the lower resolution image, leaving a fewer number of samples to be processed in the higher resolution image. Thus, the AMC structure obtains higher tracking speed, better accuracy than a single full resolution-based adaptive classifier, the rejected sample number is defined as:

$$N_{RS}^p = \xi^p N_S^p \quad (9)$$

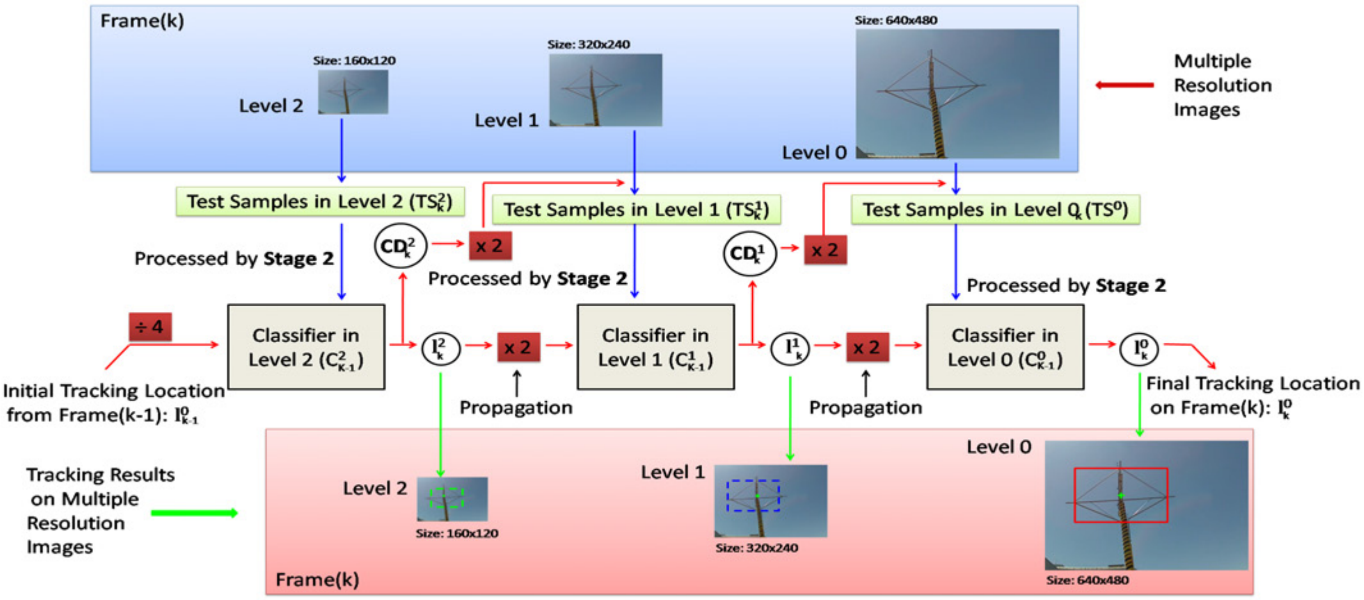
Frame(k)
Size: 160x120
Level 2
Size: 320x240
Level 1
Size: 640x480
Level 0
Multiple Resolution Images
Test Samples in Level 2 (TS²ₖ)
Test Samples in Level 1 (TS¹ₖ)
Test Samples in Level 0 (TS⁰)
Processed by Stage 2
CD²ₖ
x 2
CD¹ₖ
÷ 4
Classifier in Level 2 (C²ₖ₋₁)
Classifier in Level 1 (C¹ₖ₋₁)
Classifier in Level 0 (C⁰ₖ₋₁)
I²ₖ
I¹ₖ
I⁰ₖ
Propagation
Initial Tracking Location from Frame(k-1): I⁰ₖ₋₁
Final Tracking Location on Frame(k): I⁰ₖ
Tracking Results on Multiple Resolution Images
Size: 160x120
Size: 320x240
Size: 640x480
Frame(k)

Fig. 4. AMCMR-CT visual tracking framework. The kth frame is downsampled to create the MR structure. In Adaptive Multi-Classifier structure, lower resolution features are initially used to reject the majority of samples at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions. The C^p_{k-1} represents the adaptive classifier updated in the pth level of pyramid of (k-1)-th frame.

where, p represents the pth level in the pyramid, ξ^p is the reject ratio ($0 < \xi^p < 1$), and N^p_S is the number of test samples. Especially, the sample with maximum score in the rejected samples is the Critical Sample (CS^p_k).

4. Search Radius Propagation The euclidean distance between the location of CS^p_k and $\mathbf{I}^p_k$ is the Critical Distance (CD^p_k), which will be propagated to next higher resolution image as the search radius:

$$\alpha^{p-1}_k = 2CD^p_k \tag{10}$$

where, p represents the pth level in the pyramid, and k is the kth frame.

4. EXPERIMENT EVALUATION

In this section, we compared our AMCMR-CT tracker with 2 latest state-of-art trackers (TLD(Kalal et al. (2012)) and CT) on two different types of challenging image data: (I) Public Image Datasets, which are used to test and compare visual algorithms in computer vision community; (II) Aerial Image Databases, which are captured from our former vision-based UAV inspection projects. The evaluation measure (Bai and Li (2012)) is the Center Location Error (CLE), which is defined as the Euclidean distance from the detected target center to the (manually labeled) ground truth center at each frame.

4.1 Test 1: Comparision with Public Image Datasets

The most challenging public image datasets have been applied. The total number of evaluated frames is more than 10^4. Here, the tracking performance of *Girl* image dataset released by Birchfield[1] is shown below. This sequence contains one main challenging factor that is full 360-degree out-of-plane rotation.

During the tracking process, as shown in Figure 5 and 6, although TLD tracker is able to relocate on the target, it is easy to lose the target completely for many frames. For the CT, it can track its target, but the tracking drift generates easily in some frames when the target is out of plane, changing scale , jumping suddenly and occluded by other people. However, the AMCMR-CT outperforms these two trackers.

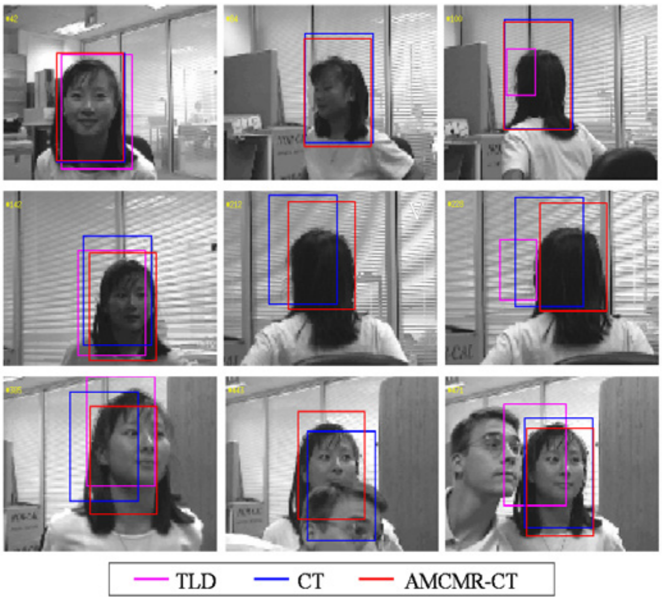

Fig. 5. Tracking *Girl* using TLD (Pink), CT (Blue) and AMCMR-CT (Red) algorithms.

[1] http://www.ces.clemson.edu/∼stb/

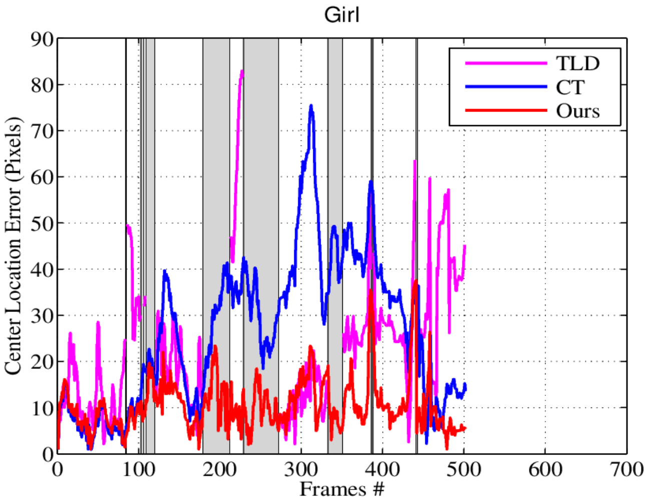

Fig. 6. The Performances of Visual Tracking for *Girl*, where, the Grey Shadow Area represents the lost measurements using TLD tracker.

4.2 Test 2: Comparision with Real Flight-based Aerial Image Databases

The real flight-based aerial images captured by different UAVs are processed. The evaluated frame number is more than 8000. Here, one aerial image database for tracking *window* recorded by CVG-UPM [2] is tested. This database includes one obvious challenging factor that is the strong motion or large displacements over time.

For the tracking performances, as shown in Figure 7 and 8, TLD can finish its tracking task in this sequence, and sometimes relocate its target when the appearance of target is similar to the initialized target appearance, but it often misaligns its target, and the misaligned error is larger than those generated by the CT and AMCMR-CT. Although the CT can tolerate the motions to some extent based on the range of searching radius, its tracking performance is not superior to the AMCMR-CT's performance.

5. VISUAL INSPECTION APPLICATION

The different evaluations described in the Section 4 have shown that the proposed AMCMR-CT framework is able to track arbitrary 2D/3D targets under different challenging conditions and obtains the better performances. In this section, this new algorithm is used in a real application, i.e. OMNIWORKS Project [3], to control and navigate UAV.

5.1 Offshore Floating Platform (OFP) Inspection

Offshore Floating Platform (OFP) and its sensors are the targets for UAV inspection in this application. It aims to replace the Engineer to check OFP and its sensors to reduce the risks, time, equipments, inspection costs and et al. Figure 9 shows the simulation platforms near the harbour, where, the Left OFP in image **I** is the static platform to simulate the sea without (or with small) wave, while the right OFP is the moving platform to simulate the ocean with different wave levels. Different types of sensors should be inspected as shown in image **II** and **III**. And the Asctec Pelican [4] is used to track these targets using AMCMR-CT.

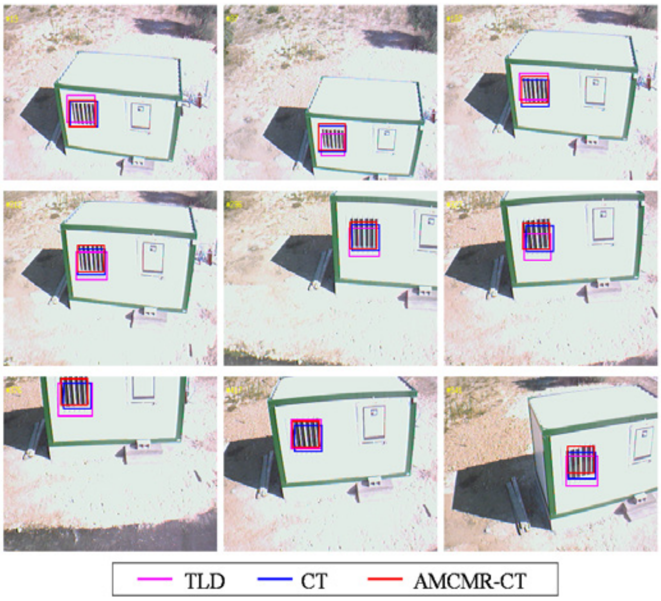

Fig. 7. Tracking *Window* using TLD (Pink), CT (Blue) and AMCMR-CT (Red) algorithms.

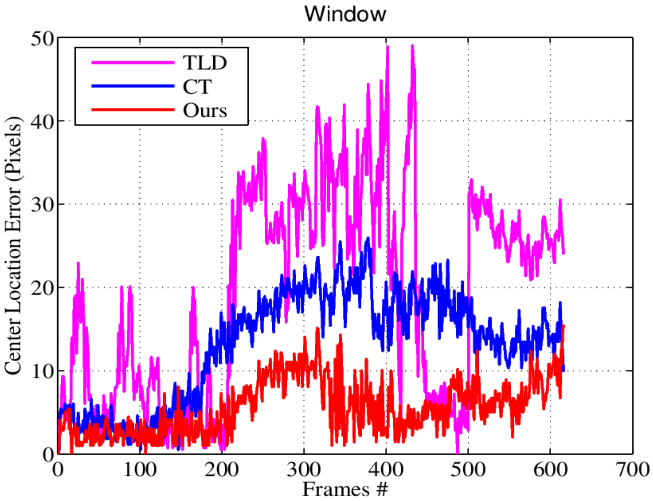

Fig. 8. The Performances of Visual Tracking for *Window*.

5.2 Tracking Performance Analysis

In this subsection, the TLD and CT trackers are applied to compare with our AMCMR-CT tracker in two different inspection tasks for OFP, the tracking targets inlcude: (I) Anemometer; (II) Moving mast.

1. Anemometer Tracking The anemometer is used to measure the speed of wind. The main challenging factors for anemometer tracking in UAV are obvious appearance change and strong motions.

In the anemometer tracking, the TLD completely lose its target from the 2nd frame. Although the CT tracks its

[2] http://www.vision4uav.com

[3] http://www.echord.info/wikis/website/omniworks

[4] http://www.asctec.de/

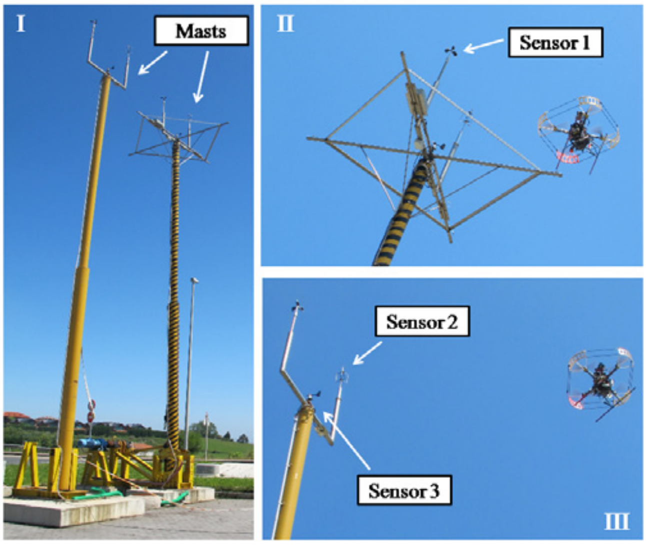

Fig. 9. One UAV application: Offshore Floating Platform (OFP) Inspection.

target well at the beginning, the drift problem occurs from the 51th frame, and then lose its target, as shown in Figure 10. While the AMCMR-CT obtains the best performance.

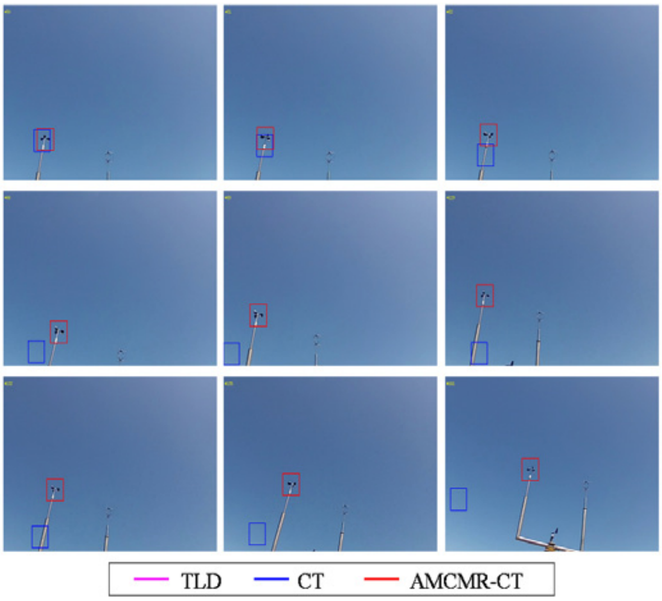

Fig. 10. Tracking *Sensor* using TLD (Pink), CT (Blue) and AMCMR-CT (Red) algorithms.

2. Moving Mast Tracking The mast is applied to fix the different sensors. This aerial image dataset includes one main challenging factor that is strong motions or large displacements.

For the moving mast tracking, from the 4th frame, the TLD completely lose its target again. The CT and AMCMR-CT both can finish their mast tracking, however, the CT still has the drift problem, its tracking accuracy is worse than the one AMCMR-CT obtained.

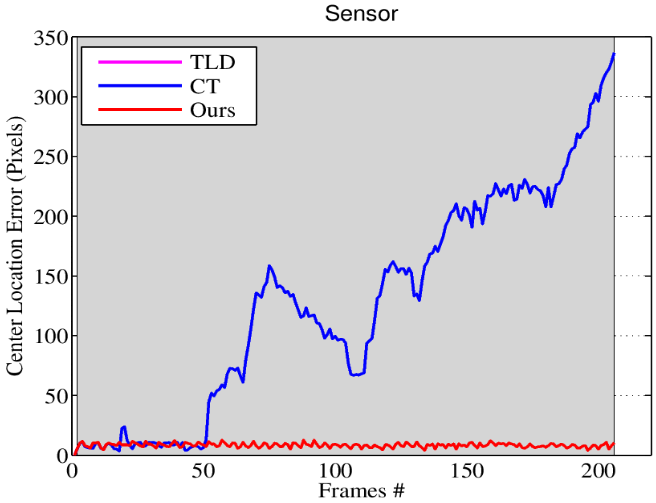

Fig. 11. The Performance of Visual Tracking for *Sensor*, where, the Grey Shadow Area represents the lost measurements using TLD tracker.

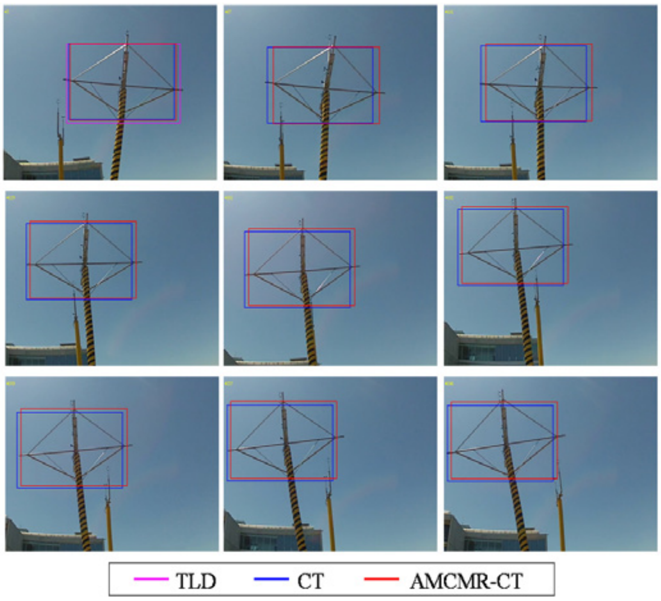

Fig. 12. Tracking *Mast* using TLD (Pink), CT (Blue) and AMCMR-CT (Red) algorithms.

The Center Location Error (CLE)(in pixels) for these four image datasets in this paper is shown in the below Table:

Sequences-Trackers	TLD	CT	AMCMR-CT
Girl	NaN	25	**11**
Window	21	13	**6**
Sensor	NaN	127	**8**
Mast	NaN	18	**5**

The related videos and more information of these tests and UAV application can be found at CVG-UPM and ColibriProjectUAV [5] websites.

6. CONCLUSIONS AND FUTURE WORKS

Previous works in the UAV visual tracking and control have often adopted the *generative*-based method, due to

[5] http://www.youtube.com/colibriprojectUAV

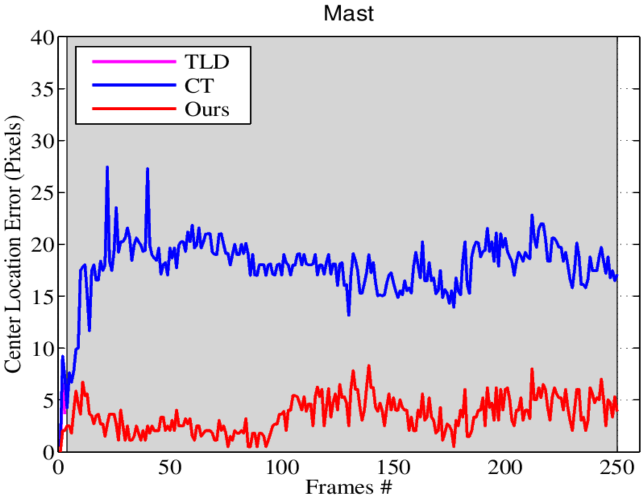

Fig. 13. The Performance of Visual Tracking for *Mast*, where, the Grey Shadow Area represents the lost measurements using TLD tracker.

the tracking targets always change their appearances, and most of the targets are in 3D shape, these kinds of methods are not widely applied in many real-time applications. While multiple positive and negative samples-based Discriminative Visual Tracking (DVT) works better in arbitrary 2D/3D target tracking.

The proposed AMCMR-CT algorithm is compared with existing state-of-art trackers (this paper presented two trackers: TLD and CT) in different kind of challenging image sequences (the total number of evaluated frames is more than 1.8×10^4) and real UAV navigation tasks, the tracking results show that this new framework is more robust, efficient and accurate under the challenging situations.

In the future works, we will provide the detailed analysis for this new framework, and add more comparisons with other existing state-of-art trackers and IMU/GPS obtained from these AMCMR-CT based UAV flight tasks.

ACKOWNLEDGMENT

The work reported in this paper is the consecution of several research stages at the Computer Vision Group-Universidad Politécnica de Madrid. This work has been sponsored by the Spanish Science and Technology Ministry under the grant CICYT DPI2010-20751-C02-01, the OMNIWORKS project (an experiment funded in the context of the ECHORD project (FP7-ICT-231143)) and the China Scholarship Council (CSC). And authors would like to thank OMNIWORKS's other two parters (Skybotix [6] and APIAXXI [7]) for valuable feedback and input.

REFERENCES

Avidan, S. (2004). Support vector tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8), 1064–1072.

Babenko, B., Yang, M.H., and Belongie, S. (2011). Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8), 1619–1632.

Bai, T. and Li, Y. (2012). Robust visual tracking with structured sparse representation appearance model. *Pattern Recognition*, 45(6), 2390–2404.

Balan, A. and Black, M. (2006). An adaptive appearance model approach for model-based articulated object tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, 758–765.

Collins, R., Liu, Y., and Leordeanu, M. (2005). Online selection of discriminative tracking features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10), 1631–1643.

Felzenszwalb, P.F., Girshick, R.B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627–1645.

Grabner, H. and Bischof, H. (2006). On-line boosting and vision. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, 260–267.

Kalal, Z., Mikolajczyk, K., and Matas, J. (2012). Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7), 1409–1422.

Martinez, C., Mondragon, I., Campoy, P., Sanchez-Lopez, J., and Olivares-Mendez, M. (2013). A hierarchical tracking strategy for vision-based applications on-board uavs. *Journal of Intelligent & Robotic Systems*, 1–23.

Mei, X. and Ling, H. (2011). Robust visual tracking and vehicle classification via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11), 2259–2272.

Mejias, L., Saripalli, S., Campoy, P., and Sukhatme, G. (2006). A visual servoing approach for tracking features in urban areas using an autonomous helicopter. In *IEEE International Conference on Robotics and Automation 2006*, 2503–2508.

Ramanan, D., Forsyth, D., and Zisserman, A. (2007). Tracking people by learning their appearance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1), 65–81.

Ross, D.A., Lim, J., Lin, R.S., and Yang, M.H. (2008). Incremental learning for robust visual tracking. *Int. J. Comput. Vision*, 77(1-3), 125–141.

Wang, Q., Chen, F., Xu, W., and hsuan Yang, M. (2011). An experimental comparison of online object tracking algorithms. In *Proceedings of SPIE: Image and Signal Processing Track*.

Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *ACM Comput. Surv.*, 38(4).

Zhang, C., Jing, Z., Tang, Y., Jin, B., and Xiao, G. (2013). Locally discriminative stable model for visual tracking with clustering and principle component analysis. *Computer Vision, IET*, 7(3).

Zhang, K., Zhang, L., and Yang, M.H. (2012). Real-time compressive tracking. In *Proceedings of the 12th European conference on Computer Vision (ECCV'12)*, 864–877.

[6] http://www.skybotix.com/

[7] http://www.apiaxxi.es/