Optical flow based detection in mixed human robot environments [⋆]

Dario Figueira, Plinio Moreno, Alexandre Bernardino, José Gaspar, and José Santos-Victor
{dfigueira, plinio, alex, jag, jasv}@isr.ist.utl.pt

Instituto Superior Técnico & Instituto de Sistemas e Robótica
1049-001 Lisboa - Portugal

Abstract. In this paper we compare several optical flow based features in order to distinguish between humans and robots in a mixed human-robot environment. In addition, we propose two modifications to the optical flow computation: (i) a way to standardize the optical flow vectors, which relates the real world motions to the image motions, and (ii) a way to improve flow robustness to noise by selecting the sampling times as a function of the spatial displacement of the target in the world.
We add temporal consistency to the flow-based features by using a temporalBoost algorithm. We compare combinations of: (i) several temporal supports, (ii) flow-based features, (iii) flow standardization, and (iv) flow sub-sampling. We implement the approach with better performance and validate it in a real outdoor setup, attaining real-time performance.

1 Introduction

Current trends in robotics research envisage the application of robots within public environments helping humans in their daily tasks. Furthermore, for security and surveillance purposes, many buildings and urban areas are being equipped with extended networks of surveillance cameras. The joint use of fixed camera networks together with robots in social environments is likely to be widespread in future applications.

The long term goal of this work, integrated in the URUS project[1], adopts this vision. The URUS project aims to achieve the interaction of robots with people in urban public areas, to improve mobility in downtown areas. A key element of the project is a monitoring and surveillance system composed by a network of fixed cameras that provide information about the human and robot activities. These multi-camera applications must also consider constraints such as real-time performance and low-resolution images due to limitations on communication bandwidth. Thus, It is fundamental to be able to detect and categorize humans and robots using low resolution and fast to compute features. We propose the use of optical flow derived features to address this problem.

Detection of humans in images is a very active research area in computer vision with important applications such as pedestrian detection , people tracking and human activity

[⋆] Research partly funded by the FCT Programa Operacional Sociedade de Informação(POSI) in the frame of QCA III, and EU Project URUS (IST-045062)

recognition . These approaches aim to model the human limbs by using features such as the silhouette [2, 3], image gradient [4], color distribution of each limb [5], optic flow [6, 7] and combinations of the features just mentioned. Detection of robots in images have become a very popular field of research in the RoboCup [1] framework, which focus on cooperative robot interaction [8–10].We address the unexplored issue of discrimination between these two classes, people and robots, which is essential to the development of algorithms that deal with *e.g.,* surveillance, in mixed human-robot environments. Our approach relies on the motion patterns extracted from optical flow, which have been used previously by Viola *et al.* [6] and Dalal *et al.* [4, 7] in order to detect pedestrian in images and videos. Viola *et al.* combine the optical flow with wavelet-based features to model the people appearance, while Dalal *et al.* compute histograms of the flow directions. Our work explores on this latter approach, comparing two types of features:

- Histogram of gradients (HOG), which computes the histogram of the optic flow orientation weighted by its magnitude.
- Motion boundary histogram (MBH), computed from the gradient of the optical flow. Similarly to HOG, this feature is obtained by the weighted histogram of the optical flow's gradient.

Using optical flow to separate robots' movement from people's movement is appealing for its independence on people and robot visual appearances (*i.e.,* color, size or shape), allowing it to model individuals with different outlooks, requiring only "different enough" patterns of movement. Since most current robots are rigid, while people tend to not be rigid at all while moving about, this a reasonable assumption to start with. Also, optical flow is not limited to high resolution images, being able to capture enough information from only a limited amount of pixels.

In order to improve the classifier's accuracy we also test the features with standardized flow described in Section 2. Using localized detections on a world reference frame, we scale the flow to its corresponding real-world metric value, creating invariance to the target's distance to the camera. In addition, we don't use consecutive images to compute flow which we refer to as spatial sub-sampled flow. We store a frame, wait for the target to move in the real world, and only after its displacement is larger than a threshold, we compute the optical flow from the stored frame to the present image.

The histogram-based features provide the data samples for the learning algorithm, GentleBoost [11]. This algorithm is a very efficient and robust classifier that adds the response of several base (weak) classifiers. In addition, we consider the temporalGentleBoost [12], a recent modification of GentleBoost that exploits the temporally local similarities of the features.

In the next section we will describe the features employed to represent the targets. Section 3 describes the learning algorithm used to learn how to distinguish people from robots. We then present some results on a real live setting and finish with the conclusions.

[1] http://www.robocup.org/

2 Target Representation

In this work we assume that detection, tracking and localization (in the ground plane) of targets in the camera's field-of-view (FOV) is already done by any of the existing algorithms available in the literature. For instance we use background subtraction [13] for detection, nearest-neighbor for tracking and homographies for localization. Our goal in this paper is to discriminate among different classes of targets using motion cues.

For (dense) optical flow computation, we use the implementation of [14][2], an algorithm that introduces a new metric for intensity matching, based on the unequal matching (i.e. unequal number of pixels in the two images can be correspondent to each other). We chose this algorithm for its good balance between computational load and robustness to noise [14].

2.1 Flow Standardization

The optical flow vectors encode the pixel displacement between two images, independent of the corresponding real displacement. This means that an object closer to the camera will have a large pixel displacement, while the same object will have small flow vectors when moving far away. Thus, it is very difficult to match similar motions by using the features computed directly from the optical flow. In order to overcome this limitation we standardize the flow using the world coordinate locations of the moving objects in the scene.

Given the displacement of each object in the world in metric coordinates, and in the image in pixels, we derive a linear scale factor to relate the optical flow, in the image, to the motion in the world. We illustrate this in Figure 1, where the gray arrows display the optical flow in two different detections, the blue arrows represents the movement in the world and the red arrows encode the mean displacement of the detected bounding boxes in the image. The flow magnitude (f pixels), is larger when the object is close, and smaller when farther away. The average displacement (P pixels) follows the same behavior, while both world displacements (M meters) keep the same value. Therefore the optical flows can be scaled to a similar value ($f.M/P$ meters).

The flow standardization just described assumes that the motions of the target's limbs are aligned to the mean displacement of the target. If this assumption is violated, the motions with other directions are projected to the direction aligned with the mean displacement's vector. Since in general, while a person is moving, their limb motions will be parallel to the motion of her body, the assumption will hold for most of the sequences.

The flow standardization described above causes the flow magnitude to be independent to the target's distance to the camera, but still dependent on the target's velocity. If an individual moves faster in some frames and slower in other frames, its displacement will be different for the respective pairs of frames, so the features extracted will be dissimilar.

We implement spatial sub-sampling of the optical flow in order to provide velocity independence to the features. This comprises, for a given target, the selection of the

[2] http://www.cs.umd.edu/˜ogale/download/code.html

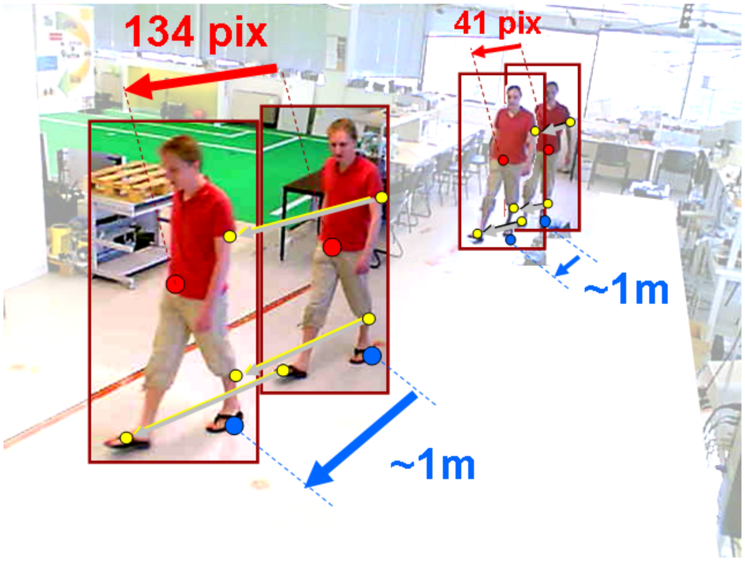

Fig. 1. Smaller grey arrows: Optical flow; Big red arrow: mean pixel displacement in the image; Big blue arrow: meter displacement in the world.

frames to compute the optical flow based on its displacement. The method includes these steps: (i) store a frame; (ii) wait for the target to move more than a threshold distance; (iii) compute the optical flow using the stored frame and the present frame. In addition, the spatial sub-sampling provides invariance to changes on the sampling frequency of the cameras.

2.2 Flow-based features

We compare two kinds of features: motion boundary histogram (MBH) [7] and histogram of gradients (HOG) [4], considering four kinds of flow data: (i) raw flow, (ii) spatially sub-sampled, (iii) standardized flow and (iv) spatially sub-sampled and standardized flow.

MBH captures the local orientations of motion edges. We do it by considering the two flow components (x and y) as independent images, and taking their gradients. To extract the spatial information of the gradient image, two types of sampling are considered: dividing the image in cartesian or polar regions. HOG is computed in a similar way but directly on the flow vectors, and we also consider the same two sampling types: cartesian and polar. In total, we compare among sixteen different combinations of features, samplings and flow data. In difference to the original MBH and HOG features, that overlap sampling regions, we don't consider overlapping.

3 Learning algorithm

The Boosting algorithm provides a framework to sequentially fit additive models in order to build a final strong classifier, $H(x_i)$. This is done minimizing, at each round, the weighted squared error, $J = \sum_{i=1}^{N} w_i (y_i - h_m(x_i))^2$, where $w_i = e^{-y_i h_m(x_i)}$ are the weights, N the number of training samples, x_i is a feature and y_i is the correspondent class label. At each round, the weak classifier with lowest error is added to the strong classifier and the data weights adapted, increasing the weight of the misclassified samples and decreasing correctly classified ones [11]. Then, in the subsequent rounds the weak classifier focus on the misclassified samples of the previous round.

In the case of GentleBoost it is common to use simple functions such as regression stumps. They have the form $h_m(x_i) = a\delta\left[x_i^f > \theta\right] + b\delta\left[x_i^f \leq \theta\right]$, where f is the number of the feature and δ is an indicator function (i.e. δ[*condition*] is one if *condition* is *true* and zero otherwise). Regression stumps can be viewed as decision trees with only one node, where the indicator function sharply chooses branch a or b depending on threshold θ and feature x_i^f. To optimize the stump one must find the set of parameters $\{a, b, f, \theta\}$ that minimizes J w.r.t. h_m. The optimal a and b are obtained by closed form and the value of pair $\{f, \theta\}$ is found using an exhaustive search [15].

A recent approach considers the temporal evolution of the features in the boosting algorithm, improving the noise robustness and performance. Ribeiro et al. [12] model temporal consistency of the features, by parameterizing time in the weak classifiers. The Temporal Stumps compute the mean classification output of the regression stump, in a temporal window of size T,

$$h_m(x_i) = a\left(\frac{1}{T}\sum_{t=0}^{T-1}\delta\left[x_{i-t}^f > \theta\right]\right) + b\left(\frac{1}{T}\sum_{t=0}^{T-1}\delta\left[x_{i-t}^f \leq \theta\right]\right). \qquad (1)$$

The temporal weak classifier of Eq. 1 can be viewed as the classic regression stump with a different "indicator function". If $T = 1$ it becomes the original regression stump, and for $T > 1$ the indicator function changes. The new indicator functions

$$\varDelta_{+}^{T}(f, \theta, T) = \frac{1}{T}\sum_{t}^{T-1}\delta\left[x_{i-t}^f > \theta\right], \quad \varDelta_{-}^{T}(f, \theta, T) = \frac{1}{T}\sum_{t}^{T-1}\delta\left[x_{i-t}^f \leq \theta\right], \qquad (2)$$

compute the percentage of points above and below the threshold θ, in the temporal window T and for the feature number f. The indicator functions with temporal consistency in Eq. 2, can take any value in the interval $[0\ 1]$, depending on the length of the temporal window used. For example, if $T = 2$ the functions can take 3 different values, $\varDelta_{+}^{T} \in \{0,\ 1/2,\ 1\}$, if $T = 3$ can take four values, $\varDelta_{+}^{T} \in \{0,\ 1/3,\ 2/3,\ 1\}$ and so on. The fuzzy output of the new "indicator function", $\varDelta$, represents the confidence of threshold choice to use the data with temporal support T. Thus, at each boosting round, we use a weighted confidence of both branches, instead of choosing only one branch.

Using the weak classifier with temporal consistency of Eq. 1 in the cost function, Ribeiro et al. [12] obtain closed expressions for the parameters a and b that minimize the error J. The optimal f, θ and T are obtained by exhaustive search. The learning algorithm shown in figure 2 is similar to GentleBoost, but optimizing the temporal stump of Eq. (1).

```
1. Given: $(x_1, y_1), \ldots, (x_N, y_N)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$, set $H(x_i) := 0$,
   initialize the observation weights $w_i = 1/N$, $i = 1, 2, \ldots, N$
2. Repeat for $m = 1, \ldots, M$
   (a) Find the optimal weak classifier $h_m^*$ over $(x_i, y_i, w_i)$.
   (b) Update strong classifier $H(x_i) := H(x_i) + h_m^*(x_i)$
   (c) Update weights for examples $i = 1, 2, \ldots, N$, $w_i := w_i e^{-y_i h_m^*(x_i)}$
```

Fig. 2. Temporal Gentleboost algorithm.

4 Results

We compute the 16 different combinations of flow-based features (Section 2.2) in three scenarios: people walking, people loitering and robot moving. The motion patterns of people walking and robot moving will be properly extracted by optical flow-based features, so they are the nominal classification scenario. People loitering on the other hand, is a difficult situation as it provides small optical flow values. Both people walking and loitering are very common activities, therefore we decide to focus on them in this work. Figure 3 shows the setup of each scenario, which includes video sequences from 10 cameras.

We grabbed five groups of sequences, where each one includes images from 10 cameras. One group with a person walking, another group with a different person walking, two groups with the same pioneer robot moving in two different conditions, and the last group with a third person loitering. The people class videos have a total of 9500 samples of the optical flow and the robot class videos have a total of 4100 samples. The segmentation and tracking of the moving objects in the scene are provided by: -LOTS background subtraction for detection [13] and nearest neighbor for tracking. The LOTS algorithm provides the bounding boxes of the regions of interest and its respective segmented pixels. Nearest neighbor is computed between the center points of the two bounding boxes.

We follow a cross validation approach to compare the classification result of the temporal GentleBoost algorithm. We build two different groups of training and testing sets. The people loitering data is always in the testing set, each person belongs to the training set for one of the experiments, and each pioneer robot sequence belongs to the training set once. The Tables 4, 2 and 3 show the average of the recognition rate for each frame using the two experiments. Each table summarizes the results for a fixed value of temporal support, T, and we notice the large performance improvement brought by the temporal support of the flow-based features when compared to the common GentleBoost ($T = 1$).

We observe three general patterns from the recognition rate:

- The polar sampling of the images performs better than the cartesian counterpart. It seems that the polar sampling is better suited for modeling the motion of the peoples' limbs, so it is easier to discriminate between people and robots.
- The Motion Boundary Histogram (MBH) feature has better performance than the optical flow histogram. The MBH has a richer representation based on two images

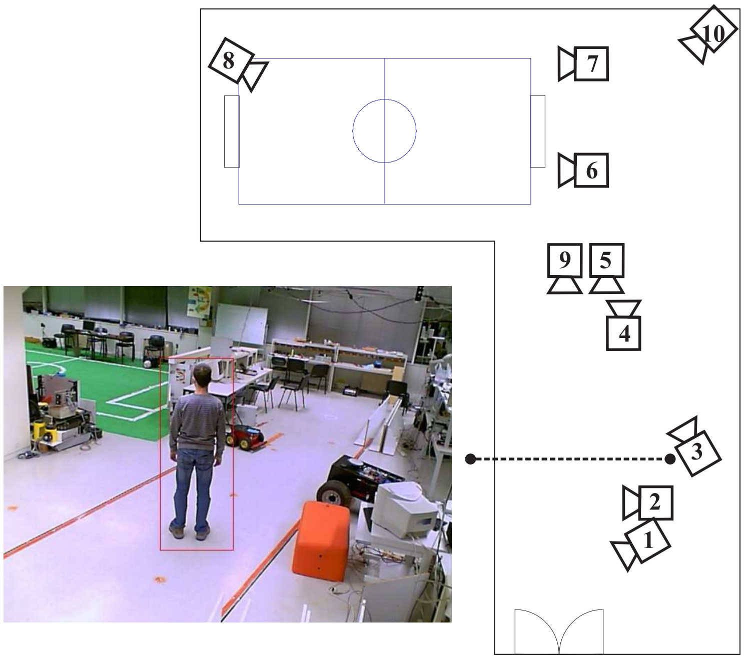

Fig. 3. Experimental setup for training scenario

Feature	sub-sampled+standardized	standardized	sub-sampled	raw flow
polar flow histogram	76.15	**92.26**	75.96	90.62
cartesian flow histogram	71.90	**87.90**	71.63	87.20
MBH cartesian	90.60	83.13	**91.39**	82.73
MBH polar	93.14	90.60	**93.67**	89.40

Table 1. Recognition rate of several features, using a maximum temporal support $T = 5$ frames of the temporal boost algorithm

Feature	sub-sampled+standardized	standardized	sub-sampled	raw flow
polar flow histogram	76.71	**87.23**	76.82	85.68
cartesian flow histogram	78.71	**85.18**	78.74	84.13
MBH cartesian	79.79	75.48	**79.98**	74.65
MBH polar	88.80	83.87	**88.94**	81.30

Table 2. Recognition rate of several features, without temporal support ($T = 1$ frames) of the temporal boost algorithm

Feature	sub-sampled+standardized	standardized	sub-sampled	raw flow
polar flow histogram	77.23	**95.43**	77.77	93.02
cartesian flow histogram	73.64	**89.32**	74.22	88.52
MBH cartesian	91.68	87.59	**92.25**	85.25
MBH polar	**94.58**	91.41	**94.58**	91.01

Table 3. Recognition rate of several features, using a maximum temporal support $T = 10$ frames of the temporal boost algorithm

that extract the first order spatial derivatives of the optical flow, while the flow histogram is a more efficient representation based on only one image, the optical flow.

– The spatial sub-sampling of the optic flow computation has a positive effect on the MBH features, while has a negative impact on the flow-based histogram features. On one hand, it seems that the MBH feature needs optical flow measurements with low levels of noise, which is provided by the spatial sub-sampling for computing the optical flow. On the other hand, the evolution in time of the optical flow histogram is better captured by the computation of the optical flow between consecutive images.
– The standardization of the optical flow has a very small improvement of the recognition rate, because all the features compute normalized histograms that provide a sort of standardization of the features.

From Table 3 we see that the best compromise between accuracy and robustness is provided by the MBH polar using the spatial sub-sampling. Thus, we implemented this feature in a C++ program that distinguishes between people and robots in real-time.

5 Conclusions

In this work we compared several optical flow based features to distinguish people from robots. We propose a way to standardize the optical flow vectors, scaling them to their corresponding metric value in the real-world, and also a more efficient and robust way of computing the optical flow that subsamples the images on time using the spatial displacement of the targets in the world. We used Temporal GentleBoost algorithm for learning, which is able to improve the classification rate by considering previous features' values, thus including a temporal support of the features. We test for several combinations of temporal supports, type of feature and flow standardization in order to verify the combination with better performance and robustness. The application of spatial sub-sampling to the optical flow reduces the computational load of the algorithm while keeping similar results to its counterparts. These computational savings guarantees real-time classification. The Motion Boundary Histogram feature with world spatial sub-sampling of the optical flow and temporal support of 10 frames have a very good trade-off between accuracy and robustness. We implement the combination just mentioned, validating it in a outdoors setting that shows the generalization capabilities of the proper combination of features, classifier and sampling approaches, providing a very good performance.

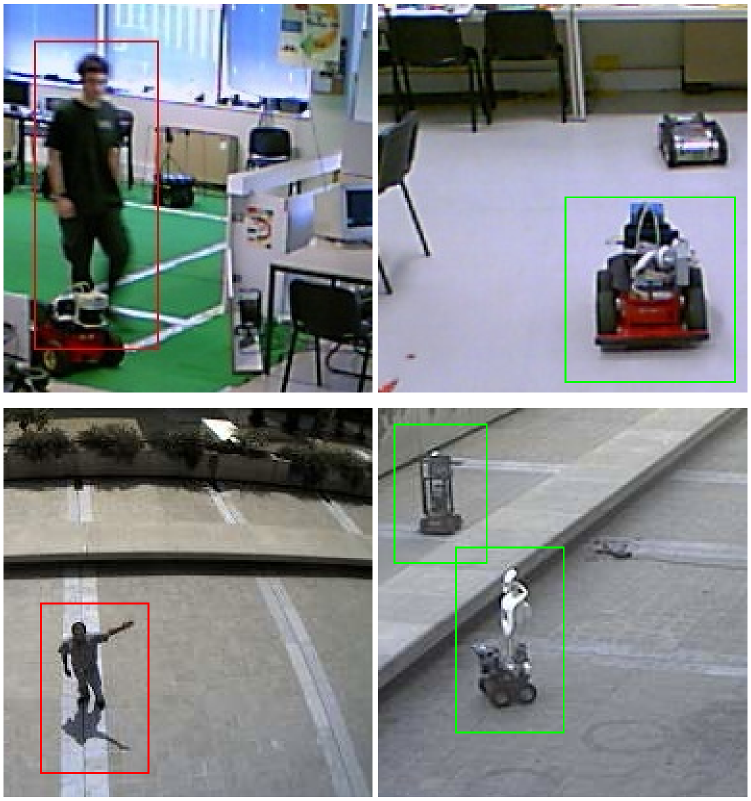

Fig. 4. Examples of person and robot training samples (on top) and real-time classification in an outdoors setting (bottom).

References

1. Sanfeliu, A., Andrade-Cetto, J.: Ubiquitous networking robotics in urban settings. In: Workshop on Network Robot Systems. Toward Intelligent Robotic Systems Integrated with Environments. Proceedings of 2006 IEEE/RSJ International Conference on Intelligence Robots and Systems (IROS2006). (2006)
2. Diaz De Leon, R., Sucar, L.: Human silhouette recognition with fourier descriptors. In: Pattern Recognition, 2000. Proceedings. 15th International Conference on. Volume 3. (2000) 709–712 vol.3
3. Wang, L., Tan, T., Ning, H., Hu, W.: Silhouette analysis-based gait recognition for human identification. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(12) (2003) 1505–1518
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1, Washington, DC, USA, IEEE Computer Society (2005) 886–893
5. Ramanan, D., Forsyth, D.A., Zisserman, A.: Tracking people by learning their appearance. Pattern Analysis and Machine Intelligence, IEEE Transactions on **29**(1) (2007) 65–81

6. Viola, P., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision **63**(2) (2005) 153–161
7. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: European Conference on Computer Vision. (2006)
8. Kaufmann, U., Mayer, G., Kraetzschmar, G., Palm, G.: Visual robot detection in robocup using neural networks. In: RoboCup 2004: Robot Soccer World Cup VIII. (2005) 262–273
9. Mayer, G., Kaufmann, U., Kraetzschmar, G., Palm, G.: Biomimetic Neural Learning for Intelligent Robots. Springer Berlin / Heidelberg (2005)
10. Lange, S., Riedmiller, M.: Appearance-based robot discrimination using eigenimages. In: RoboCup. (2006) 499–506
11. J. Friedman, T.H., Tibshirani, R.: Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). Annals of Statistics **28**(2) (2000) 337407
12. Ribeiro, P.C., Moreno, P., Santos-Victor, J.: Boosting with temporal consistent learners: An application to human activity recognition. In: Proc. of 3rd International Symposium on Visual Computing. (2007) 464–475
13. Boult, T.E., Micheals, R.J., Gao, X., Eckmann, M.: Into the woods: Visual surveillance of noncooperative and camouflaged targets in complex outdoor settings. Proceedings Of The IEEE **89**(10) (2001) 1382–1402
14. Ogale, A.S., Aloimonos, Y.: A roadmap to the integration of early visual modules. International Journal of Computer Vision **72**(1) (2007) 9–25
15. Torralba, A., Murphy, K., Freeman, W.: Sharing visual features for multiclass and multiview object detection. IEEE Transactions On Pattern Analysis and Machine Intelligence **29**(5) (2007) 854–869