Model generation for robust object tracking based on temporally stable regions

Prithviraj Banerjee
Indian Institute of Technology, Kharagpur
banerjee.prithviraj@gmail.com

Axel Pinz
Graz University of Technology, Austria
http://www.emt.tugraz.at/~pinz

Somnath Sengupta
Indian Institute of Technology, Kharagpur
ssg@ece.iitkgp.ernet.in

Abstract

Tracking and recognition of objects in video sequences suffer from difficulties in learning appropriate object models. Often a high degree of supervision is required, including manual annotation of many training images. We aim at unsupervised learning of object models and present a novel way to build models based on motion information extracted from video sequences. We require a coarse delineation of moving objects and subsequent segmentation of these motion areas into regions as preprocessing steps and analyze the resulting regions with respect to their stable detection over many frames. These 'temporally stable regions' are then used to build graphs of reliably detected object parts which form our model. Our approach combines the feature-based analysis of feature vectors for each region with the structural analysis of the graphical object models. Our experiments demonstrate the capabilities of this novel method to build object models for people and to robustly track them, but the method is in general applicable to learn object models for any object category, provided that the object moves and is observed by a stationary camera.

1. Introduction

Recent success in object category recognition has led to high expectations in applying these methods to image retrieval and to recognition of objects in still images. While tracking and recognition of objects in video sequences is of utmost importance in many applications, we observe a number of drawbacks in applying existing object categorisation methods directly to video processing. Many categorisation techniques require a high degree of supervision in the training phase, including manual delineation and annotation of objects or their bounding boxes (see e.g. [11] for a survey of categorisation methods and http://www.pascal-network.org/challenges/VOC/ for a description of the databases of the PASCAL object recognition challenge). We try to overcome these drawbacks by proposing a novel, completely unsupervised method to generate object models directly from video sequences. As a first step, in this paper, we demonstrate how such models can be generated for individual, specific people.

We assume a stationary camera observing an arbitrary scene with moving objects and require a number of preprocessing steps including a moderately good motion segmentation (i.e. a coarse delineation of areas in the image where motion occurs), and colour segmentation of these areas into homogeneous regions using Mean Shift [1]. The Mean Shift segmentation requires the user to enter three parameters: (a) Spatial Bandwidth (BW_R), (b) Colour Bandwidth (BW_C) and (c) Minimum Region Size. BW_R defines the spatial extent to which the pixels can be defined as belonging to the same region. BW_C defines the level of contrast required for two colours to be distinct. Minimum Region Size sets the lower bound limit on the size of a segmented region, to avoid fragmentation. BW_R and BW_C compete with each other during segmentation and an appropriate value needs to be set depending on the content of the video frame. These parameters are determined by the user at system initialisation. The user aims at obtaining a consistent segmentation throughout the video sequence, and should not overemphasise on a detailed segmentation. Examples of acceptable Mean Shift segmentation in the context of our work are shown in Figure 1 (parameter settings: BW_R=8, BW_C=10 with a frame size of 640×480 pixels and Minimum Region Size = 300 pixels).

This paper is focused on the subsequent processing steps:

1. Feature Extraction: The segmented regions are analysed and the relevant features are extracted (Section 2.1).

2. Custering: An Agglomerative Clustering algorithm is

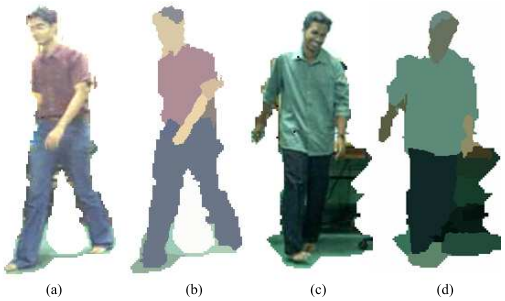

Figure 1. (a) and (c) are original motion segmented images. (b) and (d) are the respective Mean Shift segmented images

applied to cluster the features and identify the *Temporally Stable Regions (TSRs)* in a sequence (Section 2.2).

3. Model Generation: Adjacency relations are established between the TSRs and template models are generated (Section 3).

4. Model Matching: The final step consists of matching models from one set of sequence to another. Individual models of people can be searched out in group models of people (Section 4).

Finally, we present experimental results on video sequences of people in Section 5 which demonstrate the benefits and limitations of our novel method.

1.1. Related Work

Currently, a lot of effort is being spent on developing autonomous video surveillance systems, which are capable of detecting and tracking people, and subsequently performing activity analysis on their motion. Pfinder [13] is one of the earliest systems of this kind which has been applied in practice. It combines blob and contour analysis and can also use prior knowledge. Haritaoglu et al proposed the W^4 [3] video surveillance system for detection and tracking of people. W^4 classifies objects as people, vehicles or others on the basis of static size and shape properties, and dynamic analysis of shape periodicity. It is incapable of tracking only individual people. the Hydra [4] system was introduced as a preprocessing step to W^4. the Hydra system is capable of detecting a group of people in a scene, and build appearance models of the individuals in the group to keep track of them as they change positions. Isard and MacCormick presented the BraMBLe system [6] which uses Bayesian filter for tracking multiple objects in a video scene, when the number of objects present is unknown and varies over time. It also uses particle filter to perform joint inferencing on both the number of objects and their configurations. A radically different approach is applied by Kang et al in [7].

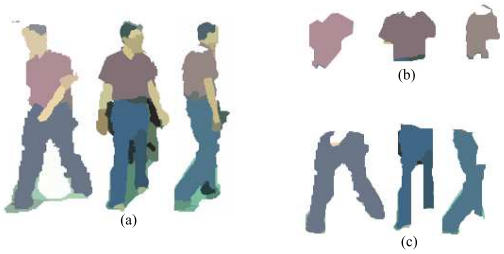

Figure 2. (a)Mean Shift segmented frames. (b) Shirt regions clustered together to form a TSR. (c) Trouser region clustered together to form a TSR

They construct effective templates of people's shapes using Self Organising Maps, and model the dynamic transition of people's shape using Hidden Markov Models. Generic object tracking systems like Comaniciu et al [2] and McKenna et al [9] have been shown to be applicable in tracking people in video sequences.

This paper is inspired by the work of Peter Tu et al [12] on Colour Annotated Adjacency Graphs for Object Recognition. Their aim was to identify objects in cluttered images using sub-graph matching. The objects are represented in a template and images are searched using weighted adjacency graphs. This paper presents a system based on similar principles of object tracking using graph based modelling.

2. Feature Extraction and Clustering

This section describes the Clustering and Feature Extraction algorithm applied for extracting the Temporally Stable Regions (TSR) from the video sequence. Temporally Stable Regions are defined as those regions which can be successfully detected over the majority of the frames in a video sequence. For example in Figure 2, consider the tracking of a person wearing a red shirt and blue trousers. After the Mean Shift segmentation, shirt and trousers will appear as a reddish and a bluish blob, respectively. If these blobs can be detected over a number of frames, then the corresponding regions are marked as Temporally Stable Regions. On the other hand, the hair region of the person is not giving a consistent blob through the video sequence.

2.1. Feature Extraction from Segmented Regions

From each Mean Shift segmented region, a vector of 22 features is computed which are as follows:

1. Colour in LUV space: The colour of the region is represented in the HSV colour space and the mean colour (3 Features) of each region is computed along with its co-variance matrix (9 features) of the three colour channels.

2. Hu Moments: The seven Hu moments [5] are computed to give a statistical description of the region which is invariant to position, size and orientation.

3. Region area: The area of a region is the number of pixels contained in the region normalised to the area of the frame. It is useful in distinguishing between similarly coloured regions, and brings in the size of the region as a distinguishing criterion for clustering.

4. Centre of gravity: The X and Y coordinates of the centre of gravity of the region are included as features. It is assumed that the objects being tracked in the images are not moving very fast with respect to the frame rate. This feature ensures that otherwise similar regions from the same frame are not grouped together.

2.2. Clustering of Region Features

Agglomerative clustering is applied on the features. A region being tracked over multiple frames will have similar feature vectors throughout the sequence, and will therefore form individual groups upon clustering. Noisy regions which keep appearing and disappearing during the video sequence do not form strong clusters and can be separated out by this method. Agglomerative clustering allows custom defined distance matrices to be used, but requires that the maximum number of clusters possible be defined before hand. The maximum number of clusters is determined by the sum of the mean and variance of the number of regions detected in each frame.

The colour features from each feature vector are evaluated on the basis of Bhattacharya distance. Bhattacharya Distance is known to successfully discriminate two colours compared to the other distance measures like Euclidean distance etc. The Bhattacharya distance for feature vector $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$D_C^2(\mathbf{x},\mathbf{y}) = \frac{1}{2}(\mathbf{x}\Sigma_x^{-1}\mathbf{x}^t + \mathbf{y}\Sigma_y^{-1}\mathbf{y}^t - 2\mathbf{z}\Sigma_z^{-1}\mathbf{z}^t) \quad (1)$$

with the additional definitions of matrix $\Sigma_z = \frac{1}{2}(\Sigma_x^{-1} + \Sigma_y^{-1})$ and vector $\mathbf{z} = \frac{1}{2}(\Sigma_x^{-1}\mathbf{x} + \Sigma_y^{-1}\mathbf{y})$. The Hu Moments, Area and Location features are compared using Euclidean Distance to give $D_{Hu}(\mathbf{x},\mathbf{y})$, $D_A(\mathbf{x},\mathbf{y})$and $D_{Loc}(\mathbf{x},\mathbf{y})$. All the distances are added up together to form the distance matrix between all the regions, i.e. $D_{Net}(\mathbf{x},\mathbf{y}) = D_C + D_{Hu} + D_A + D_{Loc}$. The Distance matrix $D_{Net}(\mathbf{x},\mathbf{y})$ is used to cluster the regions using the Average Linkage algorithm [8]. Average Linkage computes the average distance between all pairs of objects in cluster r and cluster s. The distance function used is

$$D(r,s) = \frac{1}{n_r n_s}\sum_{i=1}^{n_r}\sum_{j=1}^{n_s} dist(\mathbf{x_{ri}},\mathbf{y_{sj}}) \quad (2)$$

where $dist(\mathbf{.})$ is the Euclidean distance function. Clustering generates two types of groups of regions :

1. Proper clusters: These are clusters of regions which have been detected consistently over a sequence of frames, and hence are temporally stable. The proper clusters should have only one region from each frame. A single proper cluster forms a Temporally Stable Region (TSR).

2. Improper clusters: These clusters contain regions which are not very similar to each other. The intra-class variance is quite high in such clusters. They may also contain multiple regions from the same frame, and could be concentrated to only a few number of frames in the video sequence. Improper clusters are detrimental to the formation of stable object models for tracking, and hence are considered unwanted.

A systematic trimming process is employed to separate the proper clusters from the improper clusters. Firstly the clusters are divided into two groups based on the frequency of detection over a fixed number of frames. The threshold is decided by Otsu's method [10], which aims at minimising the intra-class variance. The threshold is bounded maximally at 50 percent, i.e. all regions appearing in 50 percent of the images are allowed to stay. The clusters with higher frequency value are retained for further processing whereas the rest are discarded. The intra-cluster variance of the retained clusters are computed (the location feature variance is ignored as any moving region will have a naturally high location variance). Clusters with high intra-class variance are rejected. Finally the remaining clusters are trimmed further to reduce the location-feature variance. This ensures that each cluster has only one or no region contributing per frame. The surviving clusters are the proper clusters detected over the sampled sequence of frames.

Each of the proper clusters represents a Temporally Stable Region in the sampled sequence, as such regions have been detected successfully over the majority of the frames in the sampled sequence with low variation in their colour, size, Hu moments and location in the frame.

3. Model Generation from TSR

A list of neighbouring TSRs is constructed from the sampled sequence, which can be represented as a graph. The nodes in the model graphs represent the TSRs detected over the sampled set of frames. For example, in Figure 3 (b),(c),(d) and (e), we can clearly see the detected TSR nodes over a sequence of frames. All the TSR nodes are grouped together to form a model graph of the object (Figure 3(a)). Edges are placed between neighbouring nodes. Two nodes are defined to be neighbouring, if the regions

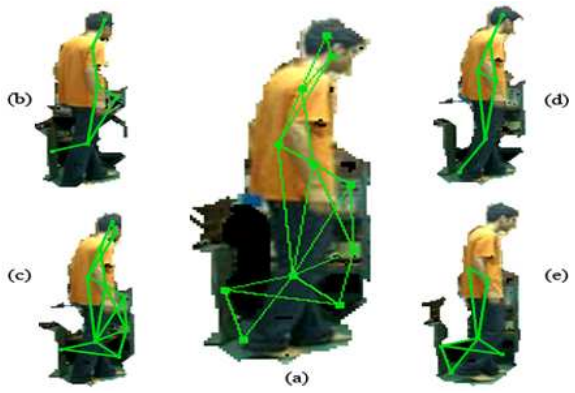

Figure 3. (a)Net model generated of human figure over a range of frames. (b),(c)(d)and(e) are the individual models detected for each frame, which combine to give the net model in (a).

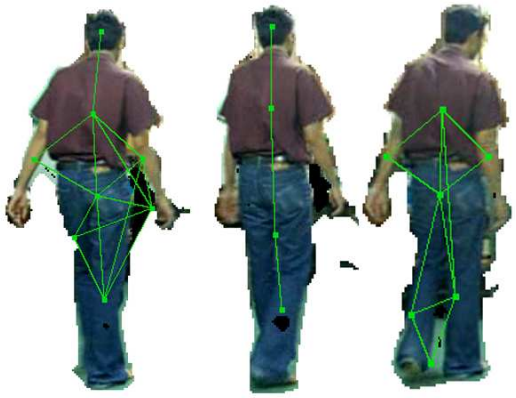

Figure 4. Examples of models generated over a larger period of time. Note that all the models have TSR nodes which are common over amongst all the models

they are representing are near to each other, within a certain limit of pixels. Each node has a corresponding 20 dimensional feature vector, which describes the colour (12 features), Hu moments (7 features) and area (1 feature) of the TSR corresponding to that node. This representative feature vector of the TSR is the average feature vector of all the regions clustered into that particular TSR. Location as a feature is ignored in graph based matching, as it varies greatly for larger lengths of frame period. The edges do not have any feature vector associated with them. They define the spatial arrangement and connectivity between the nodes, which subsequently helps in the structural analysis of the graphical object models. Model graphs are generated for successive sets of frames as shown in Figure 4, which are subsequently used for comparison with the models in the database. This enables the tracking and identification of the object underlying the Model Graph.

4. Model Matching

Tracking a person in the video implies the ability to identify the same person in two different sections of the video and assigning it a unique id throughout the video sequence. To label the people consistently, the models in the present frame have to be matched against the models detected in previous frames. The models from the previous frame sets are saved in a database. We need to find the best match for the nodes given in the present frame graph model to the nodes given in the database graph model. This is not a standard graph isomorphism problem, as the matching is dominantly based on the corresponding feature vectors of the nodes, and not on the node-edge structure of the graph. The structure plays a role in a more subtle manner, when we consider edge based voting for graph matching in section 4.2.

4.1. Feature based Graph Matching

The 'Feature based Graph Matching' tries to find the best match for a model, by comparing all the feature nodes in its graph to every other feature node present in the database. The database model obtaining the highest number of matches with the nodes of the present frame model is said to match with it.

There are some disadvantages with such a matching method. Firstly, certain nodes have been found to have common characteristics across all models. The hair region or the skin region of the face is an example of common TSRs detected for all human graph models. Hence such a node will be matched to all the human models in the database, generating false votes and erroneous matching. Secondly, one to one comparison of nodes leads to a lot of ambiguity, as certain TSRs cannot be distinguished just on the basis of colour, Hu moments and size. For example, the head region is commonly mistaken for the black coloured shoes, and the skin region of the face looks very similar to the hand region. This poses a difficulty in correctly matching the nodes, even though they are matched into the correct database model.

Such deficiencies can be overcome by a structural analysis of the graph, where the edge linkages are compared to match similar nodes.

4.2. Structural Analysis for Node Matching

It was noticed that two correctly matching nodes will also have a similar neighbourhood of nodes. For example, the boots of a man will have the trouser region as a node neighbouring to it, whereas the head region will be neighbouring to the shirt region of the person. Hence, the number of matching neighbours detected for each node will be a better matching criterion.

Let the model graphs detected in the current set of frames be numbered $P_1, \ldots, P_N$ where N are the total number of

model graphs present in the current frame set. Similarly the model graphs present in the Database are numbered as $D_1, \ldots, D_M$ where M is the total number of models present in the database. The a^{th} node in graph P_i is denoted by $n_a^{P_i}$. The edge between two nodes $n_a^{P_i}$ and $n_b^{P_i}$ is denoted by $e_{ab}^{P_i}$. Similarly, the c^{th} node in graph D_i is denoted by $n_c^{D_i}$. The edge between two nodes $n_c^{D_i}$ and $n_d^{D_i}$ is denoted by $e_{cd}^{D_i}$. The steps of the structure based matching algorithm are presented as follows:

- A one to one match is carried out between each node in model P_i, and every other node in the database models. We find the best matched node $n_{best}^{D_j}$ for each of the model in the database.
- We define a neighbour matching function $NgbMatch(n_a^{P_j}, n_{bestmatch}^{D_j})$. It computes the number of matches between the neighbours of node $n_a^{P_i}$ in P_i and the neighbours of node $n_{best}^{D_j}$ in D_j. The matching is done by taking the Euclidean distance between the representative feature vector of the TSRs corresponding to the nodes (excluding the location feature).
- For every node $n_a^{P_i} (1 \leq a \leq N)$, we compute $NgbMatch$ function and make a list of the best matched models in the database for the nodes in P_i. The list is generated as

$$List(a) = \underset{j=1\ldots M}{argmax}[NgbMatch(n_a^{P_j}, n_{best}^{D_j})] \quad (3)$$

- The model P_i gets assigned to the k^{th} model in the database where

$$k = \underset{j}{argmax}\, Histogram(List) \quad (4)$$

- After the correct database model has been identified, the nodes and edges in the database model are updated from the model in the current frame set. If a node has no match present in the database model, it is added as a new node to it, and its corresponding edges are updated.

In many situations there are ambiguous matches between nodes and their neighbours across database models. It sometimes becomes impossible to fully resolve the ambiguity, and hence a hard decision is taken to select the node with the least feature distance, although it might not be the correct choice. A most common example is the confusion between the left and the right shoe, or between a shoe and its shadow. Such pairs of objects are having a similar colour, Hu moments and area characteristics, and also share the same set of neighbours. In such cases, we resolve ambiguity by choosing the match with least distance. Such a decision might result in updating the wrong node in the database model. This is not a cause of major concern, as they are very similar in their appearance, and hence it does not matter if the shoe node gets updated with the features of the shadow underneath it. In the long run, such wrong matches do not cause instability to the database model.

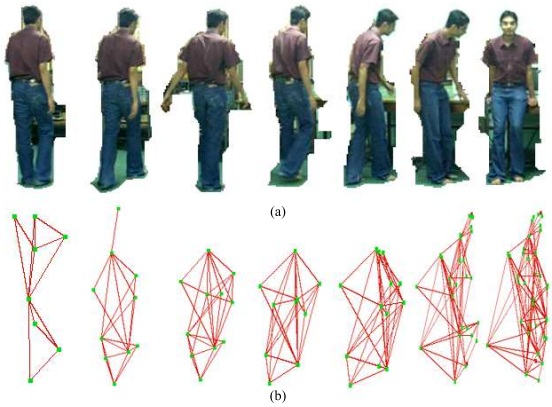

Figure 5. (a) Original frame sequence (b) Corresponding model in the database getting updated with time.

5. Experimental Results

The model generation and tracking system is first tested on simple one man videos. This gives a good insight into how the models are detected and updated into the database. Figure 5(a) shows some sample sequence frames from the video, and Figure 5(b) shows the corresponding model graphs being updated in the database.

Figure 6 shows a video sequence containing two people, who get completely occluded and then eventually reappear. The tracking system is able to successfully track both the people, and assigns a consistent unique identification number to both objects. The assignment of unique identification number is significant as it shows that our system can recognise two separate persons and track them as two separate objects even after returning from complete occlusion.

Figure 7 shows a group of people standing together. The motion segmentation module returns a single motion region in such a scenario, and hence it is difficult to judge how many people are present and to identify them based on previous actions in the video. The sequence was analysed for TSRs and the resulting model graph generated from those frames is overlayed on Figure 7. Let us assume that database object models of the individual people have been learnt from previous sections of the video (Figure 5). Clearly a subgraph matching algorithm should be able to match the database object models to the combined graph

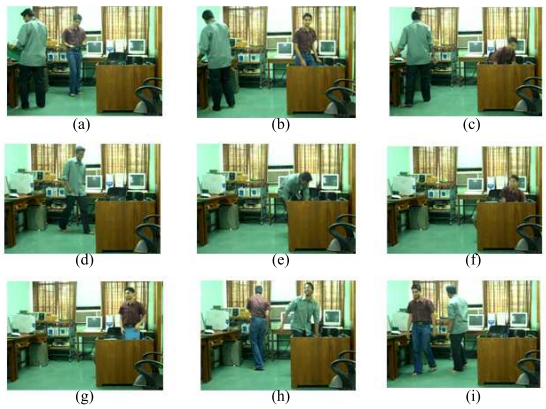

Figure 6. Sample sequence of frames consisting of two people getting temporarily occluded. The system is consistently able to track the people with unique ids after they reappear.

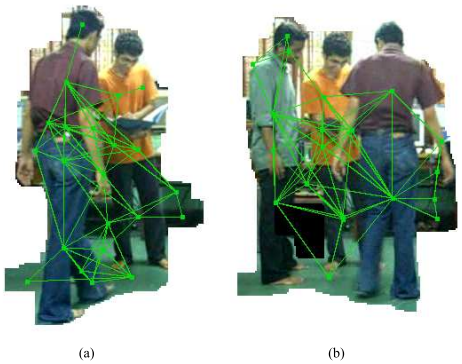

Figure 7. Models extracted from a group of people can be used to identify individual people using sub-graph matching.

structure detected for a group of people. This would greatly simplify detecting and identifying individuals in a group of people, even if they are partially occluded. Subgraph matching is obviously the next best step for this system to develop towards.

6. Conclusions

In this paper, we presented a novel method for tracking a moving object through a video sequence. It employed characterising an object by clustered regions (Temporally Stable Regions) over a collection of frames. Graphical model descriptors were generated from these Temporally Stable Regions to uniquely identify each object. A structure based graph matching algorithm was described to match the graphical model descriptors between successive sets of frames. Tests were carried out on test videos, and initial experimental results seem promising.

Our future work will focus on the application of this novel method to learn object *category* models in a completely unsupervised manner, just from video sequences of moving objects. To achieve this goal, we will have to learn which (category-specific) features still can be used to discriminate between categories (e.g. skin colour might be useful but colour of clothes will not).

References

[1] D. Comaniciu and P. Meer. Mean shift: A robust approach towards feature space analysis. In *IEEE Trans. PAMI*, volume 24(5), pages 603–619, 2002. 1

[2] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. PAMI*, 25(5):564–575, 2003. 2

[3] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: A real time system for detecting and tracking people. In *Proc. CVPR*, page 962, Washington, DC, USA, 1998. IEEE Computer Society. 2

[4] I. Haritaoglu, D. Harwood, and L. S. Davis. Hydra: Multiple people detection and tracking using silhouettes. In *VS '99: Proceedings of the Second IEEE Workshop on Visual Surveillance*, page 6, Washington, DC, USA, 1999. IEEE Computer Society. 2

[5] M. Hu. Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, 8(2):179–187, 1962. 3

[6] M. Isard and J. Maccormick. Bramble: a bayesian multiple-blob tracker. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 34–41 vol.2, 2001. 2

[7] H. Kang, D. Kim, and S. Y. Bang. Real-time multiple people tracking using competitive condensation. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*, page 10413, Washington, DC, USA, 2002. IEEE Computer Society. 2

[8] L. Kaufman and P. Rousseeuw. *Finding Groups in Data, An Introduction to Cluster Analysis*. John Wiley and Sons Inc., New York, 1990. 3

[9] S. J. McKenna, Y. Raja, and S. Gong. Object tracking using adaptive color mixture models. In *ACCV '98: Proceedings of the Third Asian Conference on Computer Vision-Volume I*, pages 615–622, London, UK, 1997. Springer-Verlag. 2

[10] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, Mar. 1979. minimize inter class variance. 3

[11] A. Pinz. Object categorization. *Foundations and Trends in Computer Graphics and Vision*, 1(4):255–353, 2006. 1

[12] P. Tu, T. Saxena, and R. Hartley. Recognizing objects using color-annotated adjacency graphs. In *Shape, Contour and Grouping in Computer Vision*, pages 246–263, London, UK, 1999. Springer-Verlag. 2

[13] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. PAMI*, 19(7):780–785, 1997. 2