# **Online-Learned Classifiers for Robust Multitarget Tracking**

Shuqing Zeng<sup>†</sup> and Yanhua Chen<sup>††</sup>

Abstract-In this paper, we propose online-learned classifiers for data association in multitarget tracking. The classifiers are dynamically constructed and incrementally online learned using image patches, which are associated based on location proimity. A biological inspired architecture is used to compute the classification label of image patch. The extracted image patches are coded and learned by a 3-layer neural network that implements in-place learning. We employ minimum-cost network flow optimization to associate tracks with the image patches based on their appearance and location proximities. The presented framework is applied to learn 11 targets encountered in a PETS2009 data set. Cross validation results show that the overall recognition accuracy is above 93%. The comparison with other learning algorithms is promising. The results of the implemented multitarget tracker demonstrate the effectiveness of the approach.

**Key Workds:** Intelligent video surveillance system, object learning, and biologically inspired neural network.

#### I. INTRODUCTION

**T** RACKING targets can be regarded as a dynamic classification of every target against the scene [11]. The ability to robustly track multiple targets is important for vehicle safety systems and video surveillance as it provides the coherent relationships among moving targets in the scene. A challenging aspect of multitarget tracking is to keep a consistent identity for targets. This is difficult when targets have little distinction in their appearance, and meanwhile their appearance undergoes significant changes as change in pose and change in the illumination of the scene. In addition, there are usually occlusion cases where targets pass in front and behind each other or behind of a part of the scene. In this paper, we address this challenging identity maintaining (or data association) problem.

Most existing algorithms to resolve identity ambiguity of targets are position matching. The assumption is that, once a target is assigned to a location in the image, no other targets can occupy that position [12]. The most used techniques impose a one-to-one mapping between targets and observations, formulate associate variable as a parameter of a generative model, and seek maximum likelihood. Recent methods add more assumptions in which tracks are constrained to follow a smooth path and cover a short distance in a short time [3], [7], [17]. By posing constraints on target position and trajectory, these works may succeed in avoiding target merging, but they still yield undesired switching identity, paticularly when the targets are moving in very close range or even interacting with each others [9]. A successful approach requires addressing a combination of the following challenges, where no existing work, as we know, can meet them all: (1) the ability to distinguish targets in close proximity or even in occluding configuration; (2) an incremental appearance model of each target that is adaptive to changes of environment and targets; (3) visual sensory sparse coding via statistical independence of developed features; (4) Online real-time speed; (5) Integration of supervised learning via top-down propagation (e.g., track identity from previous time instants) and unsupervised learning via bottom-up propagation (e.g., subspace learning) in any order suited for development.

We propose to achieve these goals by employing a generic learning network to classify a target based on its identity without using the low-level and mid-level target-specific features (e.g., Haar and Histogram of oriented gradient). We take advantage of background subtraction for fixed cameras to achieve an efficient attention selection on candidate targets. A cortex-inspired neural network integrates 3-way computations (i.e., bottom-up, top-down and lateral) to code target samples in a high-dimensional space and learn the distribution of coded "key" target patterns for favorable recognition performance. Its in-place learning mechanism provides the incremental learning optimality and comparatively low operational complexity even for a very large network.

The rest of this paper is organized as follows. Section II presents the concept of the online-learned classifiers for multitarget tracking. Section III outlines the architecture. Section IV is devoted to the algorithm derivation. The results of experiment are presented in Section V. Finally, concluding remarks are presented in Section VI.

## II. MOTIVATION AND RELATED WORK

Human adults do not experience the world as a disjointed set of moments. Instead, our visual cognitive processes exploit the spatial and temporal coherence in the multiple image frames by adequately identifying the coherent motion regions with simple motion models. Object permanence [1], an awareness of object existence even when the targets are hidden from view, may be a special case of the brain representation of spatio-temporal context through a set of "occlusion detectors", such as those found in the superior central sulcus in the ventral visual pathway [2].

Fig. 1 shows the idea of the paper. Fig. 1(a) shows a scenario that is difficult for target trackers if appearance is not used. The smoothing path assumption is broken in this case. In Fig. 1(b) the flowchart of the proposed method is illustrated. A bank of classifiers run in parallel and monitor image patches as input. Each classifier is dedicated for a

<sup>&</sup>lt;sup>†</sup>Shuqing Zeng is with the General Motors Research and Development Center, 30500 Mound Road, Warren, MI 48090 (email: zengshuq@gmail.com).

<sup>&</sup>lt;sup>††</sup>Yanhua Chen is with the Department of Biostatistics, Univ. of Michigan, Ann Arbor, MI 48109 (email: chenyanh@umich.edu)

specific active target in the field-of-view of camera and detect the presence of the target. The occlusion detector checks if the input image patch overlaps with other patches. The previous assigned identity and the image patch are treated a labeled sample if no overlap is detected. On the other hand, the classifiers output the identity of the patch when overlap presents. In other words, the system dynamically constructs a classifier and learns the classifier using previous identity as label, which are associated based on location proximity (i.e., no occlusion). Meanwhile, the classifiers determine the target's association when occlusion is detected.

For example, in Fig. 1(a), at time t-3, targets 4 and 8 show up in the field-of-view of the camera. Two classifiers for the targets are initialized. Image patches for the two targets from time t-3 to t-1 are used as labeled samples to train the two initialized classifiers. When or after the occlusion occurs at time t, the two trained classifiers are used to assign the image patches to correct identities.



Fig. 1. (a) A scenario that requires targets' appearance for keeping consistent identity. The blue solid lines are the true trajectories while the red dashed lines are estimated trajectories if smooth path is assumed. (b) The flowchart of the proposed method.

The appearance of a target has been considered in single target tracking, particularly by trackers using discriminative approach [5]. The most relevant work is by Nguyen et al. [11]. They propose an incremental probabilistic principal component analysis (PPCA) framework to learn the subspace to describe the appearance of tracks. But this unsupervised learning approach may have ambiguity issues when targets have little distinction in their appearance.

### III. ARCHITECTURE

Fig. 2 shows the architecture of the proposed system. A background subtractor generates a set of image patches enclosing target candidates. The extracted image patches are tracked over frames. The motion correspondence of image patches is sought over multiple frames and sensors by maximizing the distance and appearance similarity. An in-place local neural network provides the processing and learning the sparse-coded spatial representation of the target appearance from the patches, associated with labels derived from track identifiers.



Fig. 2. An outline of the system architecture.

#### IV. Algorithm

## A. Problem Formulation

We formulate the problem as follows. Let I image patches are detected at time instance t. Let  $z_i$  denote the *i*-th image patch,  $1 \leq i \leq K$ . Let  $x_k$  denote the state vector of the k-th track in the current frame. We use binary matrix  $\Phi$ to represent the matching matrix. The entry  $\phi_{ik} = 1$  if  $z_i$ matches  $x_k$  and 0 otherwise. Assuming each image patch  $z_i$ corresponds exactly to one track with indices  $\kappa(i)$ , we have  $\phi_{ik} = \delta_{\kappa(i)k}$  and  $\Sigma_k \phi_{ik} = 1$  for all image patch indices  $i^1$ . If  $x_k$  for  $1 \leq k \leq K$  are given (e.g., the result of last iteration), we can formulate the multitarget data association

by the following minimization problems for  $\Phi$ :

$$\min_{\Phi} \sum_{k=1} \sum_{i=1} \phi_{ik} \| C x_k - z_i \|_{\Gamma}^2$$
(1)

subject to:

$$\sum_{i=1}^{I} \phi_{ik} \le 1, \sum_{k=1}^{K} \phi_{ik} \le 1, \phi_{ik} \in \{0, 1\}$$
(2)  
$$1 \le k \le K, 1 \le i \le I$$

where C is the measurement matrix of appropriate dimension, and  $\Gamma$  is the covariance matrix of the measurement noise, which can be reduced to a minimum-cost flow problem [17] and solved using the software<sup>2</sup> that implements the

 ${}^{1}\delta$  is the discrete Dirac delta function, i.e.,  $\delta_{ik} = \begin{cases} 1 \text{ if } i = k \\ 0 \text{ Otherwise.} \end{cases}$ 

<sup>&</sup>lt;sup>2</sup>Andrew Goldberg's Network Optimization Library at http://www.avglab.com/adrew/soft.html. We use *hi-pr* and *cs2* for maximum flow and minimum-cost flow computation, respectively.

scaling push-relabel method proposed in [6] to compute the minimum-cost flow.

There are two parts of the measurement of the target: appearance  $z_a$  (i.e., the target identity generated by the classifiers) and location  $z_l$  (i.e., column and row of the patch), which we want to treat separately. In this paper, the following expression is used to compute the dissimilarity measure between  $z_i$  and  $x_k$ :

$$c_{ik} = \frac{\|x_{a,k} - z_{a,i}\|}{\sigma_a^2} + \frac{\|x_{l,k} - z_{l,i}\|}{\sigma_l^2}$$
(3)

where  $x_{a,k}$  and  $x_{l,i}$  are the corresponding target identity and location for the k-th track, respectively, and  $\sigma_a$  and  $\sigma_l$  are weight constants.

### B. Representation of Spatial Context

Fig. 3 shows that the targets in PETS 2009 data set<sup>3</sup> have little distinction in their appearance, and meanwhile each person's appearance undergoes significant changes as in different poses and view angles. In this paper, we propose a biological inspired architecture to compute the descriptor of the patches z, which is the output vector (classification labels) of the proposed neural network in Fig. 4.



Fig. 3. Examples of a target in PETS 2009 data set.

The extracted window images are learned through the proposed neural network (see Fig. 2) via 3 layers where each neuron in the output layer corresponds to one target identity. Fig. 4 shows the structure of the network connection with three consecutive layers [15], [16]. Every neuron at layer l is connected with four types of connection weights:

- Bottom-up weight vector w<sub>b</sub><sup>(l)</sup> that links connections from its bottom-up field in the previous level.
- Top-down weight vector wt<sup>(l)</sup> that links connections from its top-down field in the next level.
- 3) Lateral weight vector  $\mathbf{w}_{\mathbf{h}}^{(l)}$  that links inhibitory connections from neurons in the same layer (larger range).
- Lateral weight vector w<sub>e</sub><sup>(l)</sup> that links excitatory connections from neurons in the same layer (smaller range).

Note that each linked weight pair (i, j) shares the same value, i.e.,  $\mathbf{w}_{t,j}^{(l-1)} = \mathbf{w}_{\mathbf{b}_{j,i}^{(l)}}$ . Moreover, this work does not use explicit lateral connections, but instead uses an approximate method: the top-k winners (i.e., k largest responses) along with their excitatory neighbors update and fire. The

suppressed neurons are considered laterally inhibited and the winning neurons are considered laterally excited.



 Parameters: N: No. neurons X: Bottom-up fields Z: Top-down fields

 I: Inhibitory fields
 E: Excitatory fields

 k: Top-k winners in I
 α: Top-down influence

Fig. 4. An example of layer representations (i.e., responses) in the proposed neural network, including a specific set of resource parameters implemented. Green and red directed lines show the bottom-up and top-down connections to the firing neurons, respectively. It is noted that the bottom-up fields of layer 1 neurons are  $16 \times 16$  local areas over the entire  $64 \times 32$  image plane, with a stagger distance per 8 pixels, and the top-down fields are not available in layer 1 and layer 3. In addition, neural representations in layer 1 are reshaped to  $36 \times 431$  for visualization purpose.

The learning network is incrementally updated at discrete times, t = 0, 1, 2, ..., taking inputs sequentially from image patches and track identities, computing responses of all neurons, and producing internal and external actions through experience. Fig. 4 shows an example of network computation, layer by layer, as well as key parameters used in the network implementation.

As described in Algorithm 1, layer 1 of the proposed network develops earlier than other layers, which is inspired from the biological fact that early cortical regions in the brain (e.g., primary visual cortex) would develop earlier than the later cortical regions [8]. Given t = 1, 2, ...500000, the network receives  $64 \times 32$ -pixel (same as attention window dimension) natural image patches, which were randomly selected from the thirteen natural images<sup>4</sup>. Neurons are learned through the in-place learning algorithm described in Algorithm 2, however, without supervision on labels. After 500000 updates of layer 1 neurons, their bottom-up features tends to converge. Then the network perceives extracted patches and all the layers are developed through the same in-place learning procedure in Algorithm 2, whereas supervised signals from the track's identifier.

The network performs an open-ended online learning while internal features "emerge" through interaction with its extracellular environment. All the network neurons share the same learning mechanism and each learns on its own, as a self-contained entity using its own internal mechanisms. In-

<sup>&</sup>lt;sup>3</sup>Available at http://www.cvg.rdg.ac.uk/PETS2009/

<sup>&</sup>lt;sup>4</sup>Available at http://www.cis.hut.fi/projects/ica/imageica/

# Algorithm 1 Network processing procedure

1: for  $t = 1, 2, \dots 500000$  do

- 2: Grab a whitened natural image patch s(t).
- 3: for l = 1 do
- 4: Get the bottom-up fields  $\mathbf{x}(t)$  from  $\mathbf{s}(t)$ . The top-down fields  $\mathbf{z}(t)$  are set to **0**.
- $(\mathbf{y}(t+1), L(t+1)) =$ In-place $(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t) | L(t)),$ 5: where L(t) presents the state of current layer l, including its bottom-up and top-down weighs, neural ages, etc.
- 6: end for
- 7: end for
- 8: for  $t = 500001, 500002, \dots$  do
- 9: Grab the attention window image s(t).
- 10: Impose the label vector  $\mathbf{m}(t)$  to layer 3.
- for 1 < l < 3 do 11:
- if l = 1 then 12:
- 13: Get the bottom-up fields  $\mathbf{x}(t)$  from  $\mathbf{s}(t)$ . The top-down fields  $\mathbf{z}(t)$  are set to **0**.

14: else if l = 2 then

- Get the bottom-up fields  $\mathbf{x}(t)$  from the previous layer 15: representation (responses) and the top-down fields z(t)from  $\mathbf{m}(t)$ .
- 16: else Get the bottom-up fields  $\mathbf{x}(t)$  from the previous layer 17: representation (responses). The top-down fields z(t) are set to 0. 18:
- end if  $(\mathbf{y}(t+1), L(t+1)) = \text{In-place}(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t) | L(t)).$ 19: 20end for
- 21: end for

place learning, representing a new and deeper computational understanding of synaptic adaptation, is rooted in the genomic equivalence principle [13]. It implies that there cannot be a "global", or multi-cell, goal to the learning, such as the minimization of mean-square error for a pre-collected (batch) set of inputs and outputs. Instead, every neuron is fully responsible for its own development and online adaptation while interacting with its extracellular environment.

#### V. EXPERIMENTAL RESULTS

#### A. Cross validation

In this experiment, a ten-fold cross validation is performed to evaluate the system performance. All the samples are shuffled and partitioned to 10 folds/subsets, where 9 folds are used for training and the last fold is used for testing. This process is repeated 10 times, leaving one fold for evaluation each time. The cross validation result is shown in Fig. 5. The average recognition rate is 94.01%, and the average false positive and false negative rates are 2.94% and 6.72%, respectively.

#### B. Performance comparison

With regards to an open-ended visual perceptual development, an incremental (learning one image perception per time), online (cannot turn the system off to change or adjust), real-time (fast learning and performing speed), and extendable (the number of classes can increase) architecture is expected. We compare the following incremental learning

Algorithm 2 In-place learning procedure: (y(t + 1), L(t +1)) = In-place( $\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t) \mid L(t)$ )

- 1: for  $1 < i < N_l$  do
- Compute pre-response of neuron i from bottom-up and top-2: down connections:

$$\hat{y}_{i}^{(l)}(t+1) = g_{i} \left( (1-\alpha_{l}) \frac{\mathbf{w}_{\mathbf{b}_{i}}^{(l)}(t) \cdot \mathbf{x}_{i}^{(l)}(t)}{\|\mathbf{w}_{\mathbf{b}_{i}}^{(l)}(t)\|\|\mathbf{x}_{i}^{(l)}(t)\|} + \alpha_{l} \frac{\mathbf{w}_{\mathbf{t}_{i}}^{(l)}(t) \cdot \mathbf{z}_{i}^{(l)}(t)}{\|\mathbf{w}_{\mathbf{t}_{i}}^{(l)}(t)\|\|\mathbf{z}_{i}^{(l)}(t)\|} \right)$$
(4)

where  $\mathbf{x}_{i}^{(l)}(t)$  and  $\mathbf{z}_{i}^{(l)}(t)$  are bottom-up and top-down input fields of neuron *i*.  $g_i$  is a sigmoid function with piecewise linear approximation.  $\alpha_l$  is a layer-specific weight that controls the influence of top-down part.

#### 3: end for

4: Simulate lateral inhibition and decide the winner:

$$j = \arg \max_{i \in I^{(l)}} \hat{y}_i^{(l)}(t+1)$$

- 5: The cells in excitatory neighborhood  $E^{(l)}$  are also considered
- The responses  $y_j^{(l)}$  of winning neurons are copied from their pre-responses  $\hat{y}_j^{(l)}$ .
- 7: Update the number of hits (cell age)  $n_i$  for the winning neurons:  $n_j \leftarrow n_j + 1$ . Compute  $\mu(n_j)$  by the amnesic function:

$$\mu(n_j) = \begin{cases} 0 & \text{if } n_j \le t_1, \\ c(n_j - t_1)/(t_2 - t_1) & \text{if } t_1 < n_j \le t_2, \\ c + (n_j - t_2)/r & \text{if } t_2 < t, \end{cases}$$
(5)

where parameters  $t_1 = 20, t_2 = 200, c = 2, r = 2000$  in our implementation.

8: Determine the temporal plasticity of winning neurons, based on each age-dependent  $\mu(n_j)$ :

$$\Phi(n_j) = (1 + \mu(n_j))/n_j,$$

9: Update the synaptic weights of winning neurons using its scheduled plasticity:

$$\mathbf{w}_{\mathbf{b}_{j}}^{(l)}(t+1) = (1 - \Phi(n_{j}))\mathbf{w}_{\mathbf{b}_{j}}^{(l)}(t) + \Phi(n_{j})\mathbf{x}_{j}^{(l)}(t)y_{j}^{(l)}(t+1)$$
(6)

10: All other neurons keep their ages and weight unchanged.

v



Fig. 5. 10-fold cross validation of the proposed work.

AVERAGE PERFORMANCE & COMPARISON OF LEARNING METHODS OVER "TRUE DISJOINT" TEST

Learning	Overall	Training time	Testing time
method	accuracy	per sample	per sample
K-NN	$75.65 \pm 13.37\%$	n/a	$912\pm14.1\mathrm{ms}$
ISVM	$72.32 \pm 10.76\%$	$175.2\pm17.4\mathrm{ms}$	<b>3.1</b> ± <b>0.5</b> ms
IHDR	$81.34 \pm 7.23\%$	<b>5.8</b> ± <b>3.1</b> ms	$8.1 \pm 4.3$ ms
Proposed network	93.23± 1.57%	$131\pm9.1\mathrm{ms}$	$53.4\pm9.5\mathrm{ms}$

methods in MATLAB to classify the extracted window images  $(64 \times 32)$  as target identities: (1) K-Nearest Neighbor (K-NN), with K=1, and using a L1 distance metric for baseline performance; (2) Incremental Support Vector Machines (I-SVM) [4]; (3) Incremental Hierarchical Discriminant Regression (IHDR) [14] and (4) the proposed network described in this paper. We used a linear kernel for I-SVM, as is suggested for high-dimensional problems [10]. We did try several settings for an Radial Basis Function (RBF) kernel, but the system training becomes extremely slow and the performance improvement is not obvious.

Instead of randomly selecting samples in cross validation, we used a "true disjoint" test, where the time-organized samples are broken into ten sequential folds. Each fold is used for testing per time. In this case, the problem is more difficult, since sequences of targets in the testing fold may have never been seen. This truly tests generalization.

The results are summarized in Table I. K-NN performs fairly well, but is prohibitively slow. IHDR combines the advantage of K-NN with an automatically developed tree structure, which organizes and clusters the data well. It is extremely useful for the fast retrieval due to its logarithmic complexity. IHDR performs the recognition better than K-NN, and is much faster for the real-time training and testing. However, IHDR typically takes a lot of memory. It allowed sample merging of prototypes, but in such case it saved every training sample, and thereby did not use memory efficiently. I-SVM performed the worst on our high-dimensional data with amount of noise, but the testing speed is fastest, since its decision making is only based on a small number of support vectors. A major problem with I-SVM is its lack of extendibility. By only saving support vectors to make the best two-class decision boundary, it throws out information that may be useful in distinguishing other classes that are added later.

Overall, the proposed network is able to perform the recognition better than all other methods using only  $15 \times 15$  layer 2 neurons with a top-down supervision parameter  $\alpha = 0.3$ . It is also fairly fast, and efficient in terms of memory. The proposed work does not fail in any criteria, although it is not always the "best" in each category. The proposed work also has its major advantages in extendibility. New target classes can be added later without changing the existing learning structure of the network.

## C. Tracking Result



Fig. 6. Result of background subtraction.

Figure 6 shows the detected image patches via background subtraction. The rectangle bound boxes are the projections of a fixed size window (width=0.8 meter, height=1.85 meters) in world coordinate system to the corresponding bottom position (u, v) of the detected blobs. Figure 7 shows the tracks derived from PETS 2009 data set (S2/L1/Time\_12-34) using the learned classifier. 11 classifiers are used corresponding to the persons appeared in the images. As shown in Figure 7, the system successfully classifies the identity of the persons, and links disjoint track paths before and after occlusion.

### VI. CONCLUSIONS

In this paper, we proposed and demonstrated a novel data association framework for multitarget tracking using an online learned classifier. The learned classifiers are implemented using a biological inspired neural network. In the proposed neural network, extracted image patches are sparsely coded by the features in the first layer that were developed from the statistics of natural images. In the second layer the network is further learned in reaction to the coupled sparse





(a) view1, frame4



(c) view1, frame24



(e) view5, frame40



(g) view5, frame55



(i) view8, frame14



(k) view8, frame40

Fig. 7. Tracks generated from PETS 2009 data set.

(b) view1, frame20



(d) view1, frame48



(f) view5, frame44



(h) view5, frame96



(j) view8, frame25



(l) view8, frame57

representation and external class representations, where each cell in the network is a local class-abstracted density estimator. The proposed system architecture allows incremental and online learning, which is efficient and effective for real-time target tracking.

#### REFERENCES

- R. Baillargeon, E. Spelke, and S. Wasserman. Object permanence in five-month-old infants. *Cognition*, 20:191–208, 1985.
   C. Baker, C. Keysers, J. Jellema, B. Wicker, and D. I. Perrett. Neuronal
- [2] C. Baker, C. Keysers, J. Jellema, B. Wicker, and D. I. Perrett. Neuronal representation of disappearing and hidden objects in temporal cortex of macaque. *Exp. Brain Res.*, 140:375–381, 2001.
  [3] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow
- [3] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Proceedings of IEEE International Workshop* on *Performance Evaluation of Tracking and Surveillance*, Snowbird, UT, 2009.
- [4] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In Advances in Neural Information Processing Systems, volume 13, pages 409–415, Cambridge, MA, 2001.
- [5] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.
- [6] A. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. J. Algorithm, pages 1–29, 1997.
- [7] H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. In CVPR, 2007.
- [8] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, editors. *Principles of Neural Science*. McGraw-Hill, New York, 4th edition, 2000.
  [9] Z. Khan, T. Balch, and F. Dellaert. Mcmc data associatin and
- [9] Z. Khan, T. Balch, and F. Dellaert. Mcmc data associatin and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE Trans. Pattern Analysis* and Machine Intelligence, 28(12):1960–1972, 2006.
- [10] B. L. Milenova, J. S. Yarmus, and M. M. Campos. Svm in oracle database 10g: Removing the barriers to widespread adoption of support vector machines. In *Proc. 31st VLDB Conference*, 2005.
- [11] H. Nguyen, Q. Ji, and A. Smeulders. Spatio-temporal context for robust multitarget tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(1):52–64, 2007.
  [12] S. Roy and I. Cox. A maximum-flow formulation of the n-camera
- [12] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *International Conference on Computer Vision*, pages 492–499, 1998.
  [13] D. E. Sadava, H. C. Heller, G. H. Orians, W. K. Purves, and D. M.
- [13] D. E. Sadava, H. C. Heller, G. H. Orians, W. K. Purves, and D. M. Hillis. *Life, the science of biology*. Freeman, New York, 8th edition, 2006.
- [14] J. Weng and W. Hwang. Incremental hierarchical discriminant regression. *IEEE Trans. Neural Networks*, 18(2):397–415, 2007.
- [15] J. Weng and M. Luciw. Neuromorphic spatiotemporal processing. Technical report, MSU-CSE-08-34, 2008.
- [16] J. Weng and M. Luciw. Dually optimal neuronal layers: Lobe component analysis. *IEEE Trans. on Autonomous Mental Development*, 1(1):68–85, 2009.
- [17] L. Zhang, Y. Li, and R. Nevatia. Global data association for multiobject tracking using network flows. In CVPR, 2008.