Abstract. This paper presents a novel approach to learning a dictionary of crowd prototypes for dynamic visual scenes. Recent work in cognitive psychology suggests that crowd perception may be based on pre-attentive ensemble coding mechanisms [24] in the spirit of feedforward hierarchical models of visual processing [4]. We extend a biological model of motion processing [10] with a new dictionary learning method tailored for crowd perception. Our approach learns crowd prototypes through explicit ensemble coding mechanisms via structural and local coherence constraints. We evaluate the proposed method on multiple crowd perception problems from collective or abnormal crowd detection to tracking individuals in crowded scenes. Experimental results on crowd datasets demonstrate competitive results on par or better than the state of the art.

Keywords: Sparse Prototypes, Ensemble Coding, Crowd Perception

1 Introduction

The perception of crowd behavior has become a popular area of study straddling multiple disciplines from cognitive psychology to computer vision. Over the years, many approaches to crowd perception have been proposed. Early computer vision studies were motivated by sociology [8] or physics [1]. Social models aim at characterizing the interaction between individuals in a crowd. This can be done explicitly using either systems of non-linear coupled equations as in the “social force” model [18, 20] or implicitly via dynamic space-time correlations [12]. More recent work has extended some of these ideas using methods borrowed from group structure learning [27], visual saliency [17] or other energy-based approaches [5, 19]. A measure of intended motion using space-time statistics [13] has also been proposed as a model of people’s “efficiency”. The “collectiveness” of crowds has been estimated using manifold similarity-based measures [34]. A few notable physics-based approaches include the approach by Wu et al. who describe a chaotic invariant descriptor within the framework of particle advection [26] and the approach by Solmaz et al. who use stability analysis to identify different patterns of behaviors [23].

More recently, the “word document” framework has been used to characterize semantic correlations and dependencies between latent motion patterns [9, 21]. Several approaches for learning spatial-temporal occurrences of crowd motion
patterns have also been proposed [15, 11, 2]. A notable approach based on the structural flow of scenes has been proposed in [32] and an approach for learning typical prototypes from correlations in atomic activities in [28].

One promising class of models focuses on learning crowd prototypes using sparse coding [3, 16, 31] and closely related techniques using linear programming or matrix factorization [28, 29]. A sparse reconstruction error measure is used to assess abnormal crowd behaviors in [3, 31]. Lu et al. propose an efficient sparse combination learning framework to detect abnormal events from pyramid video structures [16]. One of the main limitations with these methods is that they typically focus on modeling local motion patterns when patterns of crowd behavior tend to be more global. In general the learned representations tend to be relatively unstable over time. Learned prototypes seldom produce rich mid-level representations of the crowd patterns, and typically fail to capture typical crowd peculiarities. To address these issues, we investigate the properties of ensemble coding mechanisms with a hierarchical model to capture the structural and collective characteristics of crowds.

Our approach is motivated by recent developments in cognitive psychology suggestive of the existence of pre-attentive ensemble coding mechanisms for crowd perception [24]. It was shown that participants estimate the intended direction of briefly presented crowds of point-light walkers better than the behavior of an individual. Such results suggest that observers rapidly pool information from multiple walkers to estimate the movement of a crowd, very much in the spirit of feedforward hierarchical models [4]. Feedforward hierarchical models of visual processing (such as computational models of the visual cortex [10] and closely related convolutional networks [25]) have been shown to exhibit competitive performance for the recognition of individual human or animal activities. To our knowledge, these have not been extended to the recognition of complex patterns of crowd behaviors.
Related work and main approach: We propose a significant extension of a popular feedforward hierarchical model of the visual cortex [10] from the recognition of individual behaviors to group behaviors. Direction-selective motion units based on optical flow calculations are used as an input stage. We propose a novel learning method to learn crowd prototypes in intermediate stages of the motion processing hierarchy based on a sparse coding model. The proposed optimization learns crowd prototypes through ensemble coding mechanisms by jointly enforcing local structure and coherence in the patterns of crowd motion. Most closely related to our learning framework are learning approaches described above based on spatial-temporal representations of crowd motion patterns [12, 21], structural flow [32] and correlations between atomic activities [28, 34]. Compared to the original approach by [10], we show that the resulting learned prototypes are more selective and more easily interpretable. As we will show, a hierarchical architecture leads to crowd prototypes which are compact and can capture the coherence and the underlying structure of motion patterns associated with crowded scenes.

In summary, this paper makes the following contributions: (1) We describe a novel mid-level representation together with an algorithm for learning crowd prototypes within a feedforward hierarchical model of motion processing; (2) Ensemble coding mechanisms are incorporated within a dictionary learning approach together with coherent and structural constraints to learn discriminative and compact prototypes; (3) The versatility of the learned crowd prototypes is validated on several crowd perception problems including measures of crowd collectiveness, the detection of abnormal motion patterns, and tracking in crowded scenes.

2 Hierarchical model of crowd processing

An overview of the system is shown on Figure 1. The basic visual representation is based on [10], which we only review briefly here. The model starts with motion-sensitive simple (S1) and complex (C1) units similar to those found in the primary visual cortex. While Jhuang et al. compared several implementations of motion-sensitive S1 units, here we consider their implementation based on optical flow, because it is particularly amenable to existing approaches for crowd perception [1, 12, 21]. Specifically, we build a population of motion-sensitive units tuned to both speed and motion direction using the optical flow estimated from local space-time 3D volumes randomly or selectively (located by the gKLT tracker as used in [34] for specific application).

Let \( \theta_{i,j} \) and \( v_{i,j} \) denote the orientation and velocity of the optical flow at image location \((i, j)\). As done in [7, 10], simple unit responses are then obtained using the following quantization:

\[
r_{S1}^{i,j}(\theta_p, v_p) = \frac{1}{2^{1 + \cos(\theta_{i,j} - \theta_p)}} q \times \exp(-|v_{i,j} - v_p|)
\]

(1)

where \( \theta_p \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\} \) and \( v_p \in \{3, 6\} \) are the preferred directions and magnitudes, while the constant \( q \) controls the width of the tuning curve [7]. S1
unit responses are computed for an $l \times l$ pixel neighborhood. In the following stage, C1 unit responses are computed by performing a local max pooling on the S1 unit responses across speeds and a local spatial neighborhood.

In subsequent processing stages, units of higher visual complexity emerge after an additional template-matching (S2 units) as well as an invariance pooling (C2 units) stage, which increases both the selectivity and invariance properties of the underlying model units. The response of S2 units is obtained by convolving C1 maps across all motion directions with a dictionary of stored prototypes. Originally, the dictionary of $K$ S2 prototypes is learned via a simple random sampling procedure. Here, instead, we propose to learn crowd prototypes via sparse coding methods, which we describe next.

3 Learning crowd prototypes

3.1 Sparse coding model

Given a set of $N$ input unit vectors $R = [r_1, r_2, \ldots, r_N] \in \mathbb{R}^{M \times N}$, learning a sparse dictionary of coding elements can be formulated as the following optimization problem:

$$
B^*, S^* = \arg \min_{B, S} \|R - BS\|_2^2 + \lambda \sum_i \|s_i\|_1, \text{s.t. } \forall i, s_i \geq 0
$$

where $S = [s_1, s_2, \ldots, s_N] \in \mathbb{R}^{K \times N}$ correspond to linear coefficients and $B = [b_1, b_2, \ldots, b_K] \in \mathbb{R}^{D \times K}$ is a matrix containing the learned basis functions as column vectors. $\lambda$ is a constant to control the tradeoff between the reconstruction error and the regularization term (here a sparsity constraint).

3.2 Model extensions

Here, we propose to incorporate the idea of ensemble coding in the form of two additional similarity preserving constraints embedded in the sparse coding objective function. Cognitive psychology experiments have suggested the existence of pre-attentive pooling mechanisms used by our visual system to force complex chaotically moving crowds to cohere into a unified and visually appealing Gestalt [24]. A structural neighborhood cohesion term forces input patterns to be similar. This corresponds to a direct implementation of reported phenomena from cognitive psychology that have shown that participants rapidly pooled information from multiple walkers to estimate the heading of a crowd [24]. A neighbor manifold coherence term incorporates explicit pooling mechanisms over output vectors to yield a more locally stable code.
Overall, these two constraints are embedded in the following optimization problem:

\[
R^*, B^*, S^* = \arg \min_{R, B, S} \left( \|R - BS\|_F^2 + \lambda \sum_{i=1}^{N} \|s_i\|_1 + \beta \sum_{i=1}^{N} \|r_i - r'_i\|_M^2 + \gamma \sum_{i=1}^{N} \sum_{j=1}^{N} \|s_i - s_j\|_2^2 W_{ij} \right) \\
\text{s.t. } \|b_i\|^2 \leq c, i = 1, \ldots, K
\] (3)

Here \(r'_i\) correspond to the average of all \(r_i\) within a spatial neighborhood for unit \(i\). \(\lambda\), \(\beta\), \(\gamma\) are the regularization parameters used to control the weights of different regularization terms. Here the learning algorithm is initialized by setting up the model C1 unit responses that are tuned to different directions of motion over a spatial neighborhood as \(r_j\), such that \(R\) is the matrix containing all \(N\) C1 unit vectors as columns. \(S\) can be thought as the responses of the S2 units.

In the above objective function, the first term is an estimate of the reconstruction error when encoding the S2 unit responses using learned prototypes and associated coefficients. The second term corresponds to a standard sparsity constraint on the coefficients, which constraints the number of prototypes used to encode a visual sample \(r_i\) to be small. We call the third and fourth terms structural and coherency constraints.

The coherency constraints should, in principle, encourage unit responses within a local neighborhood to follow a manifold characterized by the intrinsic characteristics of the motion feature while enforcing local consistency of the flow. Here we used the graph-based Laplacian regularization to formulate this constraint. In addition, a structural constraint is formulated as a generalized Tikhonov regularization problem. The term can be thought of as helping to learn crowd patterns with locally similar individual movements. This can be also thought as a denoising term, which smoothes out the local motion flow towards a common vector \(r'_i \in R'\) over a local neighborhood (weighted by a Gaussian function over space):

\[
r'_i = \arg \min_{r_i} \frac{1}{d} \sum_{j \in \mathcal{N}(i)} \exp(-\frac{\|r_i - r_j\|^2}{2\sigma^2}),
\] (4)

where \(\mathcal{N}(i)\) correspond to the set of indexes for the \(d\) nearest neighbors around \(r_i\), and \(\sigma\) is a constant.

### 3.3 Alternating Optimization

Because the objective function in Eq. 3 is not convex with respect to \(R\), \(B\) and \(S\), we use an alternative minimization approach here. The formulation becomes
convex when optimizing one variable while fixing the others. It can be converted to two alternative optimization problems, both of which are convex. We can thus alternate between Step (A) and (B) iteratively as described below: Step (A) finds the structural C1 response $R$. Eq. 3 can be rewritten by replacing the fixed term $BS$ by $b$ as detailed in the following matrix form:

$$
R^* = \operatorname{arg}\min_{R} \|R - b\|^2_F + \beta\|R - R'\|^2_M,
$$

(5)

where $\|R - R'\|^2_M = (R - R')^TQ^{-1}(R - R')$ is the Mahalanobis distance between $R$ and $R'$. $Q$ is the covariance matrix computed over $R$. In Step (A), after the neighborhood pooling (see Eq. 4) at each iteration, a closed-form solution can be computed using the generalized Tikhonov regularization as:

$$
R^* = R' + (I + \beta(B)^{-1})^{-1}(b - R')
$$

(6)

In Step (B), let $W \in \mathbb{R}^{N \times N}$ be the weight matrix corresponding to the pairwise nearest neighbors between $r_{i,j}$ with entry $W_{ij}$. The degree of $r_i$ is defined as $d_i = \sum_{j=1}^{N}W_{ij}$, and $D = \text{diag}(d_1, \ldots, d_N)$. So this terms can be rewritten as follow:

$$
\sum_{i=1}^{N} \sum_{j=1}^{N} ||s_i - s_j||^2W_{ij} = Tr(S^TLS)
$$

(7)

where $L = D - W$ is the Laplacian matrix. By fixing $R$ and incorporating the Laplacian regularizer, we update $B$ and $S$ according to:

$$
\begin{align*}
\arg\min_{B,S} & \|R - BS\|^2_F + \lambda \sum_i \|s_i\|_1 + \gamma Tr(S^TLS), \\
\text{s.t.} & \|b_i\|^2 \leq c, i = 1, \ldots, K
\end{align*}
$$

(8)

The above optimization is a typical laplacian regularization problem, which can be solved using the feature sign search algorithm [14]. The overall prototype learning approach is summarized in Algorithm 1. As $\beta \to 0$, Eq. 3 degenerates into a typical graph-based sparse coding approach. Similarly as $\beta, \gamma \to 0$, Eq. 3 degenerates to a standard sparse coding method.

### 3.4 Crowd pattern representation

Here, given a newly observed pattern of C1 unit responses $R$ and the current prototype set $B$, we learn the corresponding optimal reconstruction coefficients $S$ as S2 responses. Thus, crowd pattern labels can be assigned by considering the largest value of the coefficients corresponding to the prototype membership distribution. A sparse representation for crowd patterns can be assigned through a local max pooling at the C2 stage. As we will show next, it is possible to further extend this representation to various crowd perception problems with little changes.
Algorithm 1: Crowd Prototype Learning

1. **Input:** Given \( N \) patches of C1 units \( R = [r_1, r_2, \ldots, r_N] \in \mathbb{R}^{D \times N} \) and required parameters;
2. Initialize \( B = [b_1, b_2, \ldots, b_K] \in \mathbb{R}^{D \times K} \) and \( S = [s_1, s_2, \ldots, s_N] \in \mathbb{R}^{K \times N}; \)
3. repeat
   4. **Step (A):**
      5. Given \( B, S \), compute \( R' \) with the \( d \) nearest neighbors pooling of \( R \) according to Eq. 4;
      6. Solve \( R^* \) by generalized Tikhonov regularization in Eq. 5;
      7. According to Eq. 6, Update \( R \) with \( R^* \);
   8. **Step (B):**
      9. Given \( R \), solve \( B, S \) in Eq. 8 using feature sign search algorithm [14];
   10. Update \( B, S \);
   11. Iteration number \( i++ \);
4. until Change in \( S \) between two successive iterations is smaller than \( \varepsilon \) or max iteration number reached;
5. **Output:** Optimized \( R \in \mathbb{R}^{D \times N} \), crowd prototypes \( B \in \mathbb{R}^{D \times K} \), S2 Response Coefficient \( S \in \mathbb{R}^{K \times N}; \)

4 Experimental Results

In order to evaluate the effectiveness of the proposed crowd prototype learning approach, we carry out three experiments which include: (1) the detection of collective motion patterns, (2) the detection of abnormal crowd behaviors and (3) the tracking of individuals in crowds.

4.1 Parameter Settings

At the S1 stage, \( 5 \times 5 \) pixel patterns are convolved with nonlinear transform filters at 4 motion directions and 2 speeds. This leads to \( 5 \times 5 \times 4 \times 2 \) S1 maps for the corresponding video volume. Beyond this S1 stage, a max pooling operation over an \( 8 \times 8 \) grid (with 4 spatial steps overlap) and 2 speeds to build some tolerance to small motion distortions. At the S2 and C2 stages, crowd prototypes are learned by considering \( 5 \times 5 \times 4 \) patterns of C1 unit responses. For the subsequent neighborhood pooling, the number of C1 units neighbors \( d \), and their weights \( \sigma \) for each iteration in the alternative processing control the structured level of crowd patterns in Eq. 6. Empirically, we fixed to be 20 and \( \sigma^2 = 0.1 \). And also we found that keeping \( \lambda, \beta, \gamma \) to be 0.01, 0.5 and 0.1 in Eq. 3 yields good results during the training. The number of the learned prototypes \( K \) varies according to different tasks. These prototypes represent the intermediate-level representation of the crowd patterns, we use the sparse coefficient as the response of each prototype.

4.2 Illustrative Sample

We illustrate the learning process with a simple example in Figure 2. Given a set of C1 unit responses on the Marathon sequence [1], we iteratively optimize
(a) Sample convergence results on the Marathon sequence (first 5 iterations from left to right).

(b) Comparison between different prototype learning methods. From left to right: standard optical flow, basic Sparse Coding [14], Graph-based Sparse Coding [33], proposed crowd prototype learning.

(c) The convergence of the structural loss and objective value in term of iterations.

**Fig. 2.** Illustrative convergence results for the proposed approach (first 5 iterations).

The objective function for 8 prototypes, and assign the largest value of the S2 coefficient as the pattern labels. Notice that in Figure 2 (a), as the iteration number increases, the structural loss becomes smaller and smaller, which implies that the underlying patterns are much more similar to the neighborhood. At the same time, the patterns increasing resemble the collective patterns as indicated by the decrease of the objective function value, which means that the estimated prototypes converge during the iterations. Detailed values is illustrated in Figure 2 (c). In Figure 2 (b), we show the crowd patterns inferred by optical flow, basic Sparse Coding [14], Graph-based Sparse Coding [33] and our crowd prototype learning, respectively. This illustrates that our prototypes can effectively give more meaningful crowd patterns for structure and topology compared to the other three methods. In addition, the optimization converges quickly in less than 5 iterations. We fix the iteration number as 5 in practice for all the following settings.

### 4.3 Collective Motion Detection

In this subsection, we apply our crowd prototypes on various pedestrian videos and show the results of collective motion detection. We also measure collectiveness compared to the ratings derived from human participants.
Fig. 3. Collective crowd movement detection results using the proposed approach. Color encodes the index of the corresponding prototype.

Fig. 4. ROC curves for the classification of collectiveness levels. We compare a “prototype” score $P$ derived using the proposed approach with a “collectiveness” score $C$ and the “normalized velocity” $V$ (see text for details).

**Detection Dataset:** We detect the collective motion in the new Collective Motion Dataset [34]. The video set consists of 413 videos from 62 crowded scenes including the mall, traffic, escalators, campus, etc. Each video contains 100 frames with the ground truths labeled by 10 subjects for the levels of low, medium and high collectiveness.

**Collectiveness Measurement:** For training the prototypes, we randomly select $C_1$ units with average norm $> 0.1$ (ensure the value of the motion response is big enough) centered on keypoints which are returned by the gKLT tracker [34] in 62 crowded scenes. The prototype size is kept as 8. Sampled $C_1$ units are assigned by the learned prototypes with the largest coefficient label for each pattern in the first 10 frames, which is indicated with different colors in the examples. Intuitively, our prototypes can distinguish a variety of behaviors, such as patterns of crossing, lane, arch, etc. [23]. Figure 3 shows the detected collective motion in the dataset. The estimated crowd patterns illustrate coherent motion in structured scene, which can be further applied to crowd behavior classification and scene understanding. To quantitatively evaluate the effectiveness of our proposed crowd prototype on detecting the coherent motion. We measure the
collectiveness by utilizing the computed prototypes to infer the S2 unit coefficients for all the rest C1 units. We define our collectiveness metric score as $P$ following the algorithm in [34],

$$
P = \frac{1}{|\Omega|} \sum e^T((I - \varepsilon W_p)^{-1} - I)e
$$

(9)

where $W_p$ denotes the adjacency matrix of the graph constructed by the set $\Omega$ of S2 units. $e$ is the vector with all elements as 1. The edge $w_p(i,j)$ is the correlation between the coefficients of S2 unit $i$ and $j$, which is computed by the $\chi^2$ distance.

**Insight into prototype based collectiveness:** We compare our prototype score $P$ with the collectiveness descriptor score $C$ and the normalized velocity $V$ reported in [34]. The ROC curves for the binary classification between low, medium and high categories are shown in Figure 4. For the High-Medium and High-Low categories, our prototypes are competitive with the state-of-the-art collectiveness descriptor $C$. This is because our prototype coefficients distinguish different levels of dynamic motion while preserving the consistency and structure of the crowd. Although the Medium-Low category is hard to classify, our prototype also achieves better results. The reason comes from the alternate S and C layers which increase the selectivity and invariance of the motion feature along the hierarchical model. That effectively overcomes the influences of perspective distortion and local small activities in the regular flow.

### 4.4 Global Abnormal Event Detection

We also conduct a set of experiments for the applications of Global Abnormal Event (GAE) detection to validate the performance of crowd prototypes. The goal is to detect an abnormal frame within a stream.

**GAE on UMN and PETS 2009 Dataset:** In this experiment, we utilize the frame-level measurement [3] to test GAE detection on the UMN dataset¹, which consists of 11 clips of crowded escape video events acquired in 3 different scenarios. Each video begins with normal behaviors and ends with panic escaping. The patterns are relatively simple, so we use it for evaluating the overall performance compared with state-of-the-arts and the learning part. All the video frames are resized to $120 \times 160$ pixels for computational efficiency. We further describe GAE detection results obtained on PETS2009 S3 dataset². The patterns of PETS2009 contains various activities with a grouped multiple flow in 4 views, which could be used to test the performance of our visual representation derived from the hierarchical model. For the construction of the crowd prototypes and representations, we randomly select 300 C1 units with the size $5 \times 5 \times 4 \times 2$ patches in each 10-frame clip for both datasets. The prototypes contain 30 memberships. The sparse representation for each crowded clip are adopted by max pooling among the prototypes memberships as a histogram.

¹ UMN dataset, from http://mha.cs.umn.edu/movies/crowdactivity-all.avi.
Comparisons with the state-of-the-art: For UMN dataset, we utilize SVM with RBF kernels to train the model on the crowd representations of the clips in 10 videos and compute the FPR and TPR in the left one video. Each event is classified as normal or abnormal by the trained classifier. Figure 5(a) shows the performance of the proposed approach measured by the ROC curve compared with the state-of-the-arts (summarized by the AUC in Table 1(a)). Methods listed for comparison are directly obtained from [5, 18, 19, 3]. The results show that our crowd prototypes can achieve better performance over available state-of-the-arts including Interaction Energy Potentials [5] (IEP), Social Force [18] (SF), Streakline Potential [19] (SP) and Optical Flow (OF). Because our prototypes are capable of identifying coherent motion patterns, the proposed method is competitive for improving the performance on GAE. The slightly lower performance compared to Sparse Reconstruct Cost [3] is due to the optical flow, which is not as robust to illumination, distortion and noise as the MHOF [3].

Comparisons on visual representations: Here we compare our Crowd Prototype based sparse representation with other visual descriptors on PET-S dataset. To verify the discriminative capability of learned prototypes based representation, we alternate different descriptors while keeping the same max pooling strategy to quantize and generate the global feature vector. We train SVM on the selected 3 views and test on the rest one following the similar configuration as UMN dataset using a continuous threshold to generate ROC curves. Since our approach is partially biological inspired, we compare it with the biological motivated features GIST3D [22] and C2 feature [10]. GIST3D is closely related to the primary cortex. C2 features correspond the approach by Jhuang et al. with random sampling and template matching, and it is reported to be effective in action recognition. Figure 5(b) shows the ROC curves for the two scenes with the AUC measure shown in Table 1(b). This indicates that our crowd prototypes and C2 feature outperform the baseline optical flow and per-
form a little better than GIST3D. This shows the feedforward hierarchal model is more selective to motion changes than the GIST based method. Scene 1, which is more challenging than scene 2, contains gradual changes from normal to abnormal activity. Our method focuses on the issue of modeling the interaction and structure of the crowd motion pattern, which is more sensitive to changes in abnormality.

**Evaluation on prototype learning:** For a fair comparison of the learning part, we evaluate the prototype learning performance by fixing C1 unit responses with the same sampling strategy for all sparse coding based approaches on UMN dataset. Figure 6 reports the corresponding performance between basic Space Coding(SC) [14], Laplacian Sparse Coding(LapSC) [6] and Graph based Sparse Coding(GraphSC) [33] with varying level of prototype sizes. Our Crowd Prototypes without the 3rd term is very similar to GraphSC while our model without the last two terms corresponds to the basic SC. Notice that our model obtain obviously superior performance to that from SC, which indicates the coherent patterns are more representative for the crowd in UMN dataset. We can also discover that the performance of the structural term is also significantly promoted. This may be attributed to our model providing complementary perspectives to represent the interaction information. The prototype size does not show much impact on the detection accuracy. As above, it is found empirically to perform better when introducing both regularizations.

### 4.5 Tracking in Crowded Scene

In this experiment, we consider the application of tracking multiple individuals in a crowded scene using the proposed crowd prototype learning method to evaluate the overall efficiency as motion prior.

**Tracking Framework:** Because the high similarity between the targets and distracters as well as occlusion, crowd tracking is extremely challenging. Traditional single or multiple target tracking methods focus on how to extract discriminative appearance models, and pay little attention to how to model a target’s motion. They usually use a simple dynamic model with smooth motion and additive Gaussian noise to predict the location of the target:

$$x_{t+1} = x_t + v_t + n$$  \hspace{1cm} (10)
where \( \mathbf{v}_t \) is a 2D motion vector and \( \mathbf{n} \) is Gaussian noise. Typically, \( \mathbf{v}_t \) is computed using the state of the individual target at previous times, e.g., \( \mathbf{v}_t = \mathbf{x}_{t-1} - \mathbf{x}_{t-2} \). Due to random jitter in the predicted target location, the computed motion vector is usually not reliable. To improve tracking performance in a crowded scene, we incorporate the learning of prototypes as indicators of crowd motion for prediction and measurement in the motion model:

\[
\mathbf{x}_{t+1} = \mathbf{x}_t + \sum_{i} w(i, \mathbf{X}) \frac{V(i, \mathbf{X})}{\sum w(i, \mathbf{X})} \tag{11}
\]

where the keypoints by the gKLT tracker \( \mathbf{X} \in \Gamma \) are sampled in the candidate sampling region. \( w(i, \mathbf{X}) = \sum_{i \in \{1, \ldots, K\}} 1_{\mathbf{X} \in \text{Proto}_i} \) is the confidence score corresponding to the number of points that belongs to prototype \( i \). \( V(i, \mathbf{X}) \) is the average velocity of keypoints for prototype \( i \). We use our prototypes as a guide to reflect the likelihood of behaviors given the crowd prototypes, a process which is significantly consistent with the dynamic crowd motion.

**Evaluation on motion priors:** We compare the tracking method with and without the proposed crowd prototypes as well as state-of-the-art crowd tracking methods in three crowd sequences [1, 18]. We randomly sample the C1 unit for 5 frames to learn the prototypes with 8 prototype size and update the samples over time. Ground truth is obtained by manually labeling targets in each frame. We use the Average Position Error (APE) to compare tracking performance between different trackers. This measures the average position difference between all tracked objects and their corresponding ground truths. We implement a multiple target tracking algorithm in crowded scenes based on a real-time tracker [30]. The original tracker is for single object tracking and uses explosion search. We modified it using a particle filter framework with a simple state transition model (Eq. 10) as the baseline tracker. We compare the baseline tracker with the one which integrates our motion pattern in the state transition model (Eq. 11). Qualitative results are shown in Figure 8, from which we can see that integrating our prototype priors in the motion model significantly improves the tracking performance. The average position errors are shown in Figure 7 (Left), where the tracker with motion prior achieves the smaller tracking errors in all three sequences. We also compare with the recently proposed tracker [12, 21] for crowd tracking. The quantitative results are shown in Figure 7 (Right), which indicates our method using the prototype prior achieves the lowest errors or
Fig. 8. Tracking results on three sequences for comparison between the proposed approach (circles) vs. the approach by Zhang et al [30] which does not exploit motion priors (squares). The ground truth is shown with dots. Tracking results for different subjects are marked with different colors.

comparable results to the other approaches. As can be seen, the proposed crowd prototypes seems to capture perceptually more meaningful motion priors.

5 Conclusion

In this paper, we have described a new method to learn a dictionary of crowd prototypes with applications to crowd perception. We extend a biological model of motion processing [10] from the recognition of individual human activity to group behaviors. Motivated by human studies on crowd perception, our approach incorporates ensemble coding principles in addition to structural and local coherence constraints within a sparse coding approach for learning crowd prototypes. We have demonstrated the wide applicability of the approach to several problems in crowd perception. Experiments on public datasets demonstrate that the proposed model exhibits competitive performance against the state-of-the-arts.

References

27. Yamaguchi, K., Berg, A.C., Ortiz, L.E., Berg, T.L.: Who are you with and where are you going? In: CVPR (2011)