MULTI-LEVEL SCENE UNDERSTANDING VIA HIERARCHICAL CLASSIFICATION

Hamilton Scott Clouse *Xiao Bian* *Thanos Gentimis* *Hamid Krim*

Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC, USA

ABSTRACT

In applications where the use of video surveillance is necessary and/or beneficial, it is a common goal to identify the contents of the video automatically. Of particular interest in such applications is the ability to recognize locations in the environment, where events occur, and describe the events common to those locations. This is the goal of scene understanding.

Scene understanding is traditionally addressed from one of two separate points-of-view: the description of the underlying environment or the action taking-place throughout the scene. Each of these facets is required to address the overarching goal but, is insufficient independently to address the problem entirely. These facets are, in fact, dependent and by considering both, a more complete description becomes available. In this paper, we describe a novel, data-driven scene understanding and classification technique that captures and utilizes information about both the environment and activity within a scene.

***Index Terms*—** multilevel model, hierarchical classification, video processing, scene understanding, supervised learning

1. INTRODUCTION

In the work exposited herein, it is our goal to describe scenes of several agents, performing different actions, and then to use these descriptions in a supervised classification setting as training data to compare/classify query/test scenes.

Current techniques focus on describing either the environment or the motion/action with probability density functions (mixture models, etc.). Some of the efforts that endeavor to describe the environment utilize segmentation techniques to separate the constituents, e.g. [1]. However, many utilize a mulit-level approach to segment the constituents sequentially, e.g. [2] and [3], utilizing learned or defined distributions to classify different regions.

Works focusing on describing and/or classifying the actions in a scene are varied in approach. While some, e.g. [4], do utilize multi-level techniques for robustness, some, e.g. [5], also learn parameters to fit dynamical models while, still, others, e.g. [6], utilize statistical inference from graphical models.

While these techniques perform well and solve their intended problem, they do not address the problem of comparing scenes for both environment and activity. Only the description of one or the other (environment or motion/action) is available, not both. It is this problem we endeavor to address.

We approach solving this problem by leveraging a particular scene decomposition technique to describe the environment and the motion/action. We can then utilize the information from each sub-problem as additional information to simplify the other. The result is a two-fold description of the scene provided for quantized sub-regions. This layered, or multi-level description allows for hierarchical classification of subsequent query scenes for more thorough descriptions.

The remainder of this paper is organized as follows: Section 2 encompasses the description of our technique for scene understanding and its application, Section 3 details an experimental application of our method to some example data and we conclude in Section 4.

2. METHODS

We start with a video sequence represented by $\mathcal{X}(x, y, t)$ where $(x, y) \in \mathfrak{I}$ are the pixel coordinates in the image plane $\mathfrak{I} = [1, w] \times [1, h] \subset \mathbb{N}^+ \times \mathbb{N}^+$ with height and width indicated by h and w, respectively, and the frame index $t \in [t_i, t_f] \subset \mathbb{N}^+$. We partition it along its three dimensions, two spatial and one temporal. The temporal partitioning is done via the windowing that produces $X(x, y, t)$ with t constrained to the set $T = [t_1, t_n] \subset [t_i, t_f]$.

In this effort, the spatial partitioning remains constant throughout the time-window. That is, we assume the video sequence is produced via a stationary camera. We first segment the field-of-view (FOV) into regions that correspond to different activities within the scene. Then we provide two descriptions: one for the activity in each region and one for the regions themselves.

We first transform our video-cube X into a matrix. This is accomplished by vectorizing each frame $X_j = X(x, y, t_j)$

This work was funded via the ATR Center of AFRL's Sensor's Directorate. Special assistance provided by Dr. Wesam Sakla.

$\forall t_j \in T$ in the usual way: concatenating the columns to produce a vector

$$y_j = [X_j(x_1, y), X_j(x_2, y), \ldots, X_j(x_{w-1}, y), X_j(x_w, y)]^{tr} \tag{1}$$

These vectors are then arranged into a matrix

$$Y = [y_1, y_2, \ldots, y_{n-1}, y_n], \tag{2}$$

to represent the data. It is this matrix that we subsequently decompose using robust subspace recovery - dual sparsity pursuit (RoSuRe-DSP)[7]. The decomposition takes the form

$$Y = \mathcal{B}W + \mathcal{F}, \tag{3}$$

where the columns of $\mathcal{B}(s, t)$ are the low-rank basis vectors for the data, the background of the time-windowed video sequence, W is a coefficient matrix, and the columns of $\mathcal{F}(s, t)$ correspond to the sparse foreground components for each frame, the movers. Here, $s \in \mathcal{S}$ is the index of the pixels in each frame. Making use of the correspondence between values of s and pairs (x, y), we construct the functions $B(x, y, t)$ and $F(x, y, t)$ that are the background (low-rank) and foreground (sparse) components over the image plane $\mathcal{I}$.

2.1. The partitioning of the FOV

To ensure that the partitioning of the FOV corresponds to different activities, we consider all the motion contained in X to segment the regions. Since the decomposition has already separated the movement (F) from the stationary (B), we can perform our computations on just the moving part.

Define a summary function, $P(x, y) \in [0, 255]$, of all motion in X thus:

$$P(x, y) = \sum_{t \in T} F(x, y, t). \tag{4}$$

This image represents the sum of all foreground components throughout the time-windowed video sequence. It is possible to localize regions of activity by decomposing $P(x, y)$ into partitions U_l where $l \in \Lambda$ is an index set for the partitions, i.e. regions of interest (ROIs).

To aid in the segmentation of different regions of activity, we follow a procedure similar to [8], and define a threshold set

$$Q = \{q \in \mathfrak{I} | P(q) > \tau\}, \tag{5}$$

and an associated indicator function:

$$\mathcal{I}(q) = \begin{cases} 0, & q \in Q, \\ \infty, & \text{otherwise.} \end{cases} \tag{6}$$

The value for this threshold τ was chosen experimentally in this effort. This threshold allows the tuning of the amount of activity required to determine a ROI. Such additional formalism facilitates the computation of the traditional distance transform of the image $P(x, y)$, $\forall p \in \mathfrak{I}$ thus:

$$\mathcal{D}(p) = \min_{q \in \mathfrak{I}} \left(d(p, q) + \mathcal{I}(q) \right). \tag{7}$$

Local maxima of $\mathcal{D}(p)$ provide the boundaries of a partitioning of the set into subsets denoted U_l, our ROIs. These subsets can be projected onto the summary function $P(x, y)$, and thus the time-windowed set $X(x, y, t)$ as ROIs:

$$V_l = \{(x, y) \in \mathfrak{I} | \mathcal{D}(x, y) \in U_l\}, \forall t \in T \tag{8}$$

2.2. Description of the ROIs

To describe the environment of the scene observed in the time-windowed video sequence $X(x, y, t)$, we first separate the ROIs based on the decomposition described in §2.1. This is accomplished by considering the background of the video sequence over each region l:

$$B_l = \{B(x, y, t) | (x, y) \in V_l\}. \tag{9}$$

A feature set, f_l, for each background subset B_l is constructed following the SURF-128[9] algorithm. These descriptors are invariant under affine transformations and capture details about the constituent objects in a scene. A comparison between regions can be performed by finding and quantifying the number and strength of matches between respective descriptor feature sets.

This comparison is performed using the nearest neighbor ratio matching strategy[10]. We proceed by finding the nearest neighbors (c.f. k-nearest neighbors, k=2) to the features in the query set f_l from the training sets of descriptors. The distance (e.g. Euclidean distance), between these descriptors serves as a quantifier of the strength of the potential match. The region described by the candidate set of descriptors with the greatest number of matches to the test set is considered the most similar region.

2.3. Region Activity Description

The action description utilized in this effort is one of activity density. For each region V_l, we define the activity density:

$$\mathcal{A}_l(t) = \sum_{(x,y) \in V_l} F(x, y, t). \tag{10}$$

This results in the time-series $\mathcal{A}_l(t)$ for each ROI V_l that depicts the density of the activity in that region for he time-window T.

To eliminate the necessity of aligning these time-series for comparison, we consider the Fourier representation of these series,

$$A_l(t) = \mathscr{F}\left(\mathcal{A}_l(t)\right), \tag{11}$$

normalized for unit power, as the signature for the activity density in ROI l for time-window T. To discern/categorize these representations, the Fourier coefficients can be compared directly.

2.4. Training and Classification Framework

With these comparison tools defined, it is possible to proceed in a manner according to a supervised classification setting.

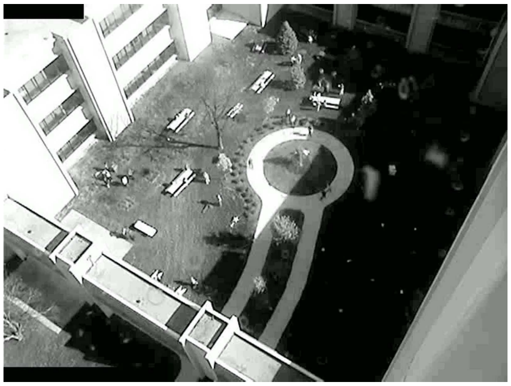

Fig. 1: Example frame of $X(x, y, t)$ showing courtyard from MSEE Data.

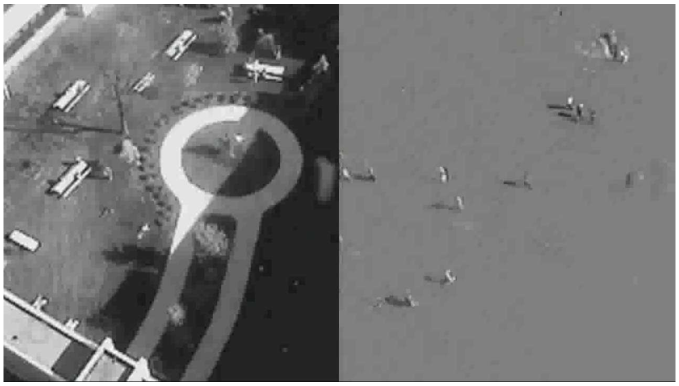

Fig. 2: Close-up of example frame showing results of (left) background, $B(x, y, t)$, and (right) foreground, $F(x, y, t)$, separation via RoSuRe-DSP.

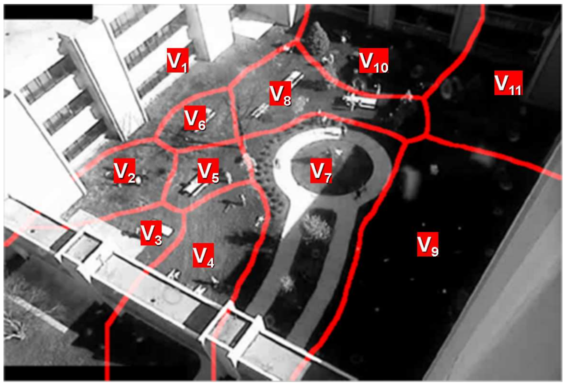

Fig. 3: Region boundaries produced from local maxima of $\mathcal{D}$

A training set is produced by performing the prescribed analysis on a large database of labelled scenes. The result will be descriptor sets and activity density signatures for each region segmented in each scene. The scenes are summarized by their constituent regions.

By supplying a sufficient database of scenes with accompanying descriptor feature sets and activity density signatures, it is possible to classify any incoming test/query scene by decomposing it into regions, according to this technique, and then comparing the descriptor sets and activity density signatures to those in the database. Once similar regions in the training set are identified, the query scene can be described as either a match to a training scene (all similar regions in one scene class), or a combination of regions from several training scenes. Due to the multiple levels of description and the fine granularity at the details, a comprehensive description for the query scene is generated based on the training set provided.

3. RESULTS AND DISCUSSION

The framework proposed in this paper was tested on a video sequence of a picnic scene, taking place in a courtyard, containing several different regions of activity including: regions where tents are being constructed, regions for playing games, regions containing picnic tables at which participants can eat and regions wherein no activity took place. The video sequence was captured from an overhead view at 10fps and a resolution of 640×480 pixels. An example image is shown in Figure 1.

A time-window length of 1.25min was chosen for these experiments to highlight some instructive results. The frames captured in the first such window were used for the test set $X(x, y, t)$.

3.1. Background/Foreground Separation

Applying RoSuRe-DSP to optimize for the form from Equation 3, the foreground $F(x, y, t)$ and background $B(x, y, t)$ of the video sequence were separated. An example is shown in Figure 2. The result of the decomposition illustrates complete separation of the foreground and background contents: the movers in the scene, e.g. people and their shadows, are extracted and even the "stationary" object, e.g. trees, that move slightly over throughout $X(x, y, t)$ are separated as background components. This accuracy in decomposition allows for the two-pronged approach of considering the motion elements and the background elements separately. However, such a breakdown also affords the possibility of leveraging the analysis of one component to address the other. Utilizing the motion elements to segment the background into ROIs is the next step.

3.2. The partitioning of the FOV

Proceeding with the partitioning, the summary function $P(x, y)$ was computed from the foreground component $F(x, y, t)$ as shown in Equation 4. The threshold for this data was set at $\tau = 150$ to produce the set Q described in Equation 5. It was this set that was utilized for the definition of the boundaries of the regions V_l, following Equations 7 and 8, which are enumerated in Figure 3.

3.3. Description of the ROIs

Once the regions V_l were segmented via their defined boundaries, descriptor sets were produced for each via the SURF-128[9] algorithm. As described in Section 2.2, utilizing these

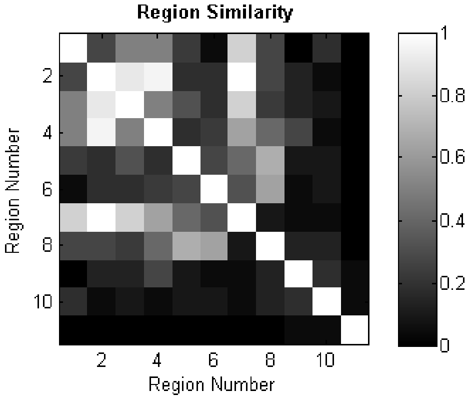

Fig. 4: Similarity between regions indicated by the number of matches, normalized.

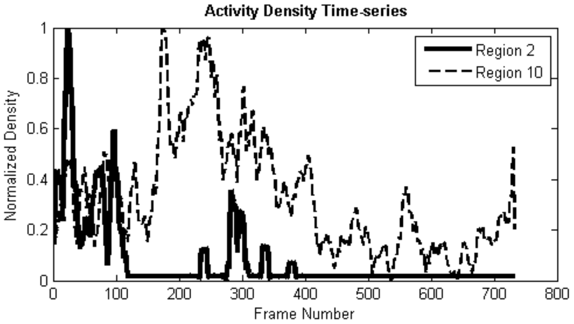

Fig. 5: Activity density time-series for regions V_2 and V_{10}, as labeled in Figure 3.

features, it's possible to compare the regions quantitatively by considering the number of matching descriptor pairs among all of the regions. An image representation of the similarity matrix so produced is shown in Figure 4. To aid in understanding, the number of matches have been normalized by the largest number of matching feature paris.

The region V_7 is quite similar to several regions due to the diversity of its contents: it contains part of a building and a large grassy area in addition to the sidewalk. These contents it shares with regions V_2, V_3 and V_4. It is less similar to regions V_5, V_6 and V_8 since these regions do not contain any portion of the building surrounding the courtyard but, do additionally contain picnic tables. Regions V_9, V_{10} and V_{11} are not significantly similar to other regions, as could be intuited from inspection.

3.4. Region Activity Description

The remaining portion of the procedure concerns the computation and comparison of the activity density signatures A_l. This was completed following the method described in Section 2.3. First, the activity densities $\mathcal{A}_l$ were computed from $F(x, y, t)$ for each ROI V_l as given by Equation 10. Examples of these time-series are shown in Figure 5.

These time-series accurately represent the intuitive understanding of the activity taking place in the corresponding regions. At the beginning of the scene, region V_2 is populated with individuals moving throughout. Soon, within the first 10s (100 frames), the crowd exits V_2 and the region remains mostly unpopulated for the majority of the remainder of the data with the exception of a few people passing-through near frame 300.

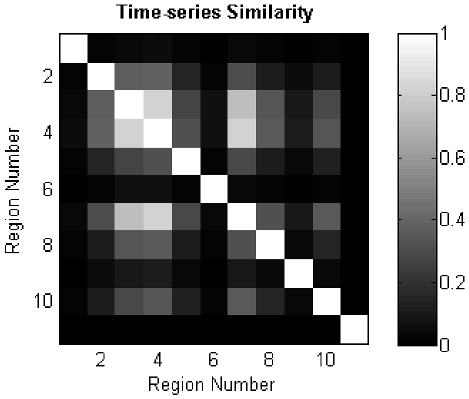

Fig. 6: Similarity between activity density signatures, normalized.

Region V_{10} is one of the more active regions in this data set in that several agents begin the scene by constructing a tent therein and exit the region once the task is complete. The activity density time-series indicated in Figure 5, again, accurately represents an intuitive understanding of the activity. A_{10} increases from the beginning of the scene to about frame 180 when the actors are entering the region and starting construction. The magnitude of A_{10} remains relatively high for many frames, corresponding to the ongoing construction of the tent. Once the construction nears completion, the activity density decreases as the agents exit V_{10}.

For comparison, we proceed to compute the activity density time signatures A_l as indicated in Equation 11. A simple correlation comparison of these signatures results in the similarity matrix shown in Figure 6. While there are some consistent similarity results between those of the region descriptors and the time-series, there are also dissimilarities, e.g. region V_1 is similar in descriptor feature set to region V_7 but, not so in time-series.

4. CONCLUSION AND FUTURE WORK

In this paper, we have described and demonstrated a novel data-driven scene understanding and classification technique. Current scene understanding approaches focus on either describing the environment capture within the scene or the action taking place during the scene. This technique addresses scene understanding as a single problem by leveraging the data available for each of these facets against the other and then combining these intertwined results into a single result. Continuing efforts are focused on testing the training/classification performance of this framework with more and varied data.

5. REFERENCES

[1] M. Pawan Kumar and Daphne Koller, “Efficiently selecting regions for scene understanding,” 2010.

[2] Li-Jia Li, Richard Socher, and Li Fei-Fei, “Towards total scene understanding: Classification, annotation and segmentation in an automatic framework,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2036–2043.

[3] Congcong Li, Adarsh Kowdle, Ashutosh Saxena, and Tsuhan Chen, “Towards holistic scene understanding: Feedback enabled cascaded classification models,” .

[4] Yang Yang, Jingen Liu, and Mubarak Shah, “Video scene understanding using multi-scale analysis.,” in *ICCV*. 2009, pp. 1669–1676, IEEE.

[5] I. Saleemi, L. Hartung, and M. Shah, “Scene understanding by statistical modeling of motion patterns,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 2069–2076.

[6] C. Town, “Ontology-driven bayesian networks for dynamic scene understanding,” in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW ’04. Conference on*, June 2004, pp. 116–116.

[7] Xiao Bian, Hamid Krim, Lucas Plaetevoet, and S Mariappan Nadar, “A comparative study of modern inference techniques for discrete energy minimization problems,” in *Submitted to Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014.

[8] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, “Distance transforms of sampled functions,” Tech. Rep., Cornell Computing and Information Science, 2004.

[9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, “Surf: Speeded up robust features,” in *In ECCV*, 2006, pp. 404–417.

[10] A. Baumberg, “Reliable feature matching across widely separated views,” in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 2000, vol. 1, pp. 774–781 vol.1.