Pattern Recognition Letters 44 (2014) 80–87

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Motion feature filtering for event detection in crowded scenes ☆

Lawrence O'Gorman *, Yafeng Yin, Tin Kam Ho

Alcatel-Lucent Bell Labs, Murray Hill, NJ, USA

ARTICLE INFO

Article history:
Available online 30 August 2013

Keywords:
Motion analysis
Event detection
Crowd analysis
Surveillance

ABSTRACT

We describe a spatio-temporal feature filtering approach that is appropriate for detecting video events in public scenes containing from many to few people. This non-discrete tracking – or pattern flow analysis – is distinguished by the fact that the usual video processing step of object segmentation is omitted; instead motion features alone are used to detect, follow, and separate activity. Motion features include location, scale, score (magnitude), direction, and velocity. The method entails gradient-based motion detection and multiscale motion feature calculation to obtain a scene activity vector. We focus on obtaining these motion features and filtering them to obtain information on activity, with the end-goal being event detection, classification, and anomaly detection. Examples of information extraction we show in this paper include: distinguishing anomalous from trend activity via shape of the activity profile over time, detecting event onset and direction of people flow using direction (and feature confidence) values, and measuring the periodicity of similar activity from magnitude values over time. We demonstrate utility of the approach on 3 video datasets: hallway, emergency event, and subway platform.

1. Introduction

It is generally accepted that video scene understanding becomes more difficult as the number of discretely moving objects increases. The general approach in video tracking research has been bottom-up in complexity, that is to detect, segment, and track one moving object, and then to attempt increasingly greater numbers until these overwhelm the ability to perform discrete segmentation and tracking. At some point, complexity also overwhelms the human visual system's ability to discretely track, so it is not unexpected that machine vision succumbs similarly.

Alternatively, a top-down complexity approach, one that tracks pattern flow versus discrete objects, is increasingly being followed for crowded scene analysis (Polana and Nelson, 1994; Zelnik-Manor and Irani, 2001; Zhong et al., 2004; Wang et al., 2009; Kratz and Nishino, 2009; Yang et al., 2009; Saleemi et al., 2010). Example applications include measuring the efficiency of passenger transfer in subway trains and determining pedestrian flows in malls and transportation terminals. Besides handling complex scenes, this approach can often extend down to single-object trajectories. This ability to yield similarly reliable results across a range of scene complexities is especially important to systems intended to function in an unsupervised manner. Of course, there is a tradeoff between discrete and non-discrete approaches, related to the presence or absence of information on individual objects, such as shape, size, and exact number.

Instead of tracking moving objects, our approach is to: (1) extract motion features – location, scale, score (magnitude), direction, and velocity; (2) filter the video stream by chosen feature(s) of interest (e.g., a certain location, a chosen scale of motion, a particular direction, a velocity range, or a combination of these), and (3) detect events by activity within chosen feature bounds, i.e., by motion feature filtering (O'Gorman et al., 2012).

One distinctive difference with respect to most previous work is choice and calculation of the motion features. Our features can be considered to be at a higher level than the common optical flow based motion features, because they are based upon spatio-temporal gradients across multiple frames versus frame-differences of intensity. We claim comparative robustness especially to lighting differences by using these features. A second distinction of this work is exclusive reliance upon feature filtering within our relatively rich set of motion features. Although feature filtering has been employed especially for crowded scene tracking, our approach builds upon previous methods with the use of higher level motion features. We make a third distinction to some previous work of similar application to ours, whose spatial and temporal precision is limited to relatively coarse quantization. We employ a multiscale representation and finer granularity in time and duration of event. We consider precision as especially important for event detection.

In Section 2, we describe previous work in motion feature extraction and crowd analysis and describe differences with our methods. In Section 3, we describe our approach, which entails

☆ This paper has been recommended for acceptance by Simone Calderara.

* Corresponding author. Tel.: +1 908 582 1783; fax: +1 908 582 3662.
E-mail addresses: Larry.OGorman@alcatel-lucent.com (L. O'Gorman), yafeng.yin1@gmail.com (Y. Yin), Tin.Ho@alcatel-lucent.com (T.K. Ho).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.08.020

three stages: motion feature calculation, scene activity vector determination, and event detection. Section 4 shows experimental results on three datasets showing how our features are used with respect to different distinguishing characteristics of the data.

2. Background

Motion extraction approaches can be classified under three categories: (1) background subtraction, (2) optical flow, and (3) spatio-temporal gradients or motion edges. Background subtraction methods usually employ a mixture of Gaussian (MOG) approach to store background intensity distribution models and perform subtraction of these from the current frame (Stauffer and Grimson, 1999; Friedman and Russell, 1997; Cheung and Kamath, 2004). Because a new background model takes time to differentiate from change due to true motion, these display error in scenes with variably changing lighting, such as outdoors. Optical flow methods detect motion as local movement of pixel intensities in time (Lucas and Kanade, 1981; Barron et al., 1994). In contrast to MOG methods, in which stable background models are built, the optical flow method detects any intensity change, whether due to motion, lighting, or variation in an otherwise uniform intensity region. In addition, the optical flow approach is relatively time-consuming, spending best-match search time within flat intensity regions where motion cannot be measured. In Yokoyama and Poggio (2005), optical flow computation expense is reduced by masking first by thresholded spatial edge magnitudes. The third category, spatio-temporal gradients, achieves a higher level of robustness with respect to lighting and intensity variation by first extracting spatial edges, and then determining their gradient over time (Kratz and Nishino, 2009; Gruenwedel et al., 2011). This is because spatial edges maintain relative consistency across lighting changes, and no edges exist within regions of uniform intensity, where there is no information to measure motion.

In Wang et al. (2009), crowded scene analysis is performed as follows. The video is quantized into 10×10 pixel cells and 10-s long, non-overlapped clips. Codewords for clips are histogram values derived from optical flow that is quantized to 4 directions. Hierarchical Bayesian models cluster atomic activities. Unusual activity is identified as distinct from these clusters.

In Kratz and Nishino (2009), gradient-based motion features are determined for non-overlapped cuboids (e.g., 30×30 pixels and 20 frames). Multivariate Gaussian modeling embodies information of dominant gradient direction and variance within cuboids. Both temporal and spatial relationships among cuboids are modeled using Hidden Markov models, and these enable the method to detect temporal or spatial deviations in activity within crowds of otherwise normal activity.

In Yang et al. (2009), optical flow features are found and 4-direction histograms used to describe non-overlapped cuboids (10×10 pixels times 5 s). Cuboids are concatenated to video clips, and contained histograms are considered words in a bag-of-words representation. Words are clustered by diffusion map embedding to find key motion patterns. In Saleemi et al. (2010) this approach is expanded beyond a single model per cuboid using k-means clustering and a Gaussian mixture model. By linking cuboid models across space and time, motion patterns are found, for example of cars going left, right, or straight at an intersection.

Our approach has similarities and differences from previous literature. Our motion extraction method falls under the category of spatio-temporal gradients, however it differs from previous methods in its determination of higher level motion features as will be described in Section 3.

Another difference involves temporal and spatial scale. We do not want to restrict ourselves to pre-chosen spatial sizes (full frames or sub-frame blocks) or temporal lengths (clips or cuboid lengths). Instead, we use a multiresolution spatial representation. This has similarities to the space–time interest point approach (Laptev, 2005) in that spatio-temporal regions of interest are found in multiscale space, however there are three important differences: (1) we determine region-of-interest – or activity – upon motion features rather than pixels, (2) instead of a regular Laplacian pyramid with Gaussian smoothing of pixels between levels, our relationship between levels is via another function (such as simple summation of feature values or dominant value, since Gaussian filtering is inappropriate for features versus pixel values), and (3) time is not treated in multiscale. Instead, we maintain full temporal resolution so event onsets and endings can be measured with highest precision.

A final difference is in the focus of this paper. Whereas most references cited here devote emphasis to their model and classification methodology, ours focuses more fundamentally at the lower processing level and in particular the choice and calculation of motion features, and filtering of these to distinguish different events types. We believe that attention toward improved motion features at the early processing stage reduces error propagation, and yields better final event detection.

3. Method

Our method entails three main steps: (1) motion feature calculation, (2) scene activity vector determination, and (3) event detection:

Step 1, *Motion Features* – Spatio-temporal gradients are found as time differences "Δ" of spatial differences,

$$G_t(x,y) = \frac{\Delta}{\Delta t}\left(\frac{\Delta I(x,y)}{\Delta x}, \frac{\Delta I(x,y)}{\Delta y}\right) \tag{1}$$

where $G_t(x,y)$ is spatio-temporal gradient at frame t and location (x,y), $I(x,y)$ is intensity, $\Delta I(x,y)/\Delta x$ is spatial edge in x (and similarly in y). The spatial edge can be found by a simple (minimal region) edge detector, such as Sobel, because larger region (or support) statistics come into account later. The time difference is between the current spatial edge image and an exponential moving average of that edge image.

The $G_t(x,y)$ images are thresholded with respect to a chosen value τ to obtain binary images of significant edges,

$$G'_t(x,y) = 1 \text{ if } G_t > \tau, 0 \text{ otherwise}. \tag{2}$$

We create what we term, *motion blur* images by combining the current thresholded gradient frame with monotonically decaying weights w_k of k previous frames as follows,

$$B_t(x,y) = \bigcup_{k=1}^{K} w_k G'_{t-k}(x,y) \tag{3}$$

$$w_k = W - k + 1, \quad 1 \leqslant k \leqslant K, \quad W \geqslant K \tag{4}$$

In (3), the notation indicates a "weighted logical OR", where the result is not 0 or 1, but w_k if $G'_{t-k} = 1$, or 0 otherwise. (If more than one G'_{t-k} is equal to 1, then the lowest value weight corresponding to the longest decayed edge is chosen.) Thus, a motion blur image contains a high value for edges of the current frame, and 1 less for the previous frame, etc., for K frames. This image looks like a single snapshot of edges of an object that moved causing blurring (see Fig. 1).

Linear regression fits are then applied in x and y to the average motion blur locations for each frame delay, within $w \times w$ windows around (x,y) locations of each motion blur frame. From the slope of the fits, ρ_x and ρ_y, the direction of motion θ is calculated,