Reading Notes

ORIGINAL

Heng Wang, Alexander Kla¨ser, Cordelia Schmid, Cheng-Lin Liu. National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences.

CATEGORIES

[**Computer Vision**]: Dense Trajectories, Action Recognition.

KEYWORDS

Computer Vision, Action Recognition, Dense Trajectories.

AUTHOR

Nino Lau, School of Data and Computer Science, Sun-Yet-Sen University.

1. BACKGROUND

Kanade–Lucas–Tomasi feature tracker

In computer vision, the Kanade–Lucas–Tomasi (KLT) feature tracker is an approach to feature extraction. KLT makes use of spatial intensity information to direct the search for the position that yields the best match. It is faster than traditional techniques for examining far fewer potential matches between the images.

The KLT feature tracker is based on two papers: In the first paper, Lucas and Kanade developed the idea of a local search using gradients weighted by an approximation to the second derivative of the image.

If **h** is the displacement between two images **F(x)** and **G(x)=F(x+h)**, then the iterative sequence of estimates will ideally converge to the best **h**.

For one-dimension case :

$$\begin{cases} h_0 = 0 \\ h_{k+1} = h_k + \dfrac{\sum_x \dfrac{w(x)\left[G(x) - F(x+h_k)\right]}{F'(x+h_k)}}{\sum_x w(x)} \end{cases}$$

For two-dimension case :

$$\begin{cases} h_0 = 0 \\ h_{k+1} = h_k + \dfrac{\sum_x w(x) F'(x+h_k)\left[G(x) - F(x+h_k)\right]}{\sum_x w(x) F'(x+h_k)^2} \end{cases}$$

Generalization to multiple dimensions :

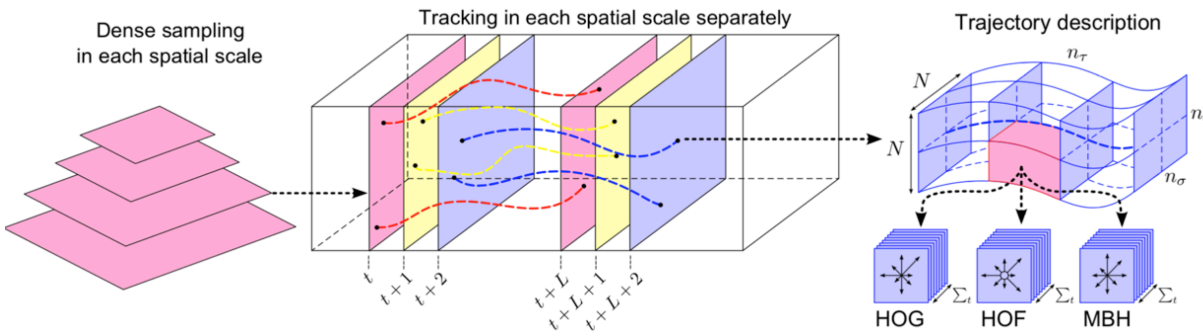
$$\begin{aligned} 0 &= \frac{\partial E}{\partial \mathbf{h}} \\ &\approx \frac{\partial}{\partial \mathbf{h}} \sum_{\mathbf{x}} \left[F(\mathbf{x}) + \mathbf{h} \left(\frac{\partial F}{\partial \mathbf{x}} \right)^T - G(\mathbf{x}) \right]^2 , \\ &= \sum_{\mathbf{x}} 2 \left[F(\mathbf{x}) + \mathbf{h} \left(\frac{\partial F}{\partial \mathbf{x}} \right)^T - G(\mathbf{x}) \right] \left(\frac{\partial F}{\partial \mathbf{x}} \right) \\ \Rightarrow \mathbf{h} &\approx \left[\sum_{\mathbf{x}} \left[G(\mathbf{x}) - F(\mathbf{x}) \right] \left(\frac{\partial F}{\partial \mathbf{x}} \right) \right] \left[\sum_{\mathbf{x}} \left(\frac{\partial F}{\partial \mathbf{x}} \right)^T \left(\frac{\partial F}{\partial \mathbf{x}} \right) \right]^{-1}, \end{aligned}$$

Existing Problems

Interest point are a popular way for representing videos. They achieve advanced results for action classification. However, consider the fact that the 2D-space domain and 1D-time domain share dramatically different characteristics. Therefore, it is intuitive to handle them in a different manner than via interest point detection in a joint 3D-space domain. The most straightforward choice is to track interest points through video sequences, where trajectories are represented as sequences of log-polar quantized velocities.

Dense sampling has shown to generate more accurate results over sparse interest points for image classification. Trajectories obtained by the KLT tracker mentioned above, are designed to track sparse interest points. Whereas, matching dense SIFT descriptors is computationally very expensive and, thus, impractical for large video datasets. This paper, the authors propose an efficient way to extract dense trajectories by tracking densely sampled points using optical flow fields, overcome the difficulties cause by camera motion and background by applying the foreground comparison method.

2. APPROACH



Method

Dense trajectories conduct by dense sampling are extracted for multiple spatial scales. Feature points are sampled on a grid spaced and tracked in each scale separately. For each 2D point of pixel at one of frames, it is tracked to the next frame by median filtering in a dense optical flow field. Drifting is one of the common problem in this process, we can get it over by limiting the length of a trajectory to L frames, and re-sampling after each epoch. After computed the dense optical flow field, we use the algorithm by Fa¨rneback (implemented in OpenCV) to extract optical flow. The shape of a trajectory encodes local motion patterns and we describe its shape by a sequence of displacement vectors. The resulting vector is normalized by the sum of the magnitudes of the displacement vectors.

To leverage the motion information in our dense trajectories, we compute descriptors, HOG (histograms of oriented gradients), HOF (histograms of optical flow), et al., within a space-time volume around the trajectory. Optical flow computes the absolute motion, which inevitably includes camera motion. MBH (motion boundary histogram) descriptor for human detection computes separately for the horizontal and vertical components of the optical flow. This descriptor encodes the relative motion between pixels. Since MBH represents the gradient of the optical flow, constant motion information is suppressed and only information about changes in the flow field (i.e., motion boundaries) is kept. Here we use MBH to describe our dense trajectories. For both HOF and MBH descriptors, we can reuse the dense optical flow that is already computed to extract dense trajectories, making our feature computation process very efficient.

3. EVALUATION

Datasets and Metric

The dense trajectories are extensively evaluated on four standard action datasets: KTH, YouTube, Hollywood2, and UCF sports. The KTH dataset views actions in front of a uniform background, whereas the Hollywood2 dataset contains real movies with significant background clutter. The YouTube videos are low quality, whereas UCF sport videos are high resolution. And the authors use a non-linear SVM with a χ^2- kernel to classify. In the case of multi-class classification, they use a one-against-rest approach and select the class with the highest score.

Experiment

The authors first make some comparisons among different descriptors, HOG, HOF and MBH, which confirms the advantage of suppressing background motion when dealing with optical flow. Then their dense trajectories and other state of art methods, showing its advancement.

By evaluating the different parameter settings for dense trajectories, they study the impact of the trajectory length, sampling step size, neighborhood size and cell grid structure. We evaluate the performance for a parameter at the time. The other parameters are fixed to the default values, i.e., trajectory length $L = 15$, sampling step size $W = 5$, neighborhood size $N = 32$ and cell grid structure $n\sigma = 2, n\tau = 3$.

4. COMMENTS

Advantages

This paper introduced an approach to model videos by combining dense sampling with feature tracking which are more robust than previous video descriptions. And introduced an efficient solution to remove camera motion by computing motion boundaries descriptors along the dense trajectories. Their descriptors combine trajectory shape, appearance, and motion information. Such a representation has shown to be efficient for action classification, but could also be used in other areas, such as action localization and video retrieval.

Optimization

An improvement of this DT method is iDT. It optimized the normalization of sum of magnitudes of the displacement vectors and generate a more accurate performance. Also iDT dismiss the bag of features, and utilize Fish Vector which is faster.