MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Keypoint trajectory coding on compact descriptor for video analysis

Tian, D.; Sun, H.; Vetro, A.

TR2016-127 August 2016

Abstract

In contrast to still image analysis, motion information offers a powerful means to analyze video. In particular, motion trajectories determined from keypoints have become very popular in recent years for a variety of video analysis tasks, including search, retrieval and classification. Additionally, cloudbased analysis of media content has been gaining momentum, so efficient communication of salient video information to perform the necessary analysis of video at the cloud server is needed. This paper describes a novel framework to efficiently represent the keypoint trajectories. In particular, an interframe prediction is designed with the option to operate in a low-delay mode. Additionally, a scalable coding method is proposed that allows for a subset of the coded trajectories in a video segment to be easily accessed. Experimental results on several popular datasets including Stanford MAR and Hopkin155 demonstrate a significant rate saving of up to 25% with our proposed trajectory coding approaches relative to a state-of-the-art reference approach.

IEEE International Conference on Image Processing (ICIP)

KEYPOINT TRAJECTORY CODING ON COMPACT DESCRIPTOR FOR VIDEO ANALYSIS

Dong Tian, Huifang Sun, Anthony Vetro

Mitsubishi Electric Research Labs (MERL)
Cambridge, Massachusetts, USA
{tian, hsun, avetro}@merl.com

ABSTRACT

In contrast to still image analysis, motion information offers a powerful means to analyze video. In particular, motion trajectories determined from keypoints have become very popular in recent years for a variety of video analysis tasks, including search, retrieval and classification. Additionally, cloud-based analysis of media content has been gaining momentum, so efficient communication of salient video information to perform the necessary analysis of video at the cloud server is needed. This paper describes a novel framework to efficiently represent the keypoint trajectories. In particular, an interframe prediction is designed with the option to operate in a low-delay mode. Additionally, a scalable coding method is proposed that allows for a subset of the coded trajectories in a video segment to be easily accessed. Experimental results on several popular datasets including Stanford MAR and Hopkin155 demonstrate a significant rate saving of up to 25% with our proposed trajectory coding approaches relative to a state-of-the-art reference approach.

Index Terms— Keypoint trajectory, video analysis, interframe prediction, scalable coding, CDVS, CDVA

1. INTRODUCTION

Many modern computer vision algorithms first identify keypoints in an image, and then extract features for those points that are invariant to translation, rotation, scaling and illumination. Examples of such features include scale-invariant feature transform (SIFT) [1], speeded-up robust features (SURF) [2], histogram of oriented gradients (HoG) [3] [4], etc. Accordingly, there has been much study on the coding and efficient transmission of such feature descriptors to enable visual analysis tasks at a server. *Compact Descriptor for Visual Analysis* (CDVS) is a standard that has been recently published by ISO/IEC in this field [5]. One limitation of this standard is that it only addresses the coding of descriptors associated with a single image frame.

For video analysis, the motion in a scene is very useful to facilitate applications such as object tracking, action recognition /detection [6], mobile robotics [7] and autonomous driving, as well as motion segmentation [8]. Since many of the image analysis schemes are based on keypoints, it is natural to consider the motion trajectories of these keypoints; the coding of these keypoint trajectories for the purpose of enabling efficient video analysis is the primary focus of this paper.

A naive method for video extracts descriptors from each image in the sequence, treating each image independently. Such a method fails to exploit the fact that features from successive video images tend to be very similar, and does not capture the motion trajectory information, resulting in a very redundant representation. Furthermore, such a method does not remove features that are not persistent from one image to the next, which may result in reduced reliability and accuracy. Thus, simply collecting individual image descriptors is not only inefficient in terms of rate, but is also expected to result in reduced performance.

Earlier work that was done in the context of MPEG-7 did define several types of motion descriptors, including those that described camera motion and motion activity for video segments, as well as motion trajectories for regions of the scene [9]. The motion trajectory descriptor was fairly high-level in that a single point was associated with objects of the scene. Since there are usually not that many objects in a scene, the representation of each trajectory did not need to be very efficient.

Another approach is to compress the descriptors derived from each image of the video, exploiting the motion of those descriptors through the video sequence. Such methods exploit an affine transform between neighboring pictures to reduce the bit-rate of the transmitted descriptors [10] [11]. However, those methods are limited in using a single affine transformation which may oversimplify the motion for many use cases, thereby reducing the accuracy of the analysis results.

Low-rank non-negative matrix factorization [12] exploits the near stationarity of salient object descriptors in the scene, which demonstrates the ability to perform object/scene clustering using a small percentage of visual descriptors from a large set extracted from each image of the video. Hence, for the frames following an image with all descriptors being fully coded, no significant bits are expected to code their corresponding visual descriptors. However, this approach does not provide a representation for motion over time, which is the major motivation of our work.

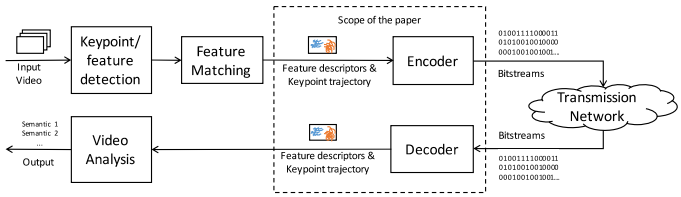

Fig. 1. System diagram

This paper proposes a novel framework to represent keypoint trajectories across pictures of a video to enable a good range of video analysis tasks that would benefit from motion information. Since keypoint trajectories are more dense than object trajectories, we study two approaches to efficiently compress the keypoint trajectories to align with the needs of different applications.

The remainder of the paper is organized as follows. The next section presents the overall system framework. Section 3 proposes an *interframe trajectory coding* approach (ITC) that is well-suited for low-delay applications, and a *scalable trajectory coding* method (STC) that provides a fine granular scalability in terms of keypoint trajectories is described in Section 4. Section 5 presents experimental results comparing the proposed methods against a homography model based benchmark method to show the superiority of the proposed methods. Finally, Section 6 provides concluding remarks.

2. SYSTEM OVERVIEW

Keypoint trajectories basically represent low-level motion information but they are considered useful for many high-level motion analysis task. To the best of our knowledge, there is no work done to efficiently compress the keypoint trajectories.

In a typical scenario as shown in Fig. 1, we propose to extract trajectory information for each keypoint and compress them at the client device (encoder side). We rely on existing methods to generate the keypoint trajectories. At the cloud server (decoder side), the keypoint trajectories are reconstructed then fed to a motion-based video analysis module to extract high-level motion descriptors or perform the desired video analysis tasks.

In an effort to build from the existing CDVS standardization framework, it is desirable for the descriptors to be used for video analysis to be compatible with the image descriptors and analysis systems. Hence, it is proposed that the feature descriptors associated with select pictures are coded in a manner that is compatible with the existing image analysis system; we will refer to such pictures as *key pictures*. We further assume that the video signal is split into *Group of Pictures* (GOP) structures for every n pictures.

More formally, a *keypoint trajectory* is represented by a sorted set of positions over time,

$$\{P(t_j) = (P_x(t_j), P_y(t_j)),\ t_j \in \{t_1, t_2, .., t_n\}\}. \quad (1)$$

Assume that there are m keypoints with associated trajectories. Let $P^c,\ c \in [1, m]$ denote the position in the c-th trajectory. In the earlier MPEG-7 work, there is only one representative point for each object, that is $m = 1$ for each single object in a picture. However, in this work, we typically consider the case with $m \gg 1$ for an object.

As the trajectory represents the travel path of a keypoint, the associated feature descriptors of the same keypoint in the subsequent picture are assumed to be unchanged and would be skipped for actual coding. Therefore, the main focus is on coding $\{P^c(t_j),\ c \in [1, m],\ j \in [2, n]\}$, given picture t_1 as a reference key picture that has been previously coded. In addition, let $\hat{P}$ denote the reconstructed trajectory from a coded bitstream.

3. INTERFRAME TRAJECTORY CODING

This section describes our proposed *interframe trajectory coding* (ITC) method, in which all keypoints in a picture will be coded before any keypoint from another picture. We describe both uni-directional and bi-directional prediction of the trajectories. For the case that the picture coding order is exactly the same as the picture presentation (capture) order, the delay to start coding would only be 1-frame, which is useful for real-time applications; this special case is referred to as *low-delay interframe trajectory coding* (LD-ITC). In contrast, with *scalable trajectory coding* (STC) to be presented in Section 4, such a low-delay mode of operation is not possible and the delay would typically be equal to the period of a full GOP.

3.1. Uni-Prediction for Trajectory Coding

With uni-prediction, the keypoints from a single reference picture are used to predict the keypoints in the current picture. Assume $P^c(t_j),\ c \in [1, m]$ in picture $t_j, j \geq 2$ is the current keypoint to be coded, and a previously coded picture $t_f, f \geq 1$ is used as a reference. A 3-step prediction using a derived *motion vector* (MV) is shown in the center of Fig. 2.

Firstly, among the keypoints that have been coded in the current picture t_j, we determine a reference keypoint r that is nearest to the current keypoint c using a distance $|\hat{P}^r(t_f) - \hat{P}^c(t_f)|$ that is measured in the reference picture t_f.

Then, a motion vector for keypoint r is calculated from picture t_f to picture t_j,

$$V^f = \hat{P}^r(t_j) - \hat{P}^r(t_f). \quad (2)$$

Next, the predicted keypoint c is given based on the motion vector in Eqn. (2),

$$P'(t_j) = \hat{P}^c(t_f) + V^f. \quad (3)$$

Finally the residual to be coded is,

$$R^c(t_j) = P^c(t_j) - P'(t_j). \quad (4)$$

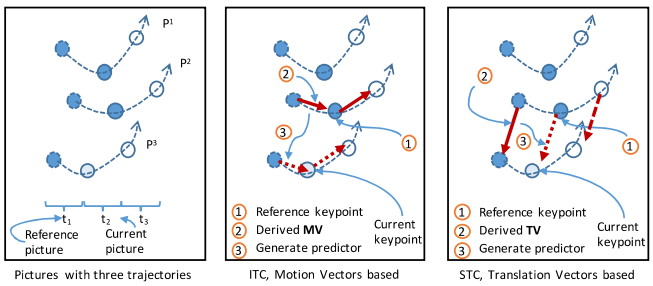

Fig. 2. Proposed trajectory coding methods

When a reference keypoint r for the current trajectory c is not present, the principle in Section 4.1 is applied, where the motion vector is set to be zero, $V^f = 0$. If the residual R^c obtained in Eqn. (4) is small enough to be ignored, it may be desirable to skip the residual coding.

With this prediction and coding scheme, the reference picture t_f could be a picture appearing before or after the current picture t_j. If we force $t_f < t_c$ to hold all the time, a low-delay configuration could be implemented.

3.2. Bi-Prediction for Trajectory Coding

In this section we consider using two reference pictures simultaneously as a bi-prediction coding mode to achieve greater rate savings. In addition to a first reference picture t_f, a second reference picture t_b will be selected to predict the keypoints in the current picture t_j. For example, t_f is selected as the nearest picture from the past or a forward reference picture, while t_b is selected as the nearest picture from the future or a backward reference picture.

As with uni-prediction, a reference keypoint r needs to first be determined among the keypoints that have already been coded in the current picture t_j. Two motion vectors corresponding to the forward and backward references are calculated as,

$$V^f = \hat{P}^r(t_j) - \hat{P}^r(t_f), V^b = \hat{P}^r(t_j) - \hat{P}^r(t_b). \quad (5)$$

Then two intermediate predicted keypoints for c are given,

$$P'(t_f, t_j) = \hat{P}^c(t_f) + V^f, P'(t_b, t_j) = \hat{P}^c(t_b) + V^b. \quad (6)$$

The final keypoint predictor c is given according to a weighted average based on the distances between reference pictures and the current picture,

$$P'(t_j) = \frac{|t_b - t_j|P'(t_f, t_j) + |t_f - t_j|P'(t_b, t_j)}{|t_f - t_j| + |t_b - t_j|}. \quad (7)$$

Finally, the residual is calculated as in Eqn. (4), which will be coded or possibly skipped in the bitstream. Note that introducing bi-prediction may break the low-delay configuration if one of the reference pictures is from the future.

4. SCALABLE TRAJECTORY CODING

Compared with the ITC approach presented in the previous section, where all keypoints in a picture are coded before start coding any keypoint in a next picture, we propose a *Scalable Trajectory Coding* (STC) scheme, in which all keypoints corresponding to a trajectory are coded before coding any keypoint from another trajectory. This approach permits fine granular scalability with respect to keypoint trajectories. That is, a decoder is able to recover some complete trajectories from a partially received bitstream making it is possible to begin an analysis of the video with a subset of the keypoint trajectories and even discard the remaining bitstream if the preliminary results are satisfactory. Of course, the server may refine its results in a fine granular manner when more trajectories become available. With the ITC approach described in section 3, however, the decoder has to decode all pictures before a complete keypoint trajectory could be obtained.

4.1. Intra-Scalable Trajectory Coding

We first describe an *Intra-Scalable Trajectory coding* (Intra-STC) scheme, whereby a trajectory is coded independently from others. Assume that we will code the c-th trajectory, $P^c = \{P^c(t_j),\ t_j \in [t_2, t_3, .., t_n]\}$ with reconstructed position $\hat{P}(t_1)$ from the available key picture. An earlier keypoint is used to predict a newer keypoint within the same trajectory. Let $\{P'(t_j), t_j \in [t_2, t_3, .., t_n]\}$ denote the predictors. A first order predictor is given in Eqn. (8),

$$P'(t_j) = \hat{P}(t_{j-1}) + v(t_j - t_{j-1}), \quad (8)$$

where $v = \frac{P(t_n) - P(t_1)}{t_n - t_1}$ is the keypoint velocity. We may use a second order predictor given the keypoint's initial velocity and its acceleration for more accurate prediction.

The prediction parameters v may be signaled at the sequence/GOP level or for trajectories in a local area. To simplify, we could select a predefined value for v, e.g. let $v = 0$, then the residuals $R^c(t_j)$ are simply the position differences of a keypoint between neighboring pictures.

4.2. Inter-Scalable Trajectory Coding

In order to better exploit redundancies between trajectories, we propose *Inter-Scalable Trajectory Coding* (Inter-STC) mode, which has the capability to predict the current trajectory from a reference trajectory (see right figure of Fig. 2). The trajectories coded in Inter-STC mode may use Intra-STC trajectories or other Inter-STC trajectories as references.

A similar 3-step procedure as used in the ITC scheme is applied for Inter-STC. However, instead of deriving a motion vector (MV) between neighboring pictures, a translation vector (TV) is derived from the selected reference trajectory r to the current trajectory within a reference picture t_f,

$$V^t = \hat{P}^c(t_f) - \hat{P}^r(t_f), \quad (9)$$

which will be used to generate a predictor,

$$P'(t_j) = \hat{P}^r(t_j) + V^t. \quad (10)$$

In Inter-STC approach, the determination of a reference trajectory only needs to be done once for all keypoints within one trajectory. However, for ITC approaches , due to the keypoint coding order, it either has to repeat the finding of a reference trajectory for each keypoint, or keep the reference trajectory in memory.

Note that the above TV-based Inter-STC may be modified to use the motion vector in Eqn. (2) while keep using STC coding order for the keypoints. The coding performance of the MV-based Inter-STC is equivalent to ITC with uni-prediction, so in our experiment, we will use TV-based Inter-STC when evaluating the STC method.

5. EXPERIMENTS

5.1. Experiment Set-up

In order to evaluate the proposed keypoint trajectory coding approaches, we chose several dataset popular in different video analysis projects. Results for four sample sequences are reported. If sorted according to scene/motion complexity from low to high, the four sequences are, $Seq1$ from Stanford MAR dataset [11] demonstrates some slow motion; $Seq2$ from KITTI dataset [7] is captured by a camera mounted on a moving car; $Seq3$ in [8] is also captured from a car and it was driven on an uneven road; and $Seq4$ from Hopkin155 dataset [13] is acquired by a hand-held camera having multiple objects with different motions plus a camera motion.

First, the trajectories are extracted based on Wang's method in [6]. Then 300 trajectories evenly distributed over each picture are selected for the coding experiments. The GOP size is set to 15 pictures. Keypoint trajectories of the 4 example sequences are shown in Fig. 3.

We implemented a benchmark approach based on homography model prediction, which is motivated by the work in [10] [11]. When finding the motion model parameters, the positions of all input keypoints from two pictures are utilized to estimate the homography model. In particular, the affine transform is selected for the homography structure. Once the affine transform is estimated, we use it to project keypoint positions in the current picture based on their locations in the previous picture. It fits for a low-delay configuration as ITC.

For the ITC scheme, we apply uni-prediction described in Section 3.1 to realize a low-delay configuration. For STC, we use the translation vector based Inter-STC in Section 4.2, since we found that the performance of the motion vector based STC is equivalent to the ITC.

5.2. Observations and Discussions

Since we focus on predictive coding of the keypoint trajectories and do not introduce any loss in the coding, we report the zero entropy of the residual trajectories generated by each approach in Table 1 as a measure of performance. The results of any analysis algorithm that uses motion trajectories would be unchanged by the coding.

Fig. 3. Keypoint trajectories of example video sequences, from left to right: $Seq1$ .. $Seq4$

-	$Seq1$	$Seq2$	$Seq3$	$Seq4$
Homography	1660	1879	1719	1973
ITC, MV-based	1529	1597	1417	1462
STC, TV-based	1541	1663	1455	1499
Bits saving (%)	**7.89**	**15.01**	**17.57**	**25.90**

Table 1. Entropy of residuals, in bits/picture

From Table 1, for the homography model based benchmark method, the bits required are increased significantly with increases in scene/motion complexity from $Seq1$ to $Seq4$. On the other hand, the proposed ITC and STC methods do not necessarily require more bits with increased scene/motion complexity. However, our approaches are more sensitive to the noise level in the trajectories. For $Seq1$ and $Seq2$ with less texture and thus more noise in the trajectory, our approaches consumes more bits than the other two sequences. Based on this observation, it might be desirable to filter the trajectories prior to coding.

In Table 1, the bits saving percentage is calculated between the MV-based ITC and the benchmark. Examining the bits used for the same sequence, it is clear that proposed methods need much less bits to code the keypoint trajectories. Additionally, the proposed methods have more significant bits saving for sequences with more complex scene/motion, which is up to about 25% bits saving for the most complex sequence.

6. CONCLUSIONS AND FUTURE WORK

This paper describes novel methods to code keypoint trajectories of videos for the purpose of analysis. Inter-frame prediction of the keypoint trajectories was proposed to efficiently compress the trajectories. This coding scheme is able to operate with low delay. Additionally, a scalable trajectory coding method has been presented, which provides fine granular scalability to the set of keypoint trajectories. Experimental results demonstrate that the proposed methods significantly outperform a conventional method that uses a homography model. Future work will consider alternative ways to generate or filter the trajectories prior to coding in order to suppress noise, with the aim to optimize rate and analysis performance.

7. REFERENCES

[1] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *Computer vision–ECCV 2006*, pp. 404–417. Springer, 2006.

[3] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 886–893.

[4] Pedro Felzenszwalb, David McAllester, and Deva Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[5] ISO/IEC JTC 1/SC29, "Information technology - multimedia content description interface - part 13: Compact descriptors for visual search," MPEG, N14956, 2014.

[6] Heng Wang, Dan Oneata, Jakob Verbeek, and Cordelia Schmid, "A robust and efficient video representation for action recognition," *International Journal of Computer Vision*, pp. 1–20, 2015.

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[8] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Computer Vision–ECCV 2008*, pp. 44–57. Springer, 2008.

[9] Sylvie Jeannin and Ajay Divakaran, "MPEG-7 visual motion descriptors," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 720–724, Jun 2001.

[10] Zhangshuai Huang, Ling-Yu Duan, Jie Lin, Shiqi Wang, Siwei Ma, and Tiejun Huang, "An efficient coding framework for compact descriptors extracted from video sequence," in *Image Processing (ICIP), 2015 IEEE International Conference on*, Sept 2015, pp. 3822–3826.

[11] Mina Makar, Vijay Chandrasekhar, Shauhyuarn Sean Tsai, David Chen, and Bernd Girod, "Interframe coding of feature descriptors for mobile augmented reality," *Image Processing, IEEE Transactions on*, vol. 23, no. 8, pp. 3352–3367, 2014.

[12] Hassan Mansour, Shantanu Rane, Petros T Boufounos, and Anthony Vetro, "Video querying via compact descriptors of visually salient objects," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2789–2793.

[13] Roberto Tron and Rene Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.