Key-Point Sequence Lossless Compression for Intelligent Video Analysis

Weiyao Lin
Shanghai Jiao Tong University

Xiaoyi He
Shanghai Jiao Tong University

Wenrui Dai
Shanghai Jiao Tong University

John See
Multimedia University

Tushar Shinde
Indian Institute of Technology Jodhpur

Hongkai Xiong
Shanghai Jiao Tong University

Lingyu Duan
Peking University

***Abstract*—Feature coding has been recently considered to facilitate intelligent video analysis for urban computing. Instead of raw videos, extracted features in the front-end are encoded and transmitted to the back-end for further processing. In this article, we present a lossless key-point sequence compression approach for efficient feature coding. The essence of this predict-and-encode strategy is to eliminate the spatial and temporal redundancies of key points in videos. Multiple prediction modes with an adaptive mode selection method are proposed to handle key-point sequences with various structures and motion. Experimental results validate the effectiveness of the proposed scheme on four types of widely used key-point sequences in video analysis.**

■ **Intelligent video analysis,** involving applications such as activity recognition, face recognition, and vehicle reidentification, has become part and parcel of smart cities and urban computing. Recently, deep learning techniques have been adopted to improve the capabilities of urban video analysis and understanding by leveraging on large amounts of video data. With widespread deployment of surveillance systems in urban areas, massive amounts of video data are captured daily from front-end cameras.

Digital Object Identifier 10.1109/MMUL.2020.2990863
Date of publication 28 April 2020; date of current version 28 August 2020.

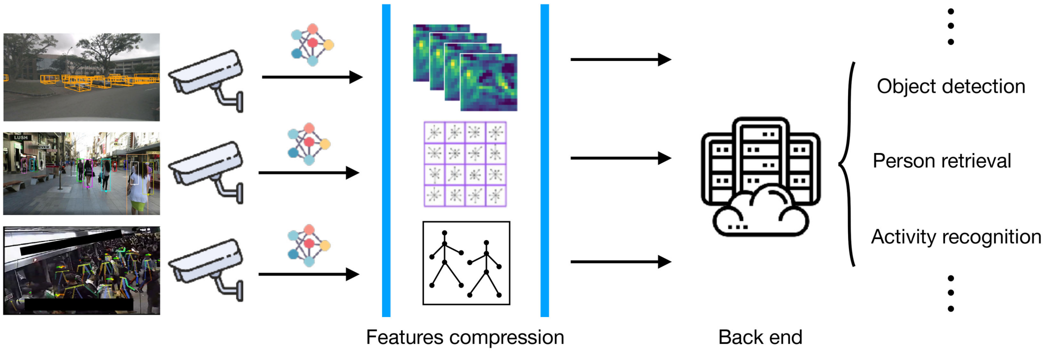


Figure 1. Illustration of the feature compression and transmission framework. Best viewed in color.

However, it remains a challenging task to transmit the large-scale data to the back-end server for analysis, although the state-of-the-art high-efficiency video coding (HEVC)[1] and on-going versatile video coding (VVC) standards present reasonably efficient solutions. An alternative strategy that transmits the extracted and compressed compact features, rather than entire video streams, from the front-end to the back-end, is illustrated in Figure 1. These *feature streams*, when passed to the back-end, enable various video analysis tasks to be achieved efficiently. Here, we summarize the advantages of transmitting information via feature coding in a lossless fashion: 1) Lossy video coding would affect the fidelity of reconstructed videos and subsequent feature extraction at the back-end, which leads to degraded accuracy in video analysis tasks; 2) Transmitting features rather than videos can mitigate privacy concerns to sensitive scenes such as in hospitals and prisons; 3) Computational balance can be struck between the front-end and back-end processing, as decoded features are directly utilized for the analysis in the back-end.

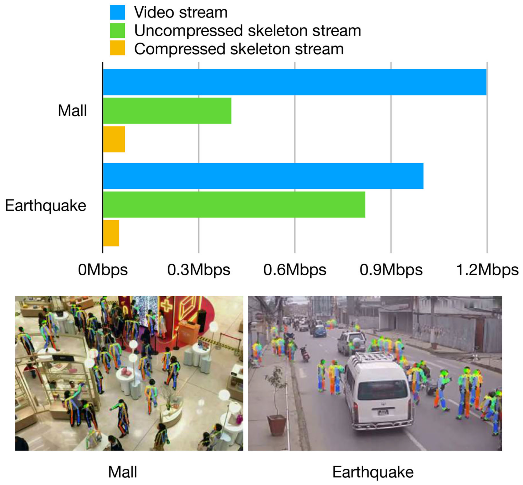


Figure 2. Two typical surveillance video sequences along with uncompressed and compressed skeleton streams. Best viewed in color.

In video analysis, common features include hand-crafted features (e.g., LoG and SIFT descriptors), deep features, and other contextual information (e.g., segmentation information, human and vehicle bounding boxes, facial and body pose landmarks). Among these features, key-point sequence is one of the most widely used type of feature. Key-point information like facial landmarks, human body key-points, bounding boxes of objects, and region-of-interests (ROIs) for videos are essential for many applications, e.g., face recognition, activity recognition, abnormal event detection, and ROI-based video transcoding.

Key-point sequences consist of the coordinates of key points in each frame and the corresponding tracking IDs. With the advances in multimedia systems, such semantic data become nonnegligible for complex surveillance scenes with a large number of objects. Figure 2 shows that uncompressed skeleton streams still take up a costly portion of typical video streams. Therefore, there is an urgency to compress these sequences effectively.

In this article, we propose a new framework for lossless compression of key-point sequences in surveillance videos to eliminate their spatial and temporal redundancies. The spatial redundancy is caused by correlations of spatial positions, while the temporal redundancy arose from the significant similarities between the positions of object key-points in consecutive frames. The proposed framework as a proposal for key-point compression has been accepted by the vision feature coding group of AITISA* as coding standard for vision features.

We start with a brief review of the feature representation for video analysis, particularly on how key-point information is extracted from videos to generate key-point sequences. Consequently, we propose a lossless compression framework for key-point sequences with adaptive selection of prediction modes to minimize spatial and temporal redundancies. Finally, we present experimental results to showcase the strengths of the proposed framework on various key-point sequences.

FEATURE REPRESENTATION IN EVENT ANALYSIS

In this section, we discuss several widely used feature representations for event analysis.

Digital Video

As a prevailing representation of video signals, digital videos consist of multiple frames of pixels with three color components. Digital video contents can be shown on mobile devices, desktop computers, and television. Compression of digital videos has been well addressed in various studies. The high-efficiency video coding (HEVC) standard improves conventional hybrid frameworks like MPEG-2 and H.264/AVC to yield quasi-equivalent visual quality with significantly reduced bit-rates, e.g., 50% bitrate saving in comparison to H.264/AVC. Recently, the on-going VVC standard is expected to further improve HEVC.

Feature Map

Generally, feature maps (in the form of 4-D tensors) are the output of applying filters to the preceding layer in neural networks. In recent years, deep convolutional neural networks have been utilized to extract deep features for video analysis. These features can be transmitted and deployed to accomplish analysis on the server side. Recently, there has been increasing interest in the compression of deep feature maps. For example, Choi and Bajic[2] employed HEVC to compress quantized 8-bit feature maps.

3D Point Cloud

3D point clouds are popular means of directly representing 3D objects and scenes in applications such as VR/AR, autonomous driving, and intelligent transportation systems. They are composed of a set of 3D coordinates and attributes (e.g., colors and normal vectors) for data points in space. However, communication of point clouds is challenging due to their huge volume of data, which necessitates effective compression techniques. As such, MPEG is finalizing the standard for point cloud compression that includes both lossless and lossy compression methods. Typically, the coordinates and attributes of point clouds are compressed separately. Coordinates are decomposed into structures such as octrees[3] for quantization and encoding. When preprocessed with *k*-dimensional (*k*-d) tree and level of details description, attributes are compressed with similar encoding process (prediction, transform, quantization, and entropy coding) as traditional image and video coding.

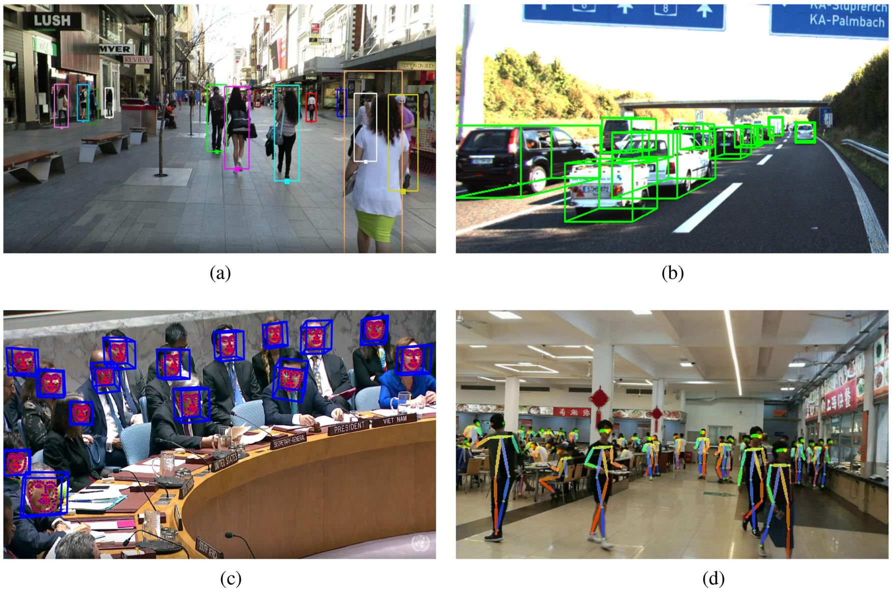
Key-Point Sequence

Various key-point sequences have been considered to improve video representation for urban video analysis. However, costs for transmission and processing are significant as there exist no efficient compression algorithms for key-point sequences.

2D Bounding Box Sequence: A 2D bounding box sequence is a sequence of 2D boxes over time for an object, as shown in Figure 3(a). A 2D box can be represented by two (diagonal or anti-diagonal) key points. Multiple sequences of 2D boxes can be combined to depict the motion variations and interactions between objects in a scene. As such, these sequences are suitable for human counting, intelligent transportation, and autonomous driving.

2D bounding box sequences can be obtained based on object detection[4,5] and tracking.[6] 2D object detection methods can be classified into anchor free[5] and anchor based[4] methods. Object tracking[6] can be viewed as bounding box matching, as it is commonly realized based on a tracking-by-detection strategy. Furthermore, the *MOT*

[*] http://www.aitisa.org.cn/



(a)

(b)

(c)

(d)

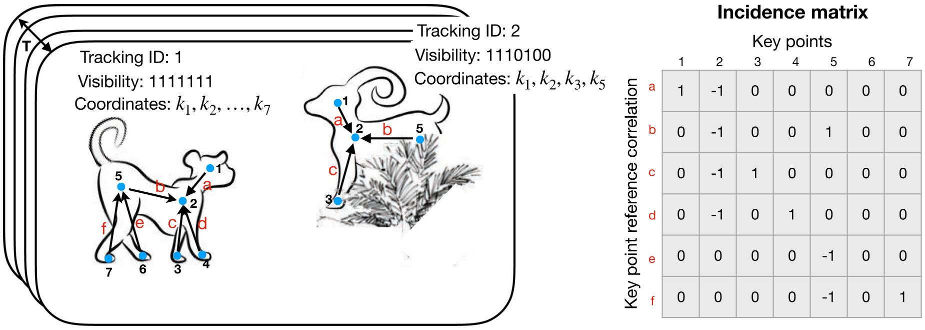
Figure 3. Examples of different key points. Best viewed in color. (a) 2D bounding boxes. (b) 3D bounding boxes. (c) Facial landmarks. (d) Skeletons.

Challenge[7] provides a standard benchmark for multiple-object tracking to facilitate the detection and tracking of dense crowds in videos.

3D Bounding Box Sequence: Similar to the 2D case, a 3D bounding box sequence is a sequence of 3D boxes of an object over time. Compared with 2D, 3D bounding boxes offer the size and position of objects in real-world coordinates to perceive their poses and reveal occlusion. A 3D bounding box shown in Figure 3(b) consists of eight points and can be represented by five parameters. Since an autonomous vehicle requires an accurate perception of its surrounding environment, 3D box sequences are fundamental to autonomous driving systems. A 3D bounding box sequence can be obtained by 3D object detection and tracking methods. 3D object detection can be realized with monocular image, point cloud, and fusion-based methods.[8,9] Monocular image based methods mainly utilize single RGB images to predict 3D bounding box, but this limits the detection accuracy. Fusion-based methods fuse the front-view images and point clouds for robust detection. Tracking with 3D bounding boxes[10] is similar to 2D object tracking, except that modeling object attributes (i.e., motion and appearance) is performed in 3D space. However, uncompressed 3D bounding box sequences are infeasible for transmission.

Skeleton Sequence of Human Bodies: Skeleton sequences can address various problems, including action recognition, person reidentification, human counting, abnormal event detection, and surveillance analysis. In general, a skeleton sequence of a human body consists of 15 body joints [shown in Figure 3(d)], which provides camera view-invariant and rich information about human kinematics. Skeleton sequences of human bodies can be obtained by pose estimation and tracking. OpenPose[11] is the first real-time multiperson 2D pose estimation approach that achieved high accuracy and real-time performance. AlphaPose[12] further presents an improved online pose tracker. *PoseTrack*[13] is proposed as a large-scale benchmark for video-based human pose estimation and tracking, where data-driven approaches have been developed to benefit skeleton-based video analysis.

Facial Landmark Sequence: Facial landmark sequence, which consists of facial key-points of a human face in video, is widely used in video-based facial behavior analysis. Figure 3(c) provides an example of facial landmarks, where 68 key-points are annotated for each human face. The dynamic motions in facial landmark sequences can produce accurate temporal representations of faces. Studies in facial landmark detection range from traditional generative models[14] to deep neural network based methods.[15] In addition, facial landmark tracking has been



Incidence matrix

Key point reference correlation	Key points 1	2	3	4	5	6	7
a	1	-1	0	0	0	0	0
b	0	-1	0	0	1	0	0
c	0	-1	1	0	0	0	0
d	0	-1	0	1	0	0	0
e	0	0	0	0	-1	0	0
f	0	0	0	0	-1	0	1

Figure 4. Example of an arbitrary form of key-point sequence in a frame and corresponding incidence matrix. Vertices (key points) are annotated with numbers 1–7 and edges are annotated with letters a-f. Best viewed in color.

well studied under constrained and unconstrained conditions.[16,17]

REPRESENTATION OF KEY-POINT SEQUENCES

Descriptor

To encode key-point sequences, we propose to represent key-point information in videos with four components: key point coordinate, incidence matrix of key points, tracking ID, and visibility indicator, as shown in Figure 4.

Key Point Coordinate: Key points of each object is expressed as a set of N coordinates of D dimensions (e.g., 2D and 3D):

$$K = \{k_1, k_2, \ldots, k_N\}, \tag{1}$$

where $k_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ with p_{ij} the coordinate in the jth dimension for the ith point.

Incidence Matrix: The encoded points can be used as references to predict the coordinates of current point. To efficiently reduce redundancies, an incidence matrix is introduced to define the references to key points. Thus, the key points of an object can be viewed as vertices of a directed graph. An edge directed from point 1 to point 2 indicates that 1 can be a reference point of 2. Given a key point, one of its adjacent vertices indicated by the incidence matrix are selected for prediction and compression. This suggests that efficient prediction and compression can be achieved by selecting adjacent vertices with higher correlations as references.

Tracking ID: Each object is assigned with a tracking ID when it first appears in the video sequence. Note that tracking ID for the same object does not change within the sequence and new objects are assigned a new tracking ID in increasing arithmetic order.

Visibility Indicator: Occlusion tends to appear in dense scenarios. This is commonly due to overlapping movements of different objects, and movements in and out of camera view. Similar to PoseTrack[13] annotations, we introduce a one-bit flag for each key point to indicate whether it is occluded.

$$V = \{v_1, v_2, \ldots, v_N\}, \qquad v_i \in \{0, 1\}. \tag{2}$$

LOSSLESS COMPRESSION FOR KEY-POINT SEQUENCES

Framework

Figure 5 illustrates the proposed framework for lossless key-point sequence compression based on the key-point sequence descriptor. Here, we consider to encode the key point coordinates, tracking IDs, and visibility indicators, as predefined incidence matrices are provided in both encoder and decoder for specific key-point sequences, e.g., facial key points, bounding boxes, and skeleton key joints. Similar to H.264/AVC and HEVC, we adopt exponential-Golomb coding to encode prediction residuals.

In this section, four different prediction modes with adaptive mode selection are developed for key-point coordinates, as they consume the bulk of the encoded bitstream. Code computation varies for different encoding modes. For independent encoding mode, each frame is separately encoded and decoded without reference

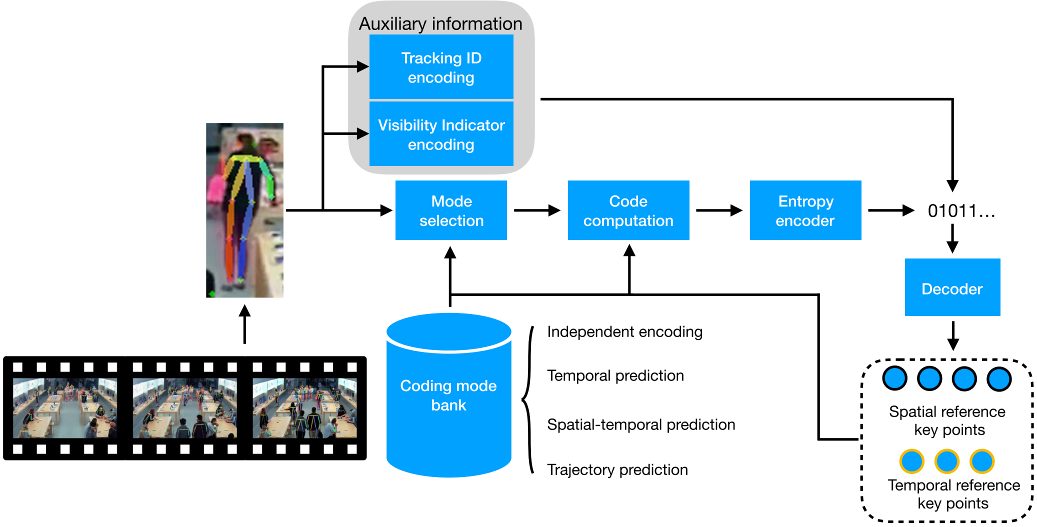


Figure 5. Proposed framework for lossless key-point sequence compression. Best viewed in color.

frames. Given references, a predict-and-encode strategy is developed to realize the encoding based on the temporal, spatial-temporal, and trajectory prediction modes. Residuals between the original data and their predictions are calculated as the codes to be encoded. Prediction residuals are then fed into the entropy encoder to generate the bit-stream. It is worth mentioning that prediction modes for the key points can be adaptively predicted using its spatial and temporal neighbors. Furthermore, the predict-and-encode strategy leverages an adaptive prediction method to combine different prediction modes for key-point sequences with various structures and semantic information. Tracking ID and visibility indicator are also encoded with the auxiliary information encoding module for communication.

Independent Encoding

For independent encoding, the key points of a single object are encoded by considering the spatial correlations without introducing references. We first encode the absolute coordinates of the key point k_s with zero in-degree. Subsequently, the difference of coordinates between two adjacent vertices defined by the incidence matrix (i.e., the edges) is encoded. The residual of independent encoding $r^{\mathrm{IE}}_{i,j}$ between the ith and jth vertices is computed

$$r^{IE}_{i,j} = k_i - k_j. \tag{3}$$

Reference-Based Prediction Modes

Besides independent encoding, three additional prediction modes are developed for temporal prediction to minimize the residuals with temporal references.

Temporal Prediction: For each object, the correlations between consecutive frames are characterized by the movements, including the translation of the main body and twists of some parts. As shown in Figure 6(a), we first obtain a colocated prediction (yellow points in the current frame) of the point from the reference frame by motion compensation with the motion vector of the central key point (yellow vector). Consequently, the temporal prediction can be expressed as

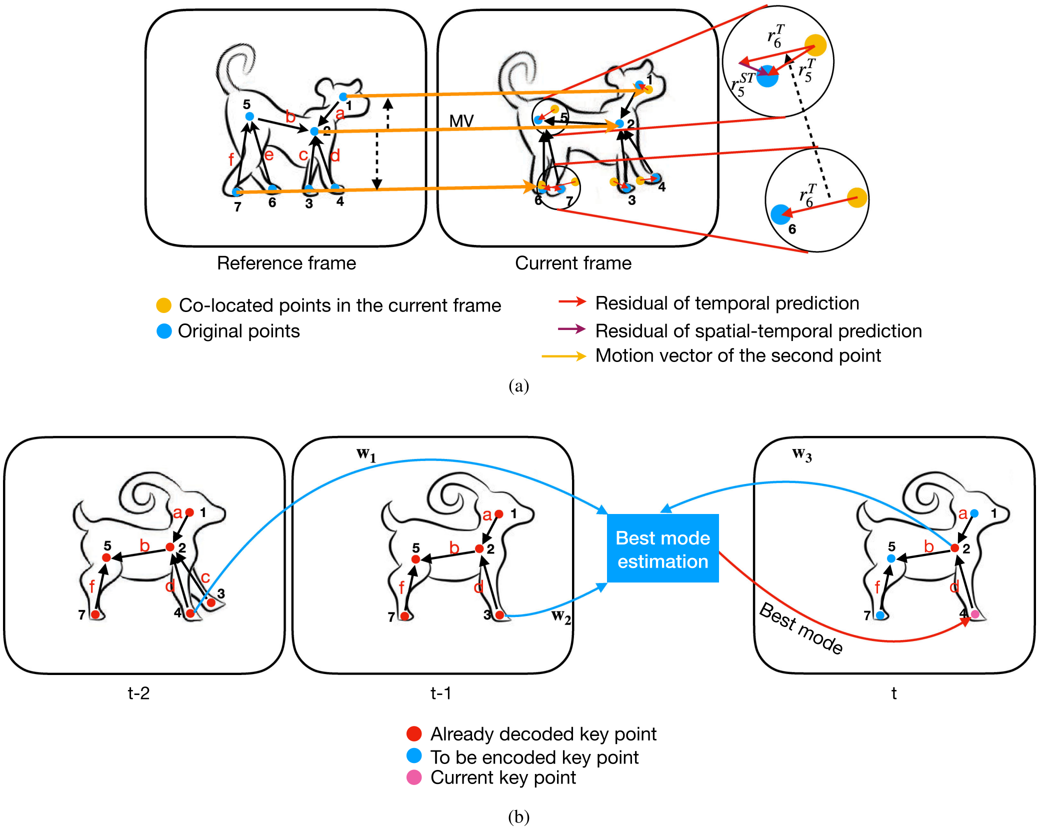
$$p^t_i = k^{t-1}_i + MV_c \tag{4}$$

where $MV_c = k^t_c - k^{t-1}_c$ and k_c is the key point with maximum out-degree in the incidence matrix. The residuals of temporal prediction $r^{T,t}_i$ (red dashed vectors) are computed for transmission and reconstruction in a lossless fashion:

$$r^{T,t}_i = k^t_i - p^t_i. \tag{5}$$

However, temporal prediction would be affected by possible twists, i.e., the gap between colocated (yellow and blue) points in the current frame.

Spatial-Temporal Prediction: The spatial-temporal correlations between key points can be utilized to improve the accuracy of prediction and further reduce the redundancy. Since adjacent


MV
1
2
3
4
5
6
7
a
b
c
d
e
f
r_6^T
r_5^T
r_5^{ST}
Reference frame
Current frame
● Co-located points in the current frame
● Original points
Residual of temporal prediction
Residual of spatial-temporal prediction
Motion vector of the second point
w1
w2
w3
Best mode estimation
Best mode
t-2
t-1
t
● Already decoded key point
● To be encoded key point
● Current key point

(a)

(b)

Figure 6. (a) Illustration of spatial and spatial-temporal prediction modes. (b) Illustration of the best mode estimation with already decoded spatial and temporal references. Best viewed in color.

points in the incidence matrix are highly correlated in the spatial domain, their movements are probably in the same direction and even with the same distance. Thus, the redundancy can be further reduced by encoding the residual of prediction p_i with respect to the prediction $p_{r(i)}$ of its reference point, as their temporal predictions are very close. For example, as shown in Figure 6 (a), the spatial-temporal prediction of the fifth point is obtained with the encoded residual of the sixth point (red vector) and the colocated temporal prediction (fifth yellow point). In this case, we can see that the to be transmitted spatial-temporal residual of the fifth point (maroon vector, r_5^{ST}) is smaller than the residual r_5^T. Formally, $r_i^{ST,t}$ can be computed by

$$r_i^{ST,t} = p_i^t - p_{r(i)}^t \tag{6}$$

where $r(i)$ is the index of ith point's reference. Equation (6) is equivalent to predicting using MV_c and the encoded residual $r_{c(i)}^{T,t}$ of the reference point.

Trajectory Prediction: The above two modes utilize the MV of the central point to accomplish temporal prediction. However, the motions of different parts of an object are complex, as they vary in direction and distance. Thus, the required bits for coding can be further reduced with more accurate prediction. For example, when we assume the motion of an object is uniform in a short time (e.g., three frames), the motion from the $(t-1)$th frame to the tth frame can be approximated with that from the $(t-2)$th frame to $(t-1)$th frame. Its predicted value is

$$tp_i^t = k_i^{t-1} + (k_i^{t-1} - k_i^{t-2}). \tag{7}$$

The residual between the predicted value and actual value is computed and transmitted.

The accuracy of trajectory prediction methods can be improved by incorporating more features at the cost of further complexity.[18] In this article, we propose a simple and efficient linear prediction based on the previous two frames.

Adaptive Mode Selection

Independent encoding mode is adopted, when key points are in the first frame or appear for the first time in sequences. When temporal references are introduced, adaptive mode selection is developed for the candidate temporal, spatial-temporal, and trajectory prediction modes. The prediction mode $m^\star$ is estimated from the encoded spatial and temporal reference points with weighted voting

$$m^\star = \arg\min_{m} \sum_{n \in N_t} w_n^t \times b_n^m + \sum_{n \in N_s} w_n^s \times b_n^m \tag{8}$$

where N_t and N_s are the sets of spatial and temporal reference points, w_n^t and w_n^s are the weights of the corresponding point n in N_t and N_s, m is the candidate modes for temporal, spatial-temporal prediction and trajectory prediction, and b_n^m is the bit-length of point n encoded with m. As depicted in Figure 6(d), the prediction mode of fourth key point in the tth frame is estimated with the reconstructed fourth key point in $(t-2)$th and $(t-1)$th frames, along with the encoded first key point in the tth frame with weights w_1, w_2, w_3, respectively.

Equation (8) indicates that $m^\star$ is determined to minimize the average bit-length of its spatial and temporal reference points encoded with all candidate modes. The weights are hyper-parameters that commonly decrease with the growth of the distance between the current point and its neighbors. Note that trajectory prediction will not always be enabled. For example, the object or point exists in the tth and $(t-1)$th frame would not appear in the $(t-2)$th frame. It is symmetric for the encoder and decoder to determine whether the trajectory prediction is adopted. Thus, we exclude it from the candidate modes, when unavailable.

Auxiliary Information Encoding

In addition to the key points, tracking ID and visibility indicator are encoded as auxiliary information. Note that they actually consume minimal bit-rates in the output bitstream.

Tracking ID: A tracking ID is assigned in arithmetic order to each object when it first appears in the video. For each frame, we sort the objects in ascending order (of tracking IDs) and encode the differences between neighboring tracking IDs.

Visibility indicator: Since visibility indicator changes slowly within two consecutive frames, one bit is used to represent whether it changes for an object. If not, the difference is encoded and transmitted.

EXPERIMENTS

Evaluation Framework

To demonstrate the robustness of the proposed lossless compression method for keypoint sequences, we evaluate four types of key points: 2D bounding boxes, human skeletons, 3D bounding boxes, and facial landmarks.

2D Bounding Box Dataset: MOT17 dataset[7] consists of 14 different sequences (7 training, 7 test sequences). Here, we evaluate the training sequences with ground-truths. We also adopt another important dataset for 2D bounding boxes, i.e., crowd-event BBX dataset,[19] which we have constructed. This dataset includes annotated 2D bounding boxes (and corresponding tracking information) in crowed scenes.

Human Skeleton Dataset: Two datasets are used for human skeleton compression: (1) PoseTrack; (2) Our crowd-event skeleton dataset.[19] For human pose estimation and tracking, PoseTrack[13] is one of the most widely used dataset with over 1356 video sequences. Five challenging sequences that contain 7–12 skeletons are chosen as test sequences in this paper. In our own collected crowd-event skeleton dataset, each skeleton is labeled with 15 key joints (e.g., eyes, nose, neck), as shown in Figure 3(d). Compared with the PoseTrack sequences, our crowd-event skeleton dataset contains a larger number of smaller skeletons in crowded scenes.

nuScenes Dataset: The nuScenes dataset[20] is a large-scale public dataset for autonomous driving. It contains 1000 driving scenes (a 20-s clip is selected for each scene) while accurate 3D bounding boxes sampled at 2 Hz over the entire dataset are annotated.

Facial Landmark Dataset: We collect three video sequences and label the landmark sequences, as existing facial landmark datasets rarely contain tracking information. The sequences contain 11 to 34 visible human faces, each having about 100 frames on average.

Table 1. Average bits for encoding one point and compression ratio for different encoding methods.

	Fixed bit-length coding	Independent encoding	Temporal prediction	Spatial-temporal prediction	Trajectories prediction	Multimodal coding
MOT17	37.41	36.65 (97.97%)	14.77 (39.49%)	14.77 (39.49%)	**13.34 (35.67%)**	14.73 (39.37%)
Crowd-event BBX	38.10	36.58 (96.01%)	**10.54 (27.66%)**	**10.54 (27.66%)**	11.31 (29.67%)	10.90 (28.59%)
PoseTrack	33.35	23.30 (69.86%)	13.84 (41.50%)	13.35 (40.02%)	13.37 (40.08%)	**12.80 (38.38%)**
Crowd-event skeleton	33.79	14.40 (42.62%)	3.17 (9.37%)	3.01 (8.92%)	4.06 (12.02%)	**2.46 (7.27%)**
nuScenes	50.29	35.48 (70.55%)	28.25 (56.18%)	27.92 (55.53%)	30.78 (61.22%)	**27.85 (55.38%)**
Facial landmarks	33.11	10.20 (30.80%)	9.37 (28.31%)	9.33 (28.18%)	9.49 (28.67%)	**9.23 (27.87%)**

The compression performance is evaluated in terms of 1) average bits for encoding one point (i.e., the ratio between total required bits for encoding and the number of encoded key points) and 2) compression ratio (i.e., the ratio between data amount before and after compression). In this article, the size of uncompressed data is calculated by encoding each coordinate of each key point with a 16-bit universal code, e.g., 32 bits for 2D coordinates and 48 bits for 3D coordinates of each key point. In Tables 1 and 2, the average bits for fixed bit-length coding are obtained by summing up the bit-lengths assigned for coordinates and required for encoding auxiliary information like tracking IDs and visibility indicators.

Results

Table 1 reports the performance of different prediction modes. Independent encoding mode is suitable for objects with dense key points (e.g., facial landmark sequences) by exploiting spatial correlations. However, it is inferior to prediction modes based on temporal reference.

The spatial-temporal prediction mode is competitive or slightly better than the temporal prediction mode, due to obvious correlations between spatially adjacent points. The largest performance gap is achieved on PoseTrack, as sports scenes in PoseTrack are regular and predictable. The trajectory prediction mode outperforms other modes on sequences with simple, predictable motions. Consequently, the multimodal coding method is developed to combine different prediction modes and improve compression performance for complex scenes.

The multimodal coding method yields the best average performance on most sequences, which validates the advantages of the proposed scheme. For 2D bounding box sequences, the multimodal coding method is equivalent or slightly inferior to the single prediction mode based on temporal references. This fact implies that the multimodal coding method is more suited for sequences with complex and unpredictable motions, while the reference-based prediction mode would favor key-point sequences with simple and predictable motions, e.g., 2D bounding box sequences.

We further downsample the video sequences for evaluations under various motion search ranges. A number of frames are skipped after each frame during encoding. To validate the effectiveness of our approach in real-world applications, we also conduct experiments on data estimated by existing algorithms and noisy data by adding zero-mean Gaussian noise, where a lot of missing and off-target key points exist. Two benchmark datasets (MOT17 and PoseTrack) are evaluated. Table 2 shows that compression performance drops when the frame skipping range increases. More importantly, under different settings, the multimodal coding method still achieves the best performance on all skeleton sequences. It demonstrates the robustness of our proposed scheme.

CONCLUSION AND OUTLOOK

In this article, we highlight the problem of lossless compression of features and shown its importance in modern urban computing applications. Importantly, we introduce a lossless key-

Table 2. Average bits for encoding one point and compression ratio for different encoding methods with different frame skip scenarios, Gaussian noise level (standard deviation) and data sources.

	Groundtruth?	Frame skip	Noise level	Fixed bit-length coding	Independent encoding	Temporal prediction	Spatial-temporal prediction	Trajectories prediction	Multimodal coding
MOT17	✓	0	0	37.41	36.65 (97.97%)	14.77 (39.49%)	14.77 (39.49%)	**13.34** **(35.67%)**	14.73 (39.37%)
	✓	1	0	37.41	36.66 (97.99%)	18.11 (48.40%)	18.11 (48.40%)	**17.31** **(46.26%)**	18.29 (48.90%)
	✓	2	0	37.41	36.67 (98.01%)	**20.29** **(54.24%)**	**20.29** **(54.24%)**	20.49 (54.77%)	20.67 (55.25%)
	✓	0	2	37.41	36.66 (98.00%)	**18.56** **(49.60%)**	**18.56** **(49.60%)**	19.07 (50.97%)	18.87 (50.45%)
	✓	0	5	37.41	36.63 (97.91%)	**21.78** **(58.23%)**	**21.78** **(58.23%)**	22.76 (60.84%)	22.07 (58.98%)
	✗	0	0	37.41	35.16 (93.99%)	15.46 (41.32%)	15.46 (41.32%)	**14.94** **(39.94%)**	15.75 (42.11%)
PoseTrack	✓	0	0	33.35	23.30 (69.86%)	13.84 (41.50%)	13.35 (40.02%)	13.37 (40.08%)	**12.80** **(38.38%)**
	✓	1	0	33.35	23.26 (69.75%)	16.76 (50.25%)	15.97 (47.89%)	17.60 (52.78%)	**15.92** **(47.74%)**
	✓	2	0	33.35	23.23 (69.65%)	18.64 (55.89%)	**17.67** **(52.99%)**	20.16 (60.46%)	**17.67** **(52.99%)**
	✓	0	2	33.35	23.33 (69.96%)	15.04 (45.10%)	14.62 (43.85%)	15.27 (45.78%)	**14.38** **(43.11%)**
	✓	0	5	33.35	23.43 (70.25%)	17.05 (51.13%)	16.79 (50.34%)	17.52 (52.52%)	**16.50** **(49.48%)**
	✗	0	0	33.35	22.35 (67.01%)	19.14 (57.38%)	19.05 (57.12%)	19.38 (58.10%)	**18.63** **(55.86%)**

point sequence compression approach where both reference-free and reference-based modes are presented. Furthermore, an adaptive mode selection scheme is proposed to deal with a variety of scenarios, i.e., camera scenes, key-point sequences, and motion degree. Forward looking, we expect that key-point sequence compression methods will play an important role in the transmission and storage of key-point data in urban computing and intelligent analysis.

ACKNOWLEDGMENTS

The paper was supported in part by the following Grants: China Major Project for New Generation of AI Grant (No.2018AAA0100400), National Natural Science Foundation of China (Grant Nos. 61971277, 61720106001, 61932022, 61971285).

REFERENCES

1. Sullivan *et al.*, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
2. H. Choi and I. V. Bajić, “Deep feature compression for collaborative object detection,” in *Proc. 25th IEEE Int. Conf. Image Process.*, 2018, pp. 3743–3747.
3. J. Elseberg *et al.*, “One billion points in the cloud–an octree for efficient processing of 3D laser scans,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 76, pp. 76–88, 2013.
4. S. Ren *et al.*, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. Adv. Neural Inf. Process. Syst. 28*, 2015, pp. 91–99.
5. H. Law *et al.*, “CornerNet: Detecting objects as paired keypoints,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.
6. A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 3464–3468.
7. A. Milan *et al.*, “Mot16: A benchmark for multi-object tracking,” 2016, *arXiv:1603.00831*.
8. X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, “Monocular 3D object detection for autonomous driving,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2147–2156.
9. X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3D object detection network for autonomous driving,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1907–1915.
10. D. Frossard and R. Urtasun, “End-to-end learning of multi-sensor 3D tracking by detection,” in *Proc. IEEE Int. Conf. Rob. Autom.*, 2018, pp. 635–642.

11. Z. Cao, T. Simon, S. Wei, and Y. Sheikh, “Realtime multi-person 2D pose estimation using part affinity fields,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7291–7299.
12. H.-S. Fang, S. Xie, Y. Tai, and C. Lu, “RMPE: Regional multi-person pose estimation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2353–2362.
13. M. Andriluka *et al.*, “PoseTrack: A benchmark for human pose estimation and tracking,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5167–5176.
14. T. F. Cootes *et al.*, “Active appearance models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jul. 2001.
15. Y. Sun, X. Wang, and X. Tang, “Deep convolutional network cascade for facial point detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3476–3483.
16. J. Yang, J. Deng, K. Zhang, and Q. Liu, “Facial shape tracking via spatio-temporal cascade shape regression,” in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 994–1002.
17. A. Yao *et al.*, “Efficient facial landmark tracking using online shape regression method,” U.S. Patent 9 361 510, Jun. 2016.
18. R. Q. Mínguez *et al.*, “Pedestrian path, pose, and intention prediction through Gaussian process dynamical models and pedestrian activity recognition,” vol. 20, no. 5, pp. 1803–1814, 2018.
19. W. Lin *et al.*, “Challenge on large-scale human-centric video analysis in complex events,” 2020. [Online]. Available: http://humaninevents.org/
20. H. Caesar *et al.*, “nuScenes: A multimodal dataset for autonomous driving,” 2019, *arXiv:1903.11027*.

Weiyao Lin is currently a full professor with the Department of Electronic Enigeering, Shanghai Jiao Tong University, Shanghai, China. His research interest includes urban computing and multimedia processing. He received the Ph.D degree from the University of Washington, Seattle, USA in 2010. He served as an associate editor for a number of journals including TIP, TCSVT, and TITS. Contact him at wylin@sjtu.edu.cn.

Xiaoyi He is currently working toward the M.S. degree at Shanghai Jiao Tong University (SJTU), Shanghai, China. His current research interests include large-scale video compression and semantic information coding. He received the B.S. degree in electronic engineering from SJTU, in 2017. Contact him at 515974418@sjtu.edu.cn.

Wenrui Dai is currently an associate professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, China. His research interests include learning-based image/video coding, image/signal processing, and predictive modeling. He received the Ph.D degree from SJTU in 2014. Contact him at daiwenrui@sjtu.edu.cn.

John See is a senior lecturer with the Faculty of Computing and Informatics, Multimedia University, Malaysia. He is currently the Chair of the Centre for Visual Computing (CVC) and he leads the visual processing Lab. From 2018, he is also a Visiting Research Fellow at Shanghai Jiao Tong University. Contact him at johnsee@mmu.edu.my.

Tushar Shinde focuses his current research interests on multimedia processing and predictive coding. He received the M.Tech. degree in Information and Communication Technology from Indian Institute of Technology, Jodhpur (IITJ), India. He is currently working toward the Ph. D degree at IITJ. Contact him at shinde.1@iitj.ac.in.

Hongkai Xiong is a distinguished professor with the Department of Electronic Engineering, Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, China. He is currently the Vice Dean of Zhiyuan College, SJTU. His research interests include multimedia signal processing and coding. He received the Ph.D. degree from SJTU in 2003. Contact him at xionghongkai@sjtu.edu.cn.

Lingyu Duan is currently a full professor with the National Engineering Laboratory of Video Technology, School of Electronics Engineering and Computer Science, Peking University (PKU), Beijing, China. He was the associate director of the Rapid-Rich Object Search Laboratory, a joint lab between Nanyang Technological University, Singapore, and PKU, since 2012. Contact him at lingyu@pku.edu.cn.