This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

Cloud-based Actor Identification with Batch-Orthogonal Local-Sensitive Hashing and Sparse Representation

Guangyu Gao, *Member, IEEE*, Chi Harold Liu, *Senior Member, IEEE*, Min Chen, *Senior Member, IEEE*, Song Guo, *Senior Member, IEEE* and Kin K. Leung, *Fellow, IEEE*

***Abstract*—Recognizing and retrieving multimedia content with movie/TV-series actors, especially querying actor-specific videos in large scale video dataset has attracted much attention in both video processing and computer vision research field. However, many existing methods have low efficiency both in training and testing processes and also less satisfactory performance. Considering these challenges, in this paper, we propose an efficient cloud-based actor identification approach with Batch-Orthogonal Local-Sensitive Hashing (BOLSH) and Multi-Task Joint Sparse Representation Classification (MTJSRC). Our approach, is featured by: (i) videos from movie/TV-series are segmented into shots with the cloud-based shot boundary detection; (ii) while faces in each shot are detected and tracked, the cloud-based BOLSH is then implemented on these faces for feature description; (iii) the sparse representation is then adopted for actor identification in each shot; (iv) finally, a simple application, actor-specific shots retrieval is realized to verify our approach. We conduct extensive experiments and empirical evaluations on a large scale dataset, to demonstrate the satisfying performance of our approach considering both accuracy and efficiency.**

***Index Terms*—Actor Identification, Cloud Computing, Shot Boundary Detection, Locality-Sensitive Hashing, Sparse Representation.**

I. Introduction

With rapid advances in digital technologies, there has been profound development in videos, especially the feature movies and TV series. Moreover, the new generation cellular networks with high transmission rate and energy efficiency provide a new approach for multimedia wireless communications, which combine the digital technologies and wireless communications to satisfy the requirement of quality of service [1]. In order to feasibly browse and retrieval these videos, it is very crucial and urgent to provide efficient and effective techniques for video analyzing and understanding. Firstly, there are several works focused on video analyzing in surveillance video, i.e. Ma *et al.* [2] built an efficient system for robust and fast people counting under occlusion through multiple cameras. Meanwhile, automatic actor identification is one of the most important techniques for video analyzing in broadcast videos, since actor identification is to label actor in videos with their corresponding names. In a movie/TV series, the actors are often the most important contents to be indexed, thus actor identification becomes a critical step in video semantic analysis (Video always refers to movie and TV series in this paper unless otherwise specified.), i.e., semantic movie index and retrieval, summarization. As mentioned in [3], [4], recently, multimedia content providers have started to offer information on cast and characters for TV series and movies during playback.

Actually, the face recognition is the most common way used for actor identification. Sang *et al.* [5] proposed the problem of faceted subtopic retrieval, which focus on more complex queries concerning political and social events or issues. Meanwhile, some of the researchers proposed to share aligned faces in a carefully crafted benchmark face recognition dataset such as the Labeled Face in the Wild[1]. By automatically detecting faces throughout the video, extracting facial features and then using these features in a supervised or unsupervised clustering process, actors can be identified and labeled. Therefore, the actor identification is generally divided into several steps: video segmentation, face detection and tracking, face and actor recognition, and actor-specific retrieval.

As has been noted in [6], although it is very intuitive to humans, automatic actor identification is still tremendously challenging due to: 1) the lack and ambiguity of available annotations; 2) many other factors, like pose, light and expression, etc., influent the way a face appeared in a frame; and 3)when there are many uncontrolled data quality factors, such as low resolution, occlusion, nonrigid deformation, large motion and complex background, which make the results of face detection and tracking unreliable; 4) the efficiency is always a concern for video processing and analysis, and it is

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.
Corresponding author: Chi Harold Liu.

G. Gao is with the School of Software, Beijing Institute of Technology, Beijing 10081, China. E-mail: guangyugao@bit.edu.cn

C. H. Liu is with the School of Software, Beijing Institute of Technology, Beijing 10081, China, and also with the Department of Computer Information and Security, Sejong University, Seoul 143-747, South Korea. E-mail: chiliu@bit.edu.cn

M. Chen is with the School of Computer Science and Technology, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan 430074, China. E-mail: minchen2012@hust.edu.cn

G. Song is with the School of Computer Science and Engineering, The University of Aizu, Tsuruga, Ikki-machi, Aizu-Wakamatsu City, Fukushima 965-8580, Japan. E-mail: sguo@u-aizu.ac.jp

K. K. Leung is with the EEE and Computing Departments, Imperial College, London SW7 2BT, UK. E-mail: kin.leung@imperial.ac.uk

This work was supported by the National Natural Science Foundation of China under Grant No. 61572220, No. 61401023 and No. 61300179. This work was also supported by the Fundamental Research Funds for the Central Universities under HUST: 2016YXMS070.

[1]http://vis-www.cs.umass.edu/lfw/index.html

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

still a unresolved problem that how to balance the efficiency and accuracy of actor identification in videos.

In order to deal with these challenges, in this work, we present a novel cloud-based actor identification approach with Batch-Orthogonal Local-Sensitive Hashing (BOLSH) [7] and sparse representation. Firstly, since there are full of various images as well as video clips in the Internet for actors, we propose to do a matching between the faces detected from the video and the exemplar faces in the gallery set, which have been searched from the Web. For the second and third challenge, when each face is detected and tracked, we use the BOLSH method to provide multi batch features, and then we used the Multi-Task Joint Sparse Representation and Classification (MTJSRC) [8] to accurately recognize face tracks. Since the *batches* is the key concept in our approach, we renamed the used sparse representation algorithm as Multi-Batches Joint Sparse Representation and Classification (MBJSRC) in this paper. And also, the kernel view of that even achieved more robust performance.

In order to deal with the fourth challenge, we introduce the Apache Spark[2] based cloud computing both for pre-processing of video segmentation and BOLSH hashing on massive face images. The cloud-based way can offer high efficiency but also maintain satisfactory identification performance. Finally, based on the results of actor identification, face tracks in each shot will be assigned with actor names, and further application of actor-specific shots retrieval is also presented.

All in all, compared with previous studies on such topic, the main contributions of this paper include:

- The cloud based Spark framework is introduced to accelerate the shot boundary detection by distributing processes on all pixels into a parallel environment.
- The BOLSH method is used for feature description which not only reduce the feature dimension but also maintain the similarity between instances.
- With BOLSH, we need to hash thousands or more faces into hash values, which will result in very low efficiency. However, the *batch* concept in BOLSH make it easy to do hashing in parallel with the cloud computing ideas, and the Spark framework is used to assign the hash processes into different virtual machines.
- The MTJSRC algorithm is a robust way for face recognition, and the *batches* in BOLSH is exactly satisfies the *tasks* in MTJSRC. Thus, the Multi-Batches Sparse Representation and Classification (MBJSRC) is constructed for actor identification on face recognition.

II. Related Work

The task of actor identification in a movie/TV-series is typically accomplished by combining multiple sources of information, e.g. image, video and text, under little or even no manual intervention. However, in movies/TV-series, the names of actors are not always available, and the appearances of actors vary in different conditions, which makes it hard to detect, track and recognize these actors.

[2]http://spark.apache.org/

Over the past two decades, extensive research efforts have been actively concentrated on this task [6], [9], which detect actor faces in photos or movies and associates them with corresponding names. Besides, there are also several methods using audio clues or both audio and vision clues, such as [10]. Meanwhile, in our previous work [11], we also proposed a semi-supervised learning strategy to address celebrity identification with collected celebrity data. More recently, Tapaswi *et al.* [10] presented a probabilistic method for identifying actors in movies/TV-series, and Bojanowski *et al.* [12] learned a joint model of actors and actions in movies using weak supervision provided scripts.

However, actor identification in video still faced a series of challenges, i.e. many factors, like pose, light and expression, etc., influence the way a face appears. Meanwhile, many uncontrolled data quality factors, such as low resolution, occlusion, nonrigid deformation, large motion and complex background, also make the results of actor identification unreliable for most image based recognition, and the situation is even worse in movies.

In order to maintain the intra-class similarity and differentiate the inter-class samples, a possible and effective way is to use the hashing methods. In the meantime, face or actor features always have very high dimensions and also the number of samples is still very large. Thus, the hash projection can not only maintain characteristics for classification and recognition, but also reduce feature dimensions for more efficient processing. Considering the characteristics of 'batch' ideas in our previous work [7] for hash projection, we combined the batch-orthogonalized random projection to generated tasks for multi-task joint sparse representation and classification [8].

In addition, Sang *et al.* [13] presented two schemes of global face-name matching based frameworks for robust character identification. Their experimental results shown that their approach is useful to improve results for clustering and identification of the face tracks extracted from uncontrolled movie videos. However, they only used 15 feature-length movies, in which, the training set has 1327 face tracks, and the testing set has 5012 tracks. Therefore, Zhang *et al.* [14] have constructed a "Celebrities on the Web" dataset which contains 2.45 million distinct images of 421436 celebrities and is orders of magnitude larger than previous datasets. Consequently, with the large-scale of the massive face or actor-based video data, the efficiency became a more and more crucial problem. Often, the problems with facial recognition based actor identification are rooted in the need for greater processing power, human and machine. Furthermore, the efficiency problems are common issues in the computer vision and pattern recognition areas. In the same time, cloud computing as a model for enabling ubiquitous network access to a shared pool of configurable computing resources, has been enjoying its flourishing.

Cloud-based methods or applications always archive more efficient performance [15], [16], [17], [18]. For example, Gao *et al.* [15] proposed a new framework of providing Handwritten Character Recognition as a Service based on cloud computing technology. Wang *et al.* creatively proposed a cloud-based approach to protect user's data, enhance media

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

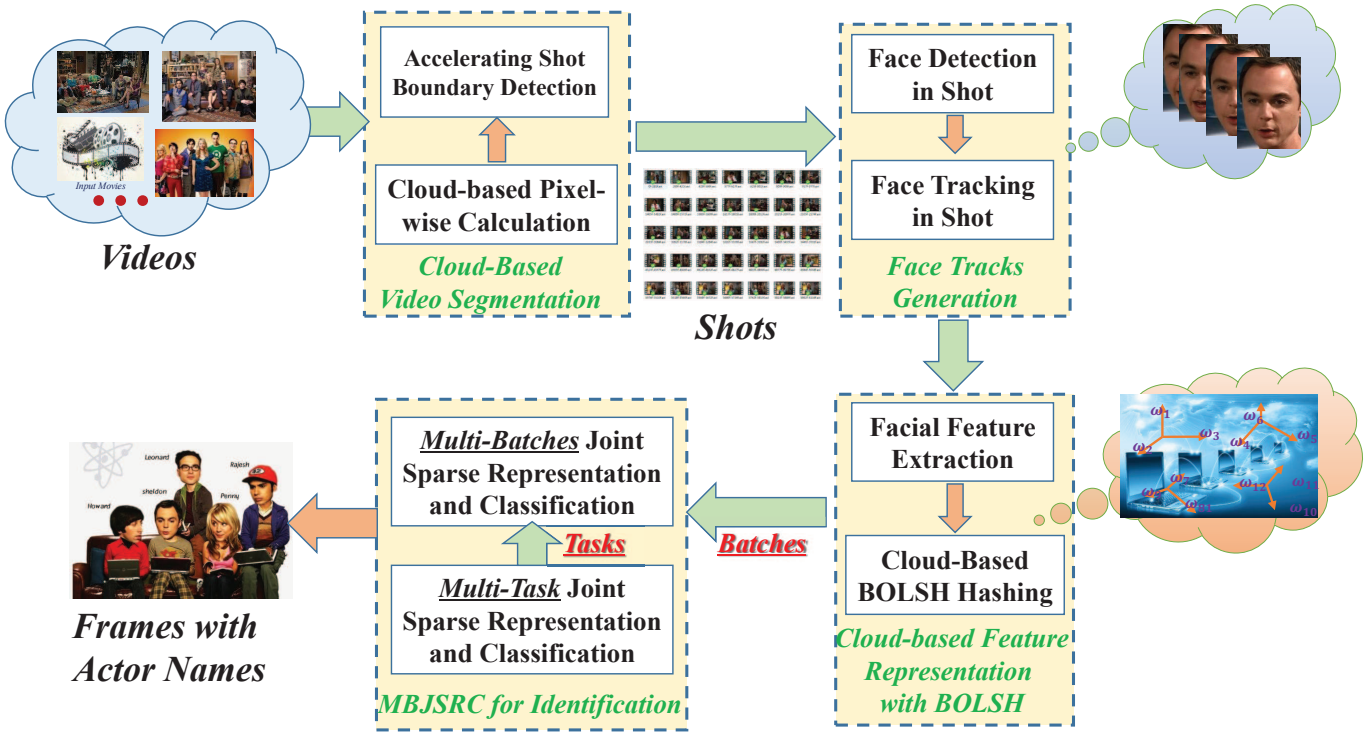

Fig. 1: Scheme illustration of the proposed Cloud-based Actor Identification.

quality and reduce transmission overhead [19]. A cloud based food recognition platform, in which an improved 2DPCA algorithm is used for object recognition, and a Hadoop based cloud server is built for this platform [16]. In [20], Shamim and Ghulam, proposed a cloud-supported framework, where speech and faces images are extracted from health monitoring purposes. Zhang *et al.* proposed a cloud-assisted drug recommendation services to provide significantly more available, reliable and efficient performance [21]. Lai *et al.* realized a network and device aware QoS approach for cloud-based mobile streaming, which effectively solves the limited bandwidth problem available for mobile streaming and different device requirements [22]. Suzuki *et al.* [17] utilized the cloud system to maintain large-scale database which includes learning keypoints. We further note that in [23], Lin *et al.* proposed a green video transmission algorithm in the mobile cloud networks. Their work utilized video clustering and channel assignment to achieve high quality video transmission.

Meanwhile, Ma *et al.* comprehensively addressed the objectives and scientific challenges of Internet of Things (IoT) [24], [25], [26]. Sheng *et al.* in [27] extensively studied the energy-efficient device-to-device (D2D) communication scheme by cooperative relaying in wireless multimedia networks. Liu *et al.* in [28], [29], [30] presented a novel resource negotiation scheme bridging between dynamic sensing tasks and heterogeneous sensors. Liu *et al.* in [31], [32], [33] proposed a novel framework and subsequent participant selection and incentive mechanism for participatory crowdsourcing including the smart device users, central platform and multiple task publishers. In [34], existing incentive mechanism are extensively surveyed and future research directions are clearly given. Liu *et al.* [35] extensively analyzed the relationship between energy consumption and smart device user behaviors, and then proposed a novel approach to select the optimal amount of participant while considering possible user rejections. Song *et al.* [36] introduced an energy consumption index to quantify the average degree of how participants feels disturbed by the energy cost, and proposed a suboptimal approach for participant selection under the multi-task sensing environment. Liu *et al.* [37] presented a quite novel family-based healthcare monitoring system for long-term chronical disease caring. Event detection systems and energy efficient approaches are given in [38], [39] including both centralized optimal approach and fully distributed suboptimal solutions by participatory sensing. Furthermore, Zhang *et al.* [40] focused on privacy leakage issues of participatory sensing and presented a participant coordination based architecture and flow to successfully protect user privacy. Finally, Yurur *et al.* in [41] presented a few posture detections schemes by using the sensor equipped smart devices.

Nevertheless, Apache Spark is a fast and general-purpose cluster computing framework for cloud computing. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. Therefore, we used the cloud ideas of Spark framework to fast the shot boundary detection and feature hashing processing, which cost most of the time and result in low efficiency in the whole framework.

Network transmission and energy consumption is also a big issue for cloud computing. Liu *et al.* [42] presented a novel concept of quality of service (QoS) index to integrate the multi-dimensional QoS requirements to ensure the degree of QoS satisfactions. In [43], the authors proposed a novel MIMO routing scheme to ensure QoS. Liu *et al.* [44] proposed

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

a novel localization-oriented sensing model and a new notion of coverage, Localization-oriented coverage (L-coverage for short), by using Bayesian estimation theory. Yu *et al.* [45] proposed a stochastic load balancing scheme, and finally provide probabilistic guarantee against the resource overloading with virtual machine migration, while minimizing the total migration overhead. Yu *et al.* [46] considered the problem of scaling up a virtual network abstraction with bandwidth guarantee in Cloud datacenters. The authors in [47] efficiently optimized the tradeoff between the energy consumption of wireless camera sensor networks and the quality of target localization.

III. Overview of Cloud-based Actor-specific Shots Retrieval

As shown in Fig. 1, our actor identification framework mainly includes four parts: cloud-based video segmentation, face tracks generation, cloud-based feature representation, and MBJSRC for recognition. Besides, we propose a actor-specific shots retrieval application based on these four parts. For the first part, namely, video segmentation, we revised an accelerating shot boundary detection in our previous work [48] by adding the parallel computing with Spark for massive pixels processing. Then, the face tracks are generated with efficient face detection and tracking methods. After, in feature representation, the SIFT features in face tracks are hashed in to new feature space with BOLSH, and also the feature hashing and dimension reducing is realized by cloud-based hashing with Spark. Finally, ideas of 'batch' in BOLSH is mapped into 'task' in the multi-task joint sparse representation algorithm, to form our classification algorithm named MBJSRC.

Hitherto, with the proposed framework, each face track has been assigned with a actor name. Based on the results of actor identification, there are many applications, such as actor/character-specific movie retrieval, personalized video summarization, intelligent playback and video semantic mining, etc. Meanwhile, with the cloud-based shot boundary detection, each video has been segmented into several shots. Actually, there are always several face tracks in each video shot, and each shot can be assigned with several actor names, which is the key word for actor-specific shots retrieval. More specifically, a cloud-based shot boundary detection method is applied to divide the movie into several shots at first. Secondly, the face detection and tracking processing are applied, and after the identification of all the detected face tracks in these shots, each shot will been labeled as several actor names. Finally, by using the character name or actor name as the query entry, the corresponding actor's spotlights shots are presented to the user.

IV. Cloud-based Shot Boundary Detection

Here, the shot changes are automatically detected using the cloud version of our previous accelerating shot boundary detection method.

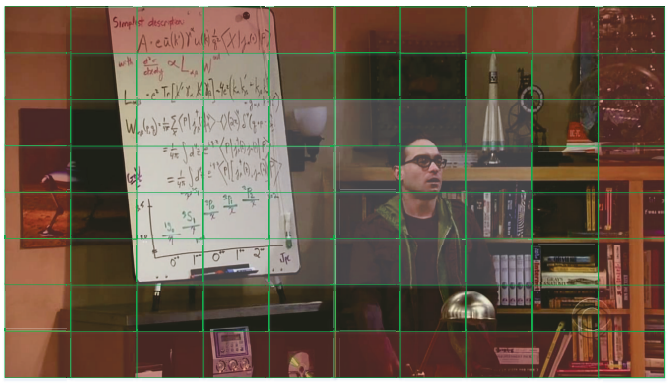

Fig. 2: Illustration of Focus Region. The frame is partitioned into 8×10 sub-regions, and the outermost round sub-regions are non-focus region, second outermost round sub-regions are the second focus region, remaining sub-regions are the most focus region.

A. Accelerating Shot Boundary Detection

At first, we described the original accelerating shot boundary detection as:

1) We accelerate the shot boundary detection process in spatial domain in two aspects: one, by processing only the pixels in *Focus Regions*.
   Specifically, a video has thousands of frames, and each frame has thousands of pixels. These vast frames and pixels make the computation complexity very high, which is the main reason that many shot boundary detection methods or systems have low efficiency. Although spatial sub-sampling of frames has been suggested to improve video processing efficiency, it still depends on the choice of the spatial window. Smaller window size is sensitive to object and camera motions, while arbitrary window size could not make the remaining pixels represent the frame well.
   Generally, the most essential information in a frame is always concentrated around the center of a frame, and the more the pixels are close to the frame center, the more important the pixels are. In order to reduce the processing time, redundant pixels should be removed and only informative pixels are kept for processing. To accomplish this, a Focus Region is defined for each frame. The Focus Region of a $P \times Q$ sized frame is extracted in the following steps.
   1-1 Each image is divided into non-overlapping sub-regions of size $(P/p) \times (Q/q)$ to get $p \times q$ number of sub-regions.
   1-2 The most external surrounding sub-regions (Colored with red in Fig. 2) are defined as the non-focus region.
   1-3 The outer-most external surrounding sub-regions (Yellow sub-regions) are defined as second focus region.
   1-4 Remaining sub-regions around the center are defined as focus region.

   To get an informative while compact representation of

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

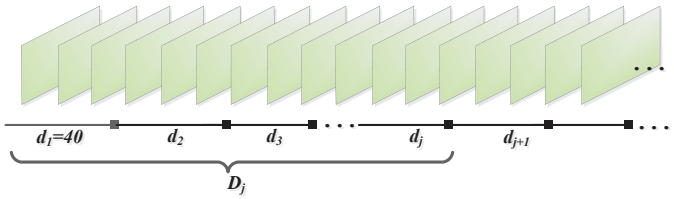

Fig. 3: Illustration of adaptive skipping interval.

a frame, the non-focus region is discarded, the second focus region is down-sampled by keeping only pixels with odd x-coordinates. The focus region is fully kept.

2) We accelerate the shot boundary detection process in temporal domain by skipping frames adaptively. Instead of degrading the accuracy, almost all boundaries could be detected including *gradual transitions* which are hard to be detected. In order to efficiently reduce the number of processed frames and also not to drop any boundaries between two shots, we set the initial skipping interval as d_1. Then, the following skipping intervals are updated adaptively based on the similarity of frames. As shown in Fig. 3, $\{d_1, d_2, \ldots\}$ denotes the sequential skipping intervals, and $\{D_1, D_2, \ldots\}$ is the serial frame number in the original video corresponding to all skipping intervals. D_k is defined as

$$D_k = \sum_{m=1}^{k} d_m. \quad (1)$$

When $\alpha_{x,y}$ is the similarity ratio between the x_{th} and y_{th} frame, we update the skipping interval d_j as follows.

$$d_j = \sum_{k=1}^{j-1} \frac{1}{j-1} \alpha_{D_{d-1}, D_k} d_k \quad (6). \quad (2)$$

That is, the greater α is, the larger of the skipping interval is. These updated intervals are reasonable. Because a great α in current skipping interval means the skipped frames are very similar, we can boldly skip more frames. But if α is small, it implies there are many changes in the skipped frames and we need to cautiously skip frames, so as to avoid classifying a motion as a shot boundary and also avoid missing shots with less frames. Generally, human visual reaction time is about $1-2$ seconds. Suppose the video frame rate is about $20-25fps$, then a shot that can cause visual reaction need last for $20-50$ frames at least. Therefor, the initial skipping interval d_1 is set to 40.

After, given the current processed frame F_i, if $\alpha_{i,i+d_j} > T_\varepsilon$ (the threshold assigned in experiment), we can assert that F_i is similar to F_{i+d_j}, and skip to process the next d_{j+1} frames. Otherwise it means that there is a shot boundary existing between F_i and F_{i+d_j}. Thereby, we use a bisection search to find a refined boundary in this range. First, we compute $\alpha_{i,i+d_j/2}$ and $\alpha_{i+d_j/2,i+d_j}$. If $\alpha_{i+d_j/2,i+d_j} > T_\varepsilon$, boundary lies in the first half of F_i to F_{i+d_j}, otherwise in the second half. Then, the same process is carried out in the first half or second half of F_i to F_{i+d_j} to refine the boundary position until half of the range is only one frame.

We evidently accelerated the shot boundary detection process and detect gradual transitions more robustly, no matter if the gradual transition is fade in/out, dissolve or wipe. Moreover, it requires to compute mutual information for $n-1$ times on a video sequence of n frames by using traditional frame by frame searching process, but in our approach, we just need to compute it for $\log n$ times.

3) A corner can be defined as the intersection of two edges. A corner can also be defined as a point for which there are two dominant and different edge directions in a local neighborhood of the point. The corner distribution is the distribution of all detected corners scattered in a image. Thus, the corner distribution of frames near candidate shot boundaries is adopted to remove most of the false boundaries and to find the precise interval of the true boundary. So far, it nearly detected all the boundaries. However, camera or object motion could also lead to significant change of frame content when we skip frames aggressively. Thus, several false shot boundaries are caused by camera or object motion, which are the main false boundaries. In order to remove these false boundaries, we used the corner distribution analysis. More specifically, 1) in abrupt transitions, a frame abruptly changes into a totally different one; 2) changes in gradual transitions always last about $5-20$ frames, which couldn't be felt by audience; 3) changes in false boundaries always last more than 100 frames. Actually, corner distribution of frames in true boundary (abrupt and gradual transitions) is very different from its forward and backward frames, but it is more stable and consistent in camera and object motion caused false alarms.

B. Cloud-based Mutual Information Calculation

Although our original shot boundary detection really accelerated the shot boundary detection, it still cost unacceptable time for massive videos. More specifically, we found that most of the time is cost for entropy and *mutual information* calculation in and between frames. In fact, in the mutual information calculation on the Focus Regions, the gray value of each pixel is summarized and then the portion of each gray value (0 to 255) is calculated. After, these portion is looked as distribution probability to generate the entropy with Shannon Theory. By analyzing the whole flowchart of *mutual information* as well as frame similarity calculation, an intuitive idea is that a data-parallel programming model for clusters of commodity machine can handle this issue well. Thus, we used the Spark framework for *mutual information* calculation.

Specifically, entropy measures the information content or "uncertainty" of X and is given by:

$$H(X) = -\sum p_X(x) \log p_X(x). \quad (3)$$

The joint entropy of *X,Y* is defined as:

$$H(X,Y) = -\sum p_{XY}(x,y) \log p_{XY}(x,y). \quad (4)$$

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

The *mutual information* between the random variables *X* and *Y* is defined as:

$$I(X,Y) = H(X) + H(Y) - H(X,Y). \quad (5)$$

Let $V = \{F_1, F_2, \ldots, F_N\}$ denotes the frames of a video clip V. For two frames (i.e. F_x and F_y), we first compute their own entropies(i.e. H_x, H_y) and their joint entropy (i.e. $H_{x,y}$). The *mutual information* between them is given by Equation (5). If $I^R_{x,y}$, $I^G_{x,y}$, $I^B_{x,y}$ respectively represent the *mutual information* of each RGB component, we set $I_{x,y} = I^R_{x,y} + I^G_{x,y} + I^B_{x,y}$ as the *mutual information* between frame F_x and F_y.

Generally, Spark provides the Resilient Distributed Dataset (RDD) abstraction trough a language integrated API in Scala[3]. In the cloud version of the shot boundary detection, we calculate the entropy and mutual information with Spark programming. In fact, we used several basic functions in Spark, i.e., $map()$, $reduce()$ etc. Analyzing Equation 3, the following pseudo-program implements the entropy calculation processes.

1) val points = sc.parallelize(list(pixels in a image))
2) val p = points.map($x => (x, 1)$).reduceByKey($(x, y) => x + y$).collect
3) var $sq = width * height$
4) p=p.mapValues(_/sq)
5) p=p.mapValues($x => (-x * logx)$)
6) var ha = p.reduce($(x, y) => x + y$)

We start by defining a RDD called points as which refer to all the pixels in a image. Actually, the calculation of joint entropy is with the same code to realized Equation 4, and we can get the mutual information between two frames with Equation 5.

With the above pseudo-program, a series of processes for all the pixels are distributed into different virtual machines in parallel. After, the whole calculation efficiency has been improved obviously, which will been shown in the experimental results.

V. Faces Detection and Tracking in Shots

The OKAO face detector [4], is used to detect frontal faces and profile faces with 30 degree towards left or right in frame. Actually, a typical movie may contains tens of thousands of detected faces. However, these faces merely arise from a few hundred "tracks" of a particular actors. Therefore it is feasible to discover the correspondences between faces and reduce the volume of the data that needs to be processed. Furthermore, stronger appearance models can be built for each actor since a face track provides multiple examples of the actor's appearance. To obtain face tracks, a robust foreground correspondence tracker [49] is applied for each shot. In practice, the face detection algorithm will be tried in the first few frames of a shot, and it will go with tracking only if face is detected. And if the faces of a actor are occluded in the beginning of a shot, that actor can not be detected and identified.

[3]http://www.scala-lang.org

[4]http://www.omron.com/r_d/coretech/vision/okao.html

Using the tracking algorithm in [49], with the assumption that the target face region can be represented by a set of superpixels without significantly destroying the boundaries between target and background, we model the prior knowledge regarding the target and the background appearance by

$$y_t(r) = \begin{cases} 1, & if\ sp(t,r) \in target, \\ -1, & if\ sp(t,r) \in background. \end{cases} \quad (6)$$

Here $sp(t,r)$ denotes the r_{th} superpixel in the t_{th} frame, and $y_t(r)$ denotes its corresponding label. A robust superpixel-based discriminative appearance model is generated based on four factors: cluster confidences, cluster centers, cluster radius and cluster members. This discriminative appearance model facilitates a tracker to discriminate the face region and the background with mid-level cues. After, the target-background confidence map is used to formulate the tracking task, and the best candidate is obtained by the maximum a posterior estimates. With the superpixels tracking, we collect faces belonging to tracks efficiently and accurately, and more details about the tracking algorithm can be seen in [49]. However, short tracks which are often introduced by false positive detections are discarded, and an example of the final face tracks is shown in Fig. 4.

To extract face features and construct the representations, a part-based descriptor extracted around local facial features [6], [9] is utilized. Here we first use a generative model [6] to locate the nine facial key-points in the detected face region, including the left and right corners of each eye, the two nostrils and the tip of the nose and the left and right corners of the mouth. Then we extract the 128-dim SIFT descriptor from each key-point and directly concatenate them together to form our final face descriptor with dimensionality 1152. Fig. 5 illustrates some selected faces with facial feature points marked in our approach.

VI. Cloud-based BOLSH

In order to make large-scale image or video processing practical, *Locality-Sensitive Hashing* is one of the way. Because it reduces the dimensionality of high-dimensional data, namely, it hashes input items so that similar items map to the same buckets with high probability. That is, Locality Sensitive Hashing can not only maintain the similarity between items, but also reduce the feature dimensions.

Sign-Random-Projection Locality-Sensitive Hashing (SRP-LSH) is a widely used hashing method, which provides an unbiased estimate of pairwise angular similarity, yet may suffer from its large estimation variance. We propose the Batch-Orthogonal Locality-Sensitive Hashing (BOLSH), as a significant improvement of SRP-LSH [7]. The proposed BOLSH not only has the properties of *Locality-Sensitive Hashing* on maintaining item similarity and reduce dimensions, but also easy to applied to the cloud computing framework with several independent *batches*.

A. BOLSH

Locality-sensitive hashing aims to hash similar data samples to the same hash code with high probability. Based on the

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

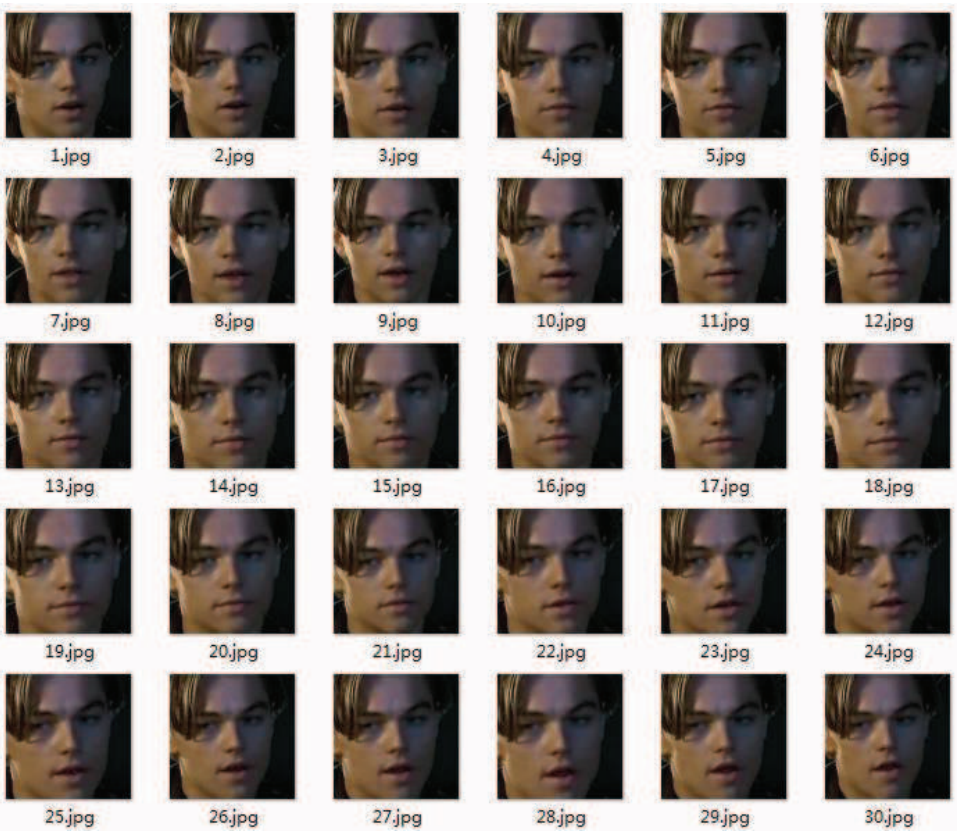

Fig. 4: Examples of the first 30 faces of a face track for 'Jack' in Titanic, and the number below the image is the index of the face in the face track.

Fig. 5: Examples of detected face with facial feature points.

locality-sensitive property, a fundamental usage of *locality-sensitive hashing* is to generate sketches, or signatures, or fingerprints, for reducing storage space while approximately preserving the pairwise similarity. These sketches or signatures can be used for higher-level applications, e.g., clustering, near-duplicate detection. Moreover, *locality-sensitive hashing* can further be used for efficient approximate nearest neighbor search, which is one of its most important applications. We can index the hash code in an efficient way, i.e., in hash tables, to enable efficient search for similar data samples to a query.

SRP-LSH is an important binary *locality-sensitive hashing* method, which is widely used and extensively studied. The Hamming distance between two codes of *SRP-LSH* provides an unbiased estimate of the pairwise angular similarity. Although *SRP-LSH* is widely used, it may suffer from the large variance of its estimation. In our previous work [7], we proposed Batch-Orthogonal Locality-Sensitive Hashing (BOLSH), as an improvement over *SRP-LSH*. Instead of independent random projections, BOLSH makes use of batch-orthogonalized random projection vectors, as illustrated in Fig. 6. It is proven in [7] that BOLSH also provides an unbiased estimate of pairwise angular similarity, and has a smaller variance than *SRP-LSH* when the angle to estimate is in $(0, \pi/2]$.

The proposed BOLSH method is closely related to many recently proposed *principal component analysis*-style learning-based hashing methods, which learn orthogonal projections. Although BOLSH is purely probabilistic and data-independent, the model of orthogonal random projection together with its theoretical justifications can help gain more insights and a better understanding of these learning-based hashing methods. Furthermore, since theoretical analysis and experiments both show that BOLSH approximates the angle between two vectors more accurately, BOLSH, in replace of *SRP-LSH*, can be used in various applications requiring massive angle-related computations, e.g., dot product, angular

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

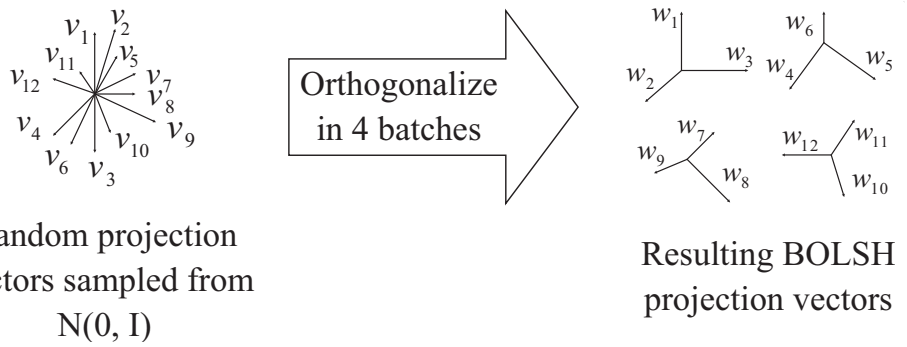

Fig. 6: Examples of 12 BOLSH projection vectors ω_i generated by orthogonalizing independent random projection vectors v_i in 4 batches.

similarity, cosine similarity, Euclidean distance.

SRP-LSH [7] is a widely used *locality-sensitive hashing* method for angular similarity, which embeds real vectors into Hamming space. Angular similarity is defined as follows:

$$sim(a,b) = 1 - \theta_{a,b}/\pi \tag{7}$$

where $\theta_{a,b} = arccos(\frac{\langle a,b\rangle}{\|a\|\|b\|}) \in [0,\pi]$ is the angle between vector a and b, and $\langle a,b\rangle$ means the inner product.

Meanwhile, a *SRP-LSH* function is defined as,

$$h_v(x) = sgn(v^T x), \tag{8}$$

where v refers to a random vector sampled from the normal distribution $\mathcal{N}(0, I_d)$ and

$$sgn(z) = \begin{cases} 1, & z \geq 0, \\ 0, & z < 0. \end{cases} \tag{9}$$

Given two data samples a and b, the *locality-sensitive* is that,

$$Pr(h_v(a) \neq h_v(b)) = \frac{\theta_{a,b}}{\pi}. \tag{10}$$

By independently sampling K d-dimensional vectors $v_1, \ldots, v_K$ from the normal distribution $\mathcal{N}(0, I_d)$, a binary-vector-valued function $h(x) = (h_{v_1}, h_{v_2}, \ldots, h_{v_K}(x)$, which concatenates K *SRP-LSH* functions, thus produces K-bit codes. Then by the locality-sensitive property, it is easy to prove that

$$\mathbb{E}[d_{Hamming}(h(a), h(b))] = \frac{K\theta_{a,b}}{\pi} = C\theta_{a,b}, \tag{11}$$

where $C = K/\pi$.

Based on the *SRP-LSH*, the ideas of BOLSH is just to orthogonalize $N(1 \leq N \leq min(K,d) = K, 1 < K \leq d$ of the random vectors sampled from the normal distribution $\mathcal{N}(0, I_d)$, where d is the dimension of data space. With orthogonalization, the resulting N vectors are no longer independently sampled, thus we group their corresponding bits together as a N-batch, and N is called the batch size. Formally, assuming that $K = N \times L$, and $1 \leq N \leq d$, K random vectors $v_1, v_2, \ldots, v_K$ are independently sampled from the normal distribution $\mathcal{N}(0, I_d)$, and then are divided into L batches with N vectors each. The QR decomposition is processed to these L batches of N vectors respectively. After, we get $K = N \times L$ projection vectors $w_1, w_2, \ldots, w_K$. This results in K BOLSH functions $(h_{w_1}, h_{w_2}, \ldots, h_{w_K})$, where h_{w_i} is defined as

$$h_{w_i}(x) = sgn(w_i^T x). \tag{12}$$

In conclusion, with the BOLSH algorithm, each data sample with d dimensions is transferred into a K dimensions vector with N batches ($K = N \times L$, each batch have L dimensions). Actually, the *batch* in BOLSH exactly matches the element of *task* in the following used multi-task joint sparse representation and classification algorithm for video face recognition.

B. BOLSH by Cloud Computing

With input: Data space dimension d, batch size $1 \leq N \leq d$, the number of batches $L \geq 1$, resulting code length $K = N \times L$, the BOLSH will generate a random matrix $H = [v_1, v_2, \ldots, v_K]$ with each element being sampled independently from the normal distribution $\mathcal{N}(0, 1)$. After the orthogonalization, we get the output projection matrix $\hat{H} = [\omega_1, \omega_2, \ldots, \omega_K]$. In fact, while we extracted a 1152 dimensions feature for each face, we set $K = 400, N = 80, L = 5$ in our experiment.

With the projection matrix of BOLSH, all faces detected and tracked from the video need to be projected into the hash space. Nevertheless, a video always has a large number of frames, and also each frame contains several faces. That is, there will be massive faces to be projected, and this will result in very low efficiency. In fact, when the projection matrix is acquired, all the face images are dealt with a series of the same pixel value wise calculations. Intuitively, the BOLSH projection can be done by map/reduce processes in a cloud computing framework.

More specifically, the map/reduce functions in the Spark computing framework are used to do the BOLSH projection for all face images in parallel. Actually, the following pseudo-program implements the BOLSH projection for all face images in parallel.

1) val faces = sc.parallelize(list(faces detected and tracked in the video))
2) val f=f.mapValues($x => (\hat{H} dot x)$)

where $\hat{H}$ is the projection vector for BOLSH. Actually, these vectors are generated randomly, and also been grouped and orthogonalized.

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

VII. Kernel-view Multi-Batch Joint Sparse Representation and Classification

Given a set of retrieved gallery face images and the extracted probe face tracks, we present in this section a simple yet efficient algorithm for face track identification. Each unlabeled face track is simply represented as a set of BOLSH projection features by image feature vectors extracted from all images in the track. One simple method for identification is to directly calculate the feature distances between a probe face track and the labeled exemplar faces, and then assign the probe face track to the nearest neighborhood. Another feasible method is to classify each image in the track independently via, e.g., sparse representation classification, and then assign the face track to the subject that achieves the highest frequency.

In this work, by viewing the identification of each image in a probe face track as a task, the face track identification can be naturally casted to a multi-task face recognition problem. This motivates us to apply the multi-task joint sparse representation model [8] for face track classification. The key advantage of multi-task learning lies in that it can efficiently make use of complementary information contained in different sub-tasks. In addition, we also extend the multi-task learning into kernel-view, which is more competitive than the state-of-the-art multiple kernel learning methods for face tracks recognition.

A. Multi-batch Joint Sparse Representation based Recognition

Suppose we have a set of exemplar faces with M subjects. Here, a subject means a person, which refers to a set of the same person's faces. Denote $X^l = [X_1^l, \ldots, X_M^l]$ as the training feature matrix, and $X_m^l \in \mathbb{R}^{d_l \times p_m}$ is associated with the m_{th} subject, where d_l is the dimensionality of the l_{th} batch of the BOLSH hash value, and $p = \sum_{m=1}^{M} p_m$ means the total number of training samples. Here, we consider a supervised L-batch (task) linear representation problem as follows:

$$\boldsymbol{y}^l = \sum_{m=1}^{M} X_m^l \omega_m^l + \varepsilon^l, l = 1, \ldots, L, \tag{13}$$

where $\boldsymbol{y} = \boldsymbol{y}^l$ means one face of a face track and $\boldsymbol{y}^l$ as a batch (task) is the l_{th} batch of each face image's BOLSH hash value in this track. Meanwhile, $\omega_m^l \in \mathbb{R}^{p_m}$ is a reconstruction coefficient vector associated with the m_{th} subject, and ε^l is as the residual term. Denote $\omega^l = [(\omega_1^l)^T, \ldots, (\omega_M^l)^T]^T$ the representation coefficients in batch l, and $w_m = [\omega_m^1, \ldots, \omega_m^L]$ the representation coefficients from the m-th subject across different batches (tasks). Furthermore, we denote $W = [\omega_m^l]_{m,l}$. Therefore, our proposed multi-task joint sparse representation model is formulated as the solution to the following problem of multi-task least square regressions with $\ell_{1,2}$ mixed-norm regularization:

$$\min_W F(W) = \frac{1}{2} \sum_{l=1}^{L} \left\| \boldsymbol{y}^l - \sum_{m=1}^{M} X_m^l \omega_m^l \right\|_2^2 + \lambda \sum_{m=1}^{M} \| \omega_m \|_2 . \tag{14}$$

Here, to optimize the model, the *accelerated proximal gradient* [8] is adopted to solve the Eqn. (14) with fast convergence rate guaranteed. The *accelerated proximal gradient* is composed by a weight matrix sequence $\hat{W}^t = [\omega_m^{l,t}]_{t \geq 1}$, and an aggregation matrix sequence $\hat{V}^t = [v_m^{l,t}]_{t \geq 1}$. The $\hat{W}^{t+1}$ is updated according to the result,

$$\hat{\omega}^{l,t+1} = \hat{v}^{l,t} - \eta \nabla^{l,t}, l = 1, \ldots, L, \tag{15}$$

$$\hat{\omega}_m^{t+1} = [1 - \frac{\lambda \eta}{\|\hat{\omega}_m^{t+1}\|_2}]_+ \hat{\omega_m^{t+1}}, m = 1, \ldots, M. \tag{16}$$

Here $\nabla^{l,t} = -(X^l)^T y^l + (X^l)^T X^l \hat{v}^{l,t}$, η is the step size parameter, and $[\bullet]_+ = max(\bullet, 0)$. In addition,

$$\hat{V}^{t+1} = \hat{W}^{t+1} + \frac{\alpha_{t+1}(1 - \alpha_t)}{\alpha_t} (\hat{W}^{t+1} - \hat{W}^t), \tag{17}$$

where α_t is directly set as $2/(t+2)$[8] in our approach.

With the *accelerated proximal gradient* algorithm, we obtained the optimal $\hat{W} = [\hat{\omega}_m^l]$, where $\hat{\omega_m^l}$ associated with the l_{th} task (batch) in the m_{th} subject. The l_{th} batch y^l of each face image f_j in a face track can be approximated as $y^l = X_m^l \hat{\omega_m^l}$. For classification and recognition, the decision is ruled in favor of the subject with the lowest total reconstruction error accumulated over all the L batches:

$$m_j^* = \arg\max_m \sum_{l=1}^{L} \theta^l \left\| y^l - X_m^l \hat{\omega}_m^l \right\|_2^2, \tag{18}$$

where $\theta^l{}_{l=1}^{L} (\sum_l \theta^l = 1)$ are the weights that measure the confidence of different batches in final decision.

There are tens of faces in each face track, and each of the face have assigned a subject label with Equ. 18. After, the whole face track is recognized with an unified subject by,

$$m^* = \arg\max_m \sum_{j=1}^{J} [\![m_j^* == m]\!]. \tag{19}$$

We call the model (14) along with classification rule (18 and 19) as the Multi-Batches Joint Sparse Representation and Classification (MBJSRC) in this paper.

B. The Kernel View Extensions Recognition

Heretofore, the face track identification is feasibly realized by the MBJSRC algorithm for sparse representation and classification. In order to combine multiple feature kernels for face track recognition, we extend the MBJSRC algorithm to the kernel version as described in [8].

For a *Reproducing Kernel Hilbert Space*, the kernel trick is to use a non-linear function $\phi^l(x_i)^T \phi^l(x_j) = g^l(x_i, x_j)$ for some given kernel function g^l. Let $G^l = \phi^l(X^l)^T \phi^l(X^l)$ be the training kernel matrix associated with the l_{th} modality of the feature, and $h^l = \phi^l(X^l)\phi^l(y^l)$ be the test kernel vector associated with the l_{th} modality. In our approach, the simple and available kernel matrix is constructed by directly using vector h^l and the column of each kernel matrix G^l as the extracted new features. In this new space, the original multi-task least square regressions with $\ell_{1,2}$ mixed-norm regularization problem can be written as:

$$\min_W F(W) = \frac{1}{2} \sum_{l=1}^{L} \left\| h^l - \sum_{m=1}^{M} G_m^l \omega_m^l \right\|_2^2 + \lambda \sum_{m=1}^{M} \|\omega_m\|_2. \tag{20}$$

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

Actually, in the experiment, the kernel matrices are computed as $\exp(-\chi^2(x, x^{'})\mu)$, and μ is set to be the mean value of the pairwise χ^2 distance on the training set.

VIII. Experimental Results

We conduct extensive experiments to evaluate the efficiency and effectiveness of the proposed cloud-based actor identification with BOLSH and sparse representation. This section is organized as follows: Subsection VIII-A introduces the details of construction of the used dataset. Subsection VIII-B demonstrates the efficiency of cloud-based shot boundary detection and the cloud-based BOLSH. VIII-C details the effectiveness of our approach with different settings in BOLSH. Meanwhile, Subsection VIII-D shows a naive approach of the Sparse Representation (SR) classifier, and also we demonstrate the performance comparison among our approach, the Nearest Neighbor (NN) and the SR classifier as well as the SVM classifier.

A. Dataset Construction

Since we mainly test our approach on a movies("Titanic" (1997)) and a TV series of *The Big Bang Theory, episode 1-5 from season 2*, we constructed our dataset from image data and video data as follows,

- **Gallery Dataset**: we select 8 actors who are the main actors in our selected movie and TV series, namely, characters of *Rose, Jack, Caledon, Leonard, Sheldon, Penny, Howard and Rajesh*. For character of each actor, we first retrieve related face images from *Google Image* and *Bing Image* respectively using the names of actors as query. Then, the above mentioned OKAO face detector are applied on the images returned by the research engine. And, totally 100 face images are added to the Gallery Dataset with the name of the actors as the label. Actually, the 100 face images means the first 50 faces from both the query results from *Google Image* and *Bing Image* respectively. Therefore, finally, there are 8×100 face images in our Gallery Dataset.
- **Video Data**: A video corpus consisting of 1 movie and 5 episodes of TV series is downloaded from the Internet. The resolution of these videos are 1280×720, and also and the frame rate is about 25fps.

Meanwhile, by only considering the detected tracks, the volume of frames that need to be processed can be largely reduced to accelerate the classification process. With the cloud-based shot boundary detection, each video is segmented into several shots, as shown in Table I (*The Bigbang* $1 \sim 5$ refer to episode $1 \sim 5$ of *The Big Bang Theory*, season 2).

After the video is segmented into shots, the tracking process takes the results of OKAO face detection[5] as input, and generates several face tracks using the tracking algorithm in [49]. Then, a nine-point SIFT feature is used in the experiments, namely, to extract face features from the exemplar faces and face tracks. Referring to the work of Everingham *et al.* [9], a generative model is adopted to locate the nine facial key-points in the detected face region, including the left and right corners of each eye, the two nostrils and the tip of the nose and the left and right corners of the mouth followed by 128-dim SIFT feature extraction process.

[5]http://www.omron.com/r_d/coretech/vision/okao.html

TABLE II: Efficiency of Cloud Based Approach

Videos	#Shot	Cost Time (s)	
		Our Approach	Method [48]
Titanic	1550	**1153**	3773
The Bigbang 1	374	**372**	1151
The Bigbang 2	387	**501**	1214
The Bigbang 3	380	**447**	1098
The Bigbang 4	368	**398**	1133
The Bigbang 5	394	**426**	1208

B. Efficiency of Cloud-based Approaches

In this subsection, we illustrate the efficiency of two cloud-based processes, namely, the cloud-based shot boundary detection and the cloud-based BOLSH. The hardware environment of the system includes the video storage and processing center: Intel $Core^{TM}$ 2 Quad Q9550 CPU, 2.83 GHz (4-kernel) frequency, 12G memory. For the cloud-based shot boundary detection as well as the cloud-BOLSH hashing, we used totally 4 nodes, including one physical machine and 3 virtual machine, and each machine has 2G memory. Firstly, we do efficiency comparison with the approach with cloud computing ideas, and the accelerating shot boundary detection methods in [48], which is shown in Table II.

In addition, in order to analyze the efficiency of our cloud-based BOLSH hash method, we manually chose 149 face tracks with 11692 face images in the movie "Titanic" and 254 face tracks with 20311 face images in the TV series of "The Big Bang Theory". Meanwhile, we extracted a 1152 dimensions feature vector for each face. That is, we evaluate our cloud-based approach on totally 32003 face images. Using the projection matrix $\hat{H}$ generated in VI-A, we project the sample matrix $X \in \mathcal{R}^{1152\times 32003}$ into a hashing matrix $X_h \in \mathcal{R}^{400\times 32003}$ with $K = 400, N = 80, L = 5$.

Since all these faces can be hashed with the same processes, we used the cloud-based framework to assign the data in HDFS database and also the projection processes into different computing nodes. With the cloud computing ideas, all the face images will be projected in parallel. Since all the projection tasks are distributed to different virtual machines, our cloud-based BOLSH hashing has obviously achieved more high efficiency compare to the original BOLSH.

C. Recognition Performance of the BOLSH and MBJSRC

While we combined the characteristic of 'batch' in BOLSH with the ideas of 'task' in MTJSRC, our approach achieved more satisfactory performance in the recognition effectiveness and accuracy. As shown in Table III, we evaluate the recognition accuracy of BOLSH combined with MTJSRC with different setting of K, L and N in BOLSH.

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

TABLE I: summary of test movies

Movies	Duration(min)	Resolution	#Shot	Genres
Titanic	195	1080×720	1550	Drama&Romance
The Bigbang 1	22.3	1080×720	374	Comedy
The Bigbang 2	21.8	1080×720	387	Comedy
The Bigbang 3	22.1	1080×720	380	Comedy
The Bigbang 4	22.5	1080×720	368	Comedy
The Bigbang 5	21.8	1080×720	394	Comedy

TABLE III: Recognition Performance of BOLSH combined with MBJSRC

Videos	#FaceTracks	Parma. of BOLSH		
		N=1152,L=1	**N=80,L=5**	N=100,L=8
Titanic	170	80.6%	**83.5%**	78.2%
The Bigbang 1	84	90.5%	**92.9%**	88.2%
The Bigbang 2	71	87.3%	**87.3%**	85.9%
The Bigbang 3	80	91.2%	**93.7%**	88.7%
The Bigbang 4	97	88.6%	**90.7%**	88.7%
The Bigbang 5	75	88.0%	**93.3%**	88.0%

D. Performance Comparisons with Different Approaches

Three baseline methods are employed for comparison: i) the nearest neighbor (NN) classifier used in [9] which directly calculates the feature distances between a probe face track and the labeled exemplar faces, and then assigns the probe face track to the nearest neighborhood; ii) the sparse representation(SR) classifier [50]; and iii) the SVM classifier. For the SR and SVM methods, they classify each image in the track independently and then assign the face track to the subject that most frequently occurs in this track. In addition, for SR algorithm in [50], we give some details about how to use it in our track level face recognition.

Suppose the matrix $X = \{X_m\}$ for the entire gallery set is the concatenation of the $p = \sum_{i=1}^{m} p_m$ training samples of all M subject classes. Denote $X_m = [\upsilon_{m,1}, \upsilon_{m,2}, \ldots, \upsilon_{m,p_m}] \in \mathbb{R}^{d \times p_m}$ as the m_{th} subject samples. For a new (test) face track $\boldsymbol{y}$ with K face images, we first classify the k_{th} face into the class $c_k \in \{1, \ldots, M\}$, and also define $C = [c_1, \ldots, c_K]$ as the class vector for the test face track. Then, we assign $c = \arg\max_m \|C - m\|_0$, which means the most frequently occurred subject class, as the final subject class for the test track. Meanwhile, the class label c_k of the k_{th} face in the track is obtained as follows.

$\boldsymbol{y_k}$ is the k_{th} face in the face track, and represented as,

$$\boldsymbol{y_k} = X\boldsymbol{\alpha} + \boldsymbol{e}, \tag{21}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^p$ is the coefficient vector. Then, to get the informative vector $\boldsymbol{\alpha} = [\alpha_1^T, \ldots, \alpha_M^T]^T$ is equivalent to the solution of the following ℓ_1-minimization problem,

$$\hat{\boldsymbol{\alpha}}_{\boldsymbol{1}} = \arg\min \|\alpha\|_1 \quad subject\ to \quad \boldsymbol{y_k} = X\boldsymbol{\alpha} + \boldsymbol{e}. \tag{22}$$

That is, to solve the following problem,

$$\min_{\boldsymbol{\alpha}} F(\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{y_k} - X\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1. \tag{23}$$

This problem can be solved in polynomial time by standard linear programming methods [51]. After, we classify $\boldsymbol{y_k}$ to the subject class that minimizes the residual between $\boldsymbol{y_k}$ and $\hat{\boldsymbol{y_{k_m}}}$:

$$c_k = \arg\min_m \|\boldsymbol{y_k} - X_m\boldsymbol{\alpha_m}\|_2. \tag{24}$$

There always exist a few incorrect faces in the gallery set, and thus training based methods, e.g., SVM and Subspace analysis, are not applicable in our setting, as shown in Table IV. In contrast, our multi-task linear representation based method is quite robust for the condemnation since the joint representation ability of noise images is lower compared with those "good" samples.

Using $N = 80, L = 5$ for BOLSH, the evaluation results are listed in Table IV, from which we can see that our approach significantly outperforms both baselines. In our experiment, the adopted *accelerated proximal gradient* algorithm converges at roughly $10 \sim 20$ rounds of iterations. The average running time is $0.31s$ per probe face track. The parameter λ in (5) is set to 0.1 throughout our experiment.

IX. Conclusion

With explosive development of social network and video sharing websites, an efficient and accurate way to index and organize videos according to the identities of the involved persons becomes heavily demanded. Meanwhile, querying actor-specific video clips in large scale video dataset has attracted much attentions in both video processing and computer vision

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

TABLE IV: Recognition Performance of Different Approaches

Videos	#FaceTracks	Our Approach	SR Method	NN Method	SVM Classifier
Titanic	170	**83.5%**	81.7%	78.2%	69.5%
The Bigbang 1	84	**92.9%**	91.7%	83.3%	78.9%
The Bigbang 2	71	**87.3%**	85.9%	74.6%	71.6%
The Bigbang 3	80	**93.7%**	88.7%	87.5%	85.1%
The Bigbang 4	97	**90.7%**	90.7%	85.5%	81.7%
The Bigbang 5	75	**93.3%**	86.7%	86.7%	84.2%

research field. Nevertheless, both the effectiveness and efficiency of many existing methods are not so satisfactory. Therefore, in this paper, we propose an efficient cloud-based actor identification approach with Batch-Orthogonal Local-Sensitive Hashing (BOLSH) and Multi-Task Joint Sparse Representation and Classification (MTJSRC) algorithm. More specifically, videos are segmented into shots with the cloud-based shot boundary detection, and also the cloud-based BOLSH is implemented on video faces for feature description. Then, the batches in BOLSH are used as tasks for the Multi-Task Joint Sparse Representation and Classification algorithm for actor identification in each face track. Extensive experiments are implemented to demonstrate the satisfying performance of our approach considering both accuracy and efficiency.

Besides, the *accelerated proximal gradient* algorithm in MTJSRC is a machine learning algorithm which run iterative optimization procedures, to minimize a target function. Therefore, in future, with the Spark programming, it can run much faster by keeping their data in memory. Therefore, as the typical example for Spark programming for logistic regression [52], we can parse the *accelerated proximal gradient* algorithm into fine processes, and assign all these processes into different Spark nodes. Furthermore, with faster processing, we can test more parameters combination to get the most excellent model and parameters.

REFERENCES

[1] X. Ge, X. Huang, Y. Wang, M. Chen, Q. Li, T. Han, and C.-X. Wang, "Energy efficiency optimization for mimo-ofdm mobile multimedia communication systems with qos constraints," *IEEE Trans. on Vehicular Technology*, vol. 63, no. 5, pp. 2127–2138, 2014.

[2] H. Ma, C. Zeng, and C. Ling, "A reliable people counting system via multiple cameras," *ACM Trans. on Intelligent Systems and Technology*, vol. 3, no. 2, p. 31, 2012.

[3] M. Bauml, M. Tapaswi, and R. Stiefelhagen, "Semi-supervised learning with constraints for person identification in multimedia data," in *Proceedings of CVPR*, 2013, pp. 3602–3609.

[4] J. Huang, H. Wang, and Y. Qian, "Game user-oriented multimedia transmission over cognitive radio networks," *IEEE Trans. on Circuits and Systems for Video Technology*, 2016.

[5] J. Sang and C. Xu, "Faceted subtopic retrieval: Exploiting the topic hierarchy via a multi-modal framework," *Journal of Multimedia*, vol. 7, no. 1, pp. 9–20, 2012.

[6] O. Arandjelovic and A. Zisserman, ""who are you?" - learning person specific classifiers from video," in *Proceedings of the CVPR*, 2009, pp. 1145–1152.

[7] J. Ji, S. Yan, J. Li, G. Gao, Q. Tian, and B. Zhang, "Batch-orthogonal locality-sensitive hashing for angular similarity," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 1963–1974, 2014.

[8] X. Yuan and S. Yan, "Visual classification with multi-task joint sparse representation," in *Proceedings of CVPR*, 2010, pp. 3493–3500.

[9] M. Everingham, J. Sivic, and A. Zisserman, "'hello! my name is... buffy' - automatic naming of characters in tv video," in *Proceedings of the 17th British Machine Vision Conference*, 2006, pp. 889–908.

[10] M. Tapaswi, M. Bäuml, and R. Stiefelhagen, "Knock knock who is it probabilistic person identification in tv series," in *Proceedings of CVPR*, 2012, pp. 2658–2665.

[11] C. Xiong, G. Gao, Z. Zha, S. Yan, H. Ma, and T. Kim, "Adaptive learning for celebrity identification with video context," *IEEE Trans. on Multimedia*, vol. 16, no. 5, pp. 1473 – 1485, 2014.

[12] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic, "Finding actors and actions in movies," in *Proceedings of ICCV*, 2013, pp. 2280–2287.

[13] J. Sang and C. Xu, "Robust face-name graph matching for movie character identification," *IEEE Trans. on Multimedia*, vol. 14, no. 3, pp. 586–596, 2012.

[14] X. Zhang, L. Zhang, X.-J. Wang, and H.-Y. Shum, "Finding celebrities in billions of web images," *IEEE Trans. on Multimedia*, vol. 14, no. 4, pp. 995–1007, 2012.

[15] Y. Gao, L. Jin, C. He, and G. Zhou, "Handwriting character recognition as a servicea new handwriting recognition system based on cloud computing," in *Proceedings of 2011 International Conference on Document Analysis and Recognition*, 2011, pp. 885–889.

[16] L. Zhu, X. Zheng, P. Li, and Y. Wang, "A cloud based object recognition platform for ios," in *Proceedings of 2014 International Conference on Identification, Information and Knowledge in the Internet of Things*, 2014, pp. 68–71.

[17] T. Suzuki and T. Ikenaga, "Keypoints of interest based on spatio-temporal feature and mrf for cloud recognition system," in *Proceedings of 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2013, pp. 1–4.

[18] M. S. Hossain, G. Muhammad, M. Alhamid, B. Song, and K. Almutib, "Audio-visual emotion recognition using big data towards 5g," *Mobile Networks and Applications*, Jan. 2016.

[19] H. Wang, S. Wu, M. Chen, and W. Wang, "Security protection between users and mobile media cloud," *IEEE Communication Magazine*, vol. 52, no. 3, pp. 73–79, 2014.

[20] M. S. Hossain and G. Muhammad, "Cloud-assisted speech and face recognition framework for health monitoring," *Mobile Networks and Applications*, vol. 20, no. 3, pp. 391–399, Feb. 2015.

[21] Y. Zhang, D. Zhang, M. M. Hassan, A. Alamri, and L. Peng, "CADRE: Cloud-assisted drug recommendation service for online pharmacies," *ACM/Springer Mobile Networks and Applications*, vol. 20, no. 3, pp. 348–355, 2015.

[22] C. Lai, H. Wang, H. Chao, and G. Nan, "A network and device aware qos approach for cloud-based mobile streaming," *IEEE Trans. on Multimedia*, vol. 15, no. 4, pp. 747–757, 2013.

[23] K. Lin, J. Song, J. Luo, W. Ji, M. Hossain, and A. Ghoneim, "GVT: Green video transmission in the mobile cloud networks," *IEEE Trans. on Circuits and Systems for Video Technology*, 2016.

[24] H. Ma, "Internet of things: Objectives and scientific challenges," *Journal of Computer science and Technology*, vol. 26, no. 6, pp. 919–924, 2011.

[25] H. Ma, L. Liu, A. Zhou, and D. Zhao, "On networking of internet of things: Explorations and challenges," *IEEE Internet of Things Journal*, 2015.

[26] M. S. Hossain and G. Muhammad, "Cloud-assisted industrial internet of things (iiot)-enabled framework for health monitoring," *Computer Networks*, vol. 101, no. (2016), pp. 192–202, June 2016.

[27] Z. Sheng, J. Fan, C. H. Liu, V. C. M. Leung, X. Liu, and K. K. Leung, "Energy-efficient relay selection for cooperative relaying in wireless

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

multimedia networks," *IEEE Trans. on Vehicular Technology*, vol. 64, no. 3, pp. 1156–1170, 2015.

[28] C. H. Liu, K. K. Leung, C. Bisdikian, and J. W. Branch, "A new approach to architecture of sensor networks for mission-oriented applications," in *Proc. of SPIE Defense Security and Sensing*, 2009, pp. 73 490L–73 490L–12.

[29] C. H. Liu, T. He, K. W. Lee, K. K. Leung, and A. Swami, "Dynamic control of data ferries under partial observations," in *IEEE WCNC'10*, 2010, pp. 1–6.

[30] C. H. Liu, J. Fan, J. W. Branch, and K. K. Leung, "Toward qoi and energy-efficiency in internet-of-things sensory environments," *IEEE Trans. on Emerging Topics in Computing*, vol. 2, no. 4, pp. 473–487, 2014.

[31] C. H. Liu, J. Fan, P. Hui, J. Wu, and K. K. Leung, "Toward qoi and energy efficiency in participatory crowdsourcing," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4684–4700, 2015.

[32] C. H. Liu, P. Hui, J. Branch, C. Bisdikian, and B. Yang, "Efficient network management for context-aware participatory sensing," in *IEEE SECON'11*, 2011, pp. 116–124.

[33] C. H. Liu, B. Yang, and T. Liu, "Efficient naming, addressing and profile services in internet-of-things sensory environments," *Elsevier Ad Hoc Networks*, vol. 18, pp. 85–101, 2014.

[34] H. Gao, C. H. Liu, W. Wang, J. Zhao, Z. Song, X. Su, J. Crowcroft, and K. K. Leung, "A survey of incentive mechanisms for participatory sensing," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 2, pp. 918–943, 2015.

[35] C. H. Liu, B. Zhang, X. Su, J. Ma, W. Wang, and K. K. Leung, "Energy-aware participant selection for smartphone-enabled mobile crowd sensing," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2015.

[36] Z. Song, B. Zhang, C. H. Liu, A. Vasilakos, J. Ma, and W. Wang, "Qoi-aware energy-efficient participant selection," in *IEEE SECON'14*, July 2014.

[37] C. H. Liu, J. Wen, Q. Yu, B. Yang, and W. Wang, "Healthkiosk: A family-based connected healthcare system for long-term monitoring," in *IEEE INFOCOM'11 Workshops*, 2011, pp. 241–246.

[38] C. H. Liu, J. Zhao, H. Zhang, S. Guo, K. K. Leung, and J. Crowcroft, "Energy efficient event detection by participatory sensing under budget constraints," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2016.

[39] B. Zhang, Z. Song, C. H. Liu, J. Ma, and W. Wang, "An event-driven qoi-aware participatory sensing framework with energy and budget constraints," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 3, p. 42, 2015.

[40] B. Zhang, C. H. Liu, J. Lu, Z. Song, Z. Ren, J. Ma, and W. Wang, "Privacy-preserving qoi-aware participant coordination for mobile crowdsourcing," *Elsevier Computer Networks*, vol. 101, pp. 29–41, 2016.

[41] O. Yurur, C. H. Liu, and W. Moreno, "Unsupervised posture detection by smartphone accelerometer," *Electronics Letters*, vol. 49, no. 8, pp. 562–564, 2013.

[42] C. H. Liu, K. K. Leung, and A. Gkelias, "A generic admission-control methodology for packet networks," *IEEE Trans. on Wireless Communications*, vol. 13, no. 2, pp. 604–617, 2014.

[43] A. Gkelias, F. Boccardi, C. H. Liu, and K. K. Leung, "Mimo routing with qos provisioning," in *IEEE ISWPC'08*, 2008, pp. 46–50.

[44] L. Liu, X. Zhang, and H. Ma, "Localization-oriented coverage in wireless camera sensor networks," *IEEE Trans. on Wireless Communications*, vol. 10, no. 2, pp. 484–494, 2011.

[45] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang, and Y. Pan, "Stochastic load balancing for virtual resource management in datacenters," *IEEE Trans. on Cloud Computing*, 2015.

[46] L. Yu and Z. Cai, "Dynamic scaling of virtualized networks with bandwidth guarantees in cloud datacenters," in *IEEE INFOCOM'16*, 2016.

[47] L. Liu, X. Zhang, and H. Ma, "Optimal node selection for target localization in wireless camera sensor networks," *IEEE Trans. on Vehicular Technology*, vol. 59, no. 7, pp. 3562–3576, 2010.

[48] G. Gao and H. Ma, "To accelerate shot boundary detection by reducing detection region and scope," *Multimedia Tools and Applications*, vol. 71, no. 3, pp. 1749–1770, 2014.

[49] S. Wang, H. Lu, F. Yang, and M. Yang, "Superpixel tracking," in *Proceedings of CVPR*, 2011, pp. 1323–1330.

[50] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–226, 2009.

[51] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[52] M. Zaharia, M. Chowdhury, T. Das, A. Dave, and J. Ma, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, 2012.

Guangyu Gao (M'15) is an Assistant Professor with the School of Software, Beijing Institute of Technology, Beijing China. He received his Ph.D. degree in Computer Science and Technology from Beijing University of Posts and Telecommunications (BUPT) in 2013, and M.S. degree in Computer Science and Technology from Zhengzhou University, Zhengzhou, China in 2007. He also spent about one year at National University of Singapore, Singapore as a government sponsored joint Ph.D. student from July 2012 to Apr. 2013. His current research interests include multimedia, computer vision, video analysis, machine learning, and big data. He is a member of IEEE.

Chi Harold Liu (M'10-SM'15) receives his Ph.D. degree in Electronic Engineering from Imperial College, London, U.K., in 2010, and his B.Eng. degree in Electronic and Information Engineering from Tsinghua University, Beijing, China, in 2006, respectively.

He is currently a Full Professor and serving as the Vice Dean at the School of Software, Beijing Institute of Technology, Beijing, China. From 2014, he also serves as the Director of Data Science Institute, the Director of IBM Mainframe Excellence Center (Beijing), the Director of IBM Big Data and Analysis Technology Center, and the Director of National Laboratory of Data Intelligence for China Light Industry. Before moving to academia, he worked for IBM T. J. Watson Research Center and IBM Research - China as a staff researcher and project manager from 2010 to 2013, worked as a postdoctoral researcher at Deutsche Telekom Laboratories, Germany in 2010. His current research interests include the Internet-of-Things (IoT), big data analytics, mobile computing, and wireless ad hoc, sensor, and mesh networks.

Prof. Liu was elected into the "High-Level Overseas Talents Return Home Program", by Ministry of Human Resource and Social Security, China, in 2015, received the Distinguished Young Scholar Award from Beijing Institute of Technology in 2013, received IBM First Plateau Invention Achievement Award in 2012, received IBM First Patent Application Award in 2011, and was interviewed by EEWeb.com as the Featured Engineer in 2011. He has published more than 80 prestigious conference and journal papers and owned more than 10 EU/U.S./China patents. He serves as the editor for KSII Trans. on Internet and Information Systems from 2013, and the book editor for six books published by Taylor & Francis Group, USA and Wiley. He also has served as the general chair of IEEE SECON'13 workshop on IoT Networking and Control, IEEE WCNC'12 workshop on IoT Enabling Technologies, and ACM UbiComp'11 Workshop on Networking and Object Memories for IoT. He served as the consultant to the Asian Development Bank, Bain & Company, and KPMG, USA, and the peer reviewer for Qatar National Research Foundation, Qatar, and National Science Foundation, China. He also serves as the Lead Guest Editor for IEEE SENSORS JOURNAL, Special Issue on Software Defined Wireless Sensor Networks, and Guest Editor for IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, Special Issue on Sensor Data Computing as a Service in Internet of Things. He is a Senior Member of Chinese Institute of Electronics, a Senior Member of IEEE, and a member of ACM. For more information, please visit: http://haroldliu.weebly.com

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2579305, IEEE Transactions on Multimedia

Min Chen (M'08-SM'09) is a professor in School of Computer Science and Technology at Huazhong University of Science and Technology (HUST). He is Chair of IEEE Computer Society (CS) Special Technical Communities (STC) on Big Data. He was an assistant professor in School of Computer Science and Engineering at Seoul National University (SNU) from Sep. 2009 to Feb. 2012. He worked as a Post-Doctoral Fellow in Department of Electrical and Computer Engineering at University of British Columbia (UBC) for three years. Before joining UBC, he was a Post-Doctoral Fellow at SNU for one and half years. He received Best Paper Awardfrom IEEE ICC 2012, and Best Paper Runner-up Award from QShine 2008. He serves as editor or associate editor for Information Sciences, Wireless Communications and Mobile Computing, IET Communications, IET Networks, Wiley I. J. of Security and Communication Networks, Journal of Internet Technology, KSII Trans. Internet and Information Systems, International Journal of Sensor Networks. He is managing editor for IJAACS and IJART. He is a Guest Editor for IEEE Network, IEEE Wireless Communications Magazine, etc. He is Co-Chair of IEEE ICC 2012-Communications Theory Symposium, and Co-Chair of IEEE ICC 2013-Wireless Networks Symposium. He is General Co-Chair for the 12th IEEE International Conference on Computer and Information Technology (IEEE CIT-2012) and Mobimedia 2015. He is General Vice Chair foTridentcom 2014. He is Keynote Speaker for CyberC 2012, Mobiquitous 2012 and Cloudcomp 2015. He has more than 260 paper publications, including 120+ SCI papers, 50+ IEEE Trans./Journal papers, 8 ISI highly cited papers and 1 hot paper. He has published two books: OPNET IoT Simulation (2015) and Big Data Inspiration (2015) with HUST Presss, and a book on big data: Big Data Related Technologies (2014) with Springer Series in Computer Science. His Google Scholars Citations reached 6,200+ with an h-index of 38. His top paper was cited 740+ times, while his top book was cited 480 times as of June 2016. He is an IEEE Senior Member since 2009. His research focuses on Cyber Physical Systems, IoT Sensing, 5G Networks, Mobile Cloud Computing, SDN, Healthcare Big Data, Medica Cloud Privacy and Security, Body Area Networks, Emotion Communications and Robotics, etc.

Song Guo (M'02-SM'11) received the PhD degree in computer science from the University of Ottawa, Canada. He is currently a full professor at the University of Aizu, Japan. His research interests are mainly in the areas of wireless network, cloud computing, big data, and cyber-physical system. He has authored/edited 7 books and over 300 papers in refereed journals and conferences in these areas. He serves/served in editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Emerging Topics in Computing, IEEE Communications Magazine, Wireless Networks, Wireless Communications and Mobile Computing, and many other major journals. He has been the general/program chair or in organizing committees of numerous international conferences. Dr. Guo is a senior member of IEEE, a senior member of ACM, and an IEEE Communications Society Distinguished Lecturer.

Kin K. Leung (F'01) received his B.S. degree from the Chinese University of Hong Kong in 1980, and his M.S. and Ph.D. degrees from University of California, Los Angeles, in 1982 and 1985, respectively.

He joined AT&T Bell Labs in New Jersey in 1986 and worked at its successor companies, AT&T Labs and Bell Labs of Lucent Technologies, until 2004. Since then, he has been the Tanaka Chair Professor in the Electrical and Electronic Engineering (EEE), and Computing Departments at Imperial College in London. He serves as the Head of Communications and Signal Processing Group in the EEE Department at Imperial. His research interests focus on networking, protocols, optimization and modeling issues of wireless broadband, sensor and ad-hoc networks. He also works on multi-antenna and cross-layer designs for the physical layer of these networks.

He received the Distinguished Member of Technical Staff Award from AT&T Bell Labs in 1994, and was a co-recipient of the 1997 Lanchester Prize Honorable Mention Award. He was elected as an IEEE Fellow in 2001. He received the Royal Society Wolfson Research Merits Award from 2004 to 2009 and became a member of Academia Europaea in 2012. Along with his co-authors, he also received several best paper awards at major conferences. He has actively served on many conference committees. He serves as a member (2009-11) and the chairman (2012-15) of the IEEE Fellow Evaluation Committee for Communications Society. He was a guest editor for the IEEE Journal on Selected Areas in Communications (JSAC), IEEE Wireless Communications and the MONET journal, and as an editor for the JSAC: Wireless Series, IEEE Transactions on Wireless Communications and IEEE Transactions on Communications. Currently, he is an editor for the ACM Computing Survey and International Journal on Sensor Networks.

1520-9210 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.