MULTIMEDIA COMMUNICATIONS TECHNICAL COMMITTEE
http://www.comsoc.org/~mmc

MMTC Communications - Frontiers

Vol. 16, No. 2, March 2021

CONTENTS

SPECIAL ISSUE ON the Multimedia Data Processing Techniques and Their Applications

Guest Editor: Dapeng Wu
Chongqing University of Posts and Telecommunications, China
wudp@cqupt.edu.cn

This special issue of Frontiers focuses on the multimedia data processing techniques and their applications in various domains. The research topics of the papers in this special issue include how to estimate the synthesizability of dynamic textures, an end-to-end tooth segmentation mechanism, a contact-free breath tracking system, and a qualitative survey of different types of graph databases.

The first paper proposed propose a tractable solution for estimating the synthesizability of dynamic textures by learning regression models with appropriate spatiotemporal descriptors. The proposed system also suggests the "best" synthesis method from off-the-shelf algorithms. The proposed work investigates the "dynamic textureness" property of dynamic patterns in the video to automatically discern dynamic textures and divide dynamic textures into two subcategories based on the spatial modes to suit different synthesis capacity.

The second paper propose an end-to-end tooth segmentation mechanism which leverages a two-stream graph convolutional network to learn, from heterogenous multi-view inputs, the discriminative geometric features. The proposed system consists of a C stream and an N-stream, which will learn multi-scale features independently, from coordinates and normal vectors, respectively. The complementary single-view representations will be combined by a fusion branch, which aims at minimizing the inter-view confusion.

The third paper presented a contact-free breath tracking system using the off-the-shelf WiFi devices to track the human breath. The system leverages the phase variation of the CSI information to achieve the tracking human breath. Both hardware and software corrections are applied to address the phase distortions in received data to obtain accurate phase information. For example, the work first calibrates the time-invariant PPL Phase Offset by the hardware and then use software to remove the time-varying carrier frequency offset, sampling frequency offset and packet detection delay offset.

The fourth paper provided a qualitative survey of different types of graph databases and compare them with traditional database systems. In this work, a general overview and taxonomy of existing graph database systems are first provided, followed by an outline of the existing framework for graph database management. It then discusses in detail the power and limitation of graph databases, together with the basic requirements in modeling of graph databases. In addition, the challenges of handling and processing graph databases in present scalable and huge data computational applications are also presented.

Dapeng Wu is currently a professor at the Chongqing University of Posts and Telecommunications, Chongqing, China. He authored more than 100 publications and two books. His research interests are in social computing, wireless networks, and big data. Prof. Wu serves as TPC Chair of 10th Mobimedia and program committee member for numerous international conferences and workshops. He served or is serving as an Editor or/and Guest Editor for several technical journals, such as IEEE IoT, Elsevier Digital Communications and Networks, ACM/Springer Mobile Network and Applications. He is a senior member of the IEEE.

Learning the Synthesizability of Dynamic Texture Samples

Feng Yang1, Gui-Song Xia2, Dengxin Dai3, Liangpei Zhang2
1 School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065,China
2 State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, 430079, China
3 Computer Vision Laboratory, ETH Zurich, 8092 Zurich, Switzerland
yangfeng@cqupt.edu.cn, guisong.xia@whu.edu.cn, dai@vision.ee.ethz.ch, zlp62@whu.edu.cn

1. Introduction

Given a dynamic texture sample, often a sequence of images, exemplar-based dynamic texture synthesis (EDTS) is targeted to produce video sequences that look perceptually equivalent to the input, with similar appearance and dynamics. In the past decades, numerous EDTS methods have been proposed, but none of them has been able to address all kinds of dynamic textures equally well. The main reason is because generated results are subject to not only the synthesis algorithm but also the input exemplar, where the former promises technical support and the latter provides the material to be simulated. In this paper, we are interested in the synthesizability of dynamic texture samples: given a dynamic texture sample, what is its possibility of being synthesized by EDTS methods, and which EDTS algorithm is most suited to completing the task?

Although it is intuitive that some videos will be easier to synthesize than others, quantifying this intuition has not been addressed in previous studies. There are no previous investigations that try to quantify individual dynamic textures (DTs) in terms of how synthesizable they are, and there are few computer vision systems to predict synthesizability and suggest suitable synthesized methods for dynamic texture (DT) samples. Additionally, there are no databases of videos calibrated in terms of the degree of synthesizability for each dynamic texture. Motivated by these issues, this paper investigates the synthesizability of dynamic texture samples via a learning scheme. Our work is distinguished in the following aspects:

- We propose a tractable solution for estimating the synthesizability of dynamic textures by learning regression models with appropriate spatiotemporal descriptors. At the same time, for a given DT sample, our system also suggests the "best" synthesis method from off-the-shelf EDTS algorithms.
- To the best of our knowledge, we are the first to investigate the "dynamic textureness" property of dynamic patterns in the video to automatically discern dynamic textures and divide dynamic textures into two subcategories based on the spatial modes to suit different synthesis capacity.
- We propose a novel SCOP-DT descriptor for dynamic texture representation, which captures geometrical aspects and temporal consistency simultaneously.

2. Related Work

The synthesizability aims at helping to identify good samples and suggesting suitable EDTS methods. Recent years have witnessed significant progress in exemplar-based texture synthesis algorithms, roughly divided into two main categories of approaches: parametric and nonparametric methods. Parametric methods usually define a parametric model consisting of a set of statistical measurements that cover the spatial extent and temporal domain of dynamic textures, e.g., the spatiotemporal autoregressive (STAR) model [1], linear dynamic system (LDS) model [2] and its variants [3], [4], [5]. Parametric methods build explicit models with a mathematical foundation, but the main challenge lies in designing rigid and meaningful mathematical models that can capture the essence of different dynamic patterns. Due to incomplete or unenforced statistics, parametric models often fail to synthesize complex geometric patterns. Rather than build explicit models with estimated parameters, nonparametric methods often bypass modeling the spatial-temporal mathematical mechanism of dynamic texture and include copy-based methods and feature-oriented synthesis. Copy-based methods produce new dynamic textures by resampling small parts from an input sample as elements to synthesize in the spatiotemporal domain [6], [7] or along time [8]. Though efficient and visual results are strikingly good, copy-based methods likely lead to verbatim reproduction, similar to a mere copy-paste. Methods of feature-oriented synthesis match the statistics of features between synthetic videos and original videos [9], [10]. This subgroup of methods does not generate verbatim repetition and instead carefully chooses or designs statistics and spatiotemporal features.

3. Method to Learn Dynamic Texture Synthesizability

In this paper, we firstly formulate the problem of learning dynamic texture synthesizability, and then investigate the dynamic texture representation relevant to synthesizability. Next, the learning method is depicted.

Problem formulation

Denote $V \in \mathbb{R}^{U\times d}$ as a video with d channels defined in the space-time domain $U = \{0, \dots, H-1\} \times \{0, \dots, W-1\} \times \{0, \dots, T-1\}$. In particular, $d = 1$ for grayscale videos and $d = 3$ for color videos. The synthesizability $s \in [0,1]$ of a DT sample V indicates the probability that EDTS methods will produce good synthesized results for V before using any synthesis method. A DT sample V with synthesizability score s is denoted as (V, s).

Dynamic texture representation

For dynamic texture representations relevant to synthesizability, we revisit general spatial-temporal features and design a novel SCOP-DT descriptor for dynamic textures. We extend the SCOP descriptor from the image space to the space-time dynamic texture by implicitly incorporating temporal cues and propose a method that implicitly captures the self-similarity behavior of the DT sequence along the time axis, referred to as the SCOP-DT descriptor/feature in this paper.

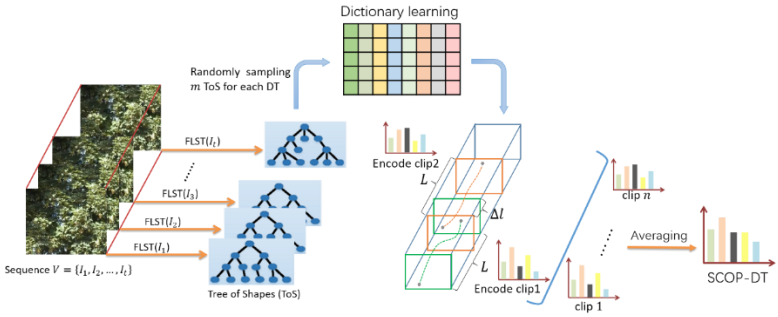


Fig. 1: The proposed representation of shape-based cooccurrence patterns for dynamic textures (SCOP-DT).

The extraction of SCOP-DT in the video is illustrated in Fig. 1. We use fast level set transformation (FLST) to calculate the tree of shapes (ToS) for each frame in the DT video. Due to self-similarities between frame images in dynamic textures, the extracted ToS of each frame is also similar to the other frames. To construct dictionaries of shape cooccurrence patterns in ToSs, we randomly choose m ToSs of m frames from every training video to learn the codewords in the dictionaries. In the encoding stage, based on the temporal consistency in the DTs, we implicitly encode temporal information to incorporate both shape and dynamic aspects for joint spatial-temporal representation.

Learning the synthesizability of dynamic texture samples

a) Learning synthesizability: We suppose that the synthesizability score is learnable and predictable and can be formulated as a regression problem. Given the DT samples and labeled synthesizability scores $\{(V_i, s_i)\}_{i=1}^{N}$ in the training set, we formalize the learning problem as a regression model f:

$$s_i = f(\mathbf{F} \circ V_i) \tag{1}$$

Let the DT sample V be described by a set of K features $\{\mathbf{x}^k\}_{k=1}^{K}$,

$$\forall 1 \le k \le K,\ \mathbf{x}^k = \mathbf{F}_k \circ V, \tag{2}$$

where $\mathbf{F}_k$ is the feature extractor, e.g., SCOP-DT, C3D and LBP-TOP.

Using the feature representation of the DT sample V_i in (2), we denote the regression model as

$$f_k(\mathbf{x}_i^k) = \mathbf{w}_k^T \phi(\mathbf{x}_i^k) + b_k, \quad f_k(\mathbf{x}_i^k) = s_i, \tag{3}$$

where $\mathbf{w}_k$ and b_k are parameters of regression model f_k corresponding to feature representation $\mathbf{F}_k$, with T denoting the vector transpose, and ϕ is the linear or nonlinear mapping.

To build the regression model (3), we train the model on the training set $\{(\{\mathbf{x}_i^k\}_{k=1}^{K}, s_i)\}_{i=1}^{N}$ to determine parameters $\mathbf{w}_k$ and b_k by

$$(\mathbf{w}_k, b_k) = arg \min_{(\mathbf{w}_k, b_k)} \sum_{i=1}^{N} (f_k(\mathbf{x}_i^k) - s_i)^2. \tag{4}$$

Then, for testing the DT sample V_{test} with an unknown synthesizability score, we use the trained regression model (3) to predict the synthesizability score s_{test}^k with feature representation $\mathbf{F}_k$. The predicted synthesizability score $s_{test} \in [0,1]$ is a computable index to quantify how well a dynamic

texture can be synthesized by only analyzing the original sample, where the larger the score is, the better the synthesizability.

b) Predicting synthesizability: We select the optimal regression model relevant to each feature, e.g., support vector machine (SVM) or random forest (RF). Regression models on the training data with labeled synthesizability scores are trained for each feature respectively as depicted in the formulation (3). The trained models are then used to predict the synthesizability of a given video. Thus, for a video, K synthesizability scores can be predicted with respect to K features. Then, the weighted average of the K scores is calculated to form a final synthesizability prediction score. The weights are set manually according to the performance of each individual feature for the effective combination. The final output prediction score s is a weighted average of s^k given the K different features of the unknown test sample V_{test}:

$$s(V_{test}) = \sum_{k=1}^{K} \alpha_k f_k(\mathbf{F}_k \circ V_{test}), \tag{5}$$

where α_k are weighting factors $w.r.t.$ K features C3D, LBPTOP and SCOP-DT.

The scheme of learning and predicting synthesizability by aggregating features is illustrated in Fig. 2. Firstly, we use regression models SVM and RF for every single feature to predict synthesizability, and compare the performance of two regression models. Secondly, we choose the optimal feature and regression model among them, and set weights manually for feature combination in formula (5). Finally, the predicted synthesizability scores of feature combination on decision level are output.

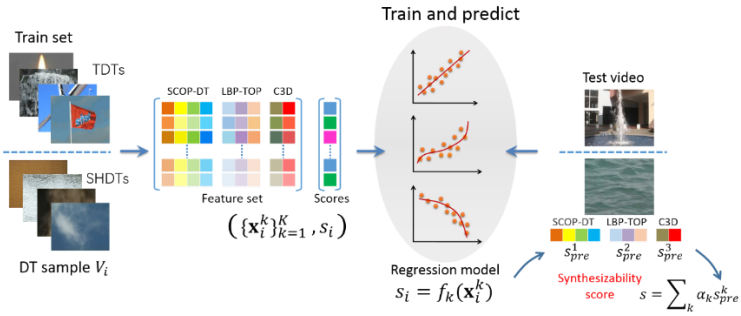


Fig. 2: Learning and predicting synthesizability by aggregating features to train regression models.

4. Conclusions

This paper investigated the synthesizability of dynamic texture samples via a learning scheme. We solved the learning problem by regression model with the proposed SCOP-DT descriptor and other spatiotemporal features. We constructed a dynamic texture dataset related to synthesizability. Experiments verified that the method to predict dynamic texture synthesizability is effective. It is helpful to find good DT examples for synthesis, and to choose an appropriate EDTS method for synthesis.

References

[1] M. Szummer and R. W. Picard, “Temporal texture modeling,” in Proc. IEEE Int. Conf. Image Process., vol. 3, 1996, pp. 823–826.
[2] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, “Dynamic textures,” Int. J. Comput. Vis., vol. 51, no. 2, pp. 91–109, 2003.
[3] L. Yuan, F. Wen, C. Liu, and H.-Y. Shum, “Synthesizing dynamic texture with closed-loop linear dynamic system,” in Proc. Eur. Conf. Comput. Vis., 2004, pp. 603–616.
[4] F. Woolfe and A. Fitzgibbon, “Shift-invariant dynamic texture recognition,” in Proc. Eur. Conf. Comput. Vis., 2006, pp. 549–562.
[5] A. B. Chan and N. Vasconcelos, “Classifying video with kernel dynamic textures,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2007, pp. 1–6.
[6] V. Kwatra, A. Sch¨odl, I. Essa, G. Turk, and A. Bobick, “Graphcut textures: image and video synthesis using graph cuts,” ACM Trans. Graph., vol. 22, no. 3, pp. 277–286, 2003.
[7] L.-Y. Wei and M. Levoy, “Fast texture synthesis using tree-structured vector quantization,” in Proc. SIGGRAPH, 2000, pp. 479–488.
[8] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa, “Video textures,” in Proc. SIGGRAPH, 2000, pp. 489–498.
[9] F. Yang, G.-S. Xia, L. Zhang, and X. Huang, “Stationary dynamic texture synthesis using convolutional neural networks,” in Proc. IEEE Int. Conf. Signal Process., 2016, pp. 1135–1139.
[10] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, “Texture mixing and texture movie synthesis using statistical learning,” IEEE Trans. Vis. Comput. Graph., vol. 7, no. 2, pp. 120–135, 2001.

Feng Yang received the B.S. degree in science and technology of remote sensing from Wuhan University, Wuhan, China, in 2013. And she received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2018. She is currently a lecturer with School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. Her research interests include dynamic texture analysis, pattern recognition, and remote sensing image processing.

Gui-Song Xia (Senior Member, IEEE) received the Ph.D. degree in image processing and computer vision from LTCI, CNRS, Télécom ParisTech, Paris, France, in 2011. From 2011 to 2012, he was a Postdoctoral Researcher with the Centre de Recherche en Mathématiques de la Décision, CNRS, Paris-Dauphine University, Paris, for one and a half years. He is currently working as a Full Professor in computer vision and photogrammetry at Wuhan University. In 2018, he has also been working as a Visiting Scholar at DMA, École Normale Supérieure (ENS-Paris), for two months. He is also a Guest Professor with the Future Lab AI4EO, Technical University of Munich (TUM), Munich, Germany. His current research interests include mathematical modeling of images and videos, structure from motion, perceptual grouping, and remote sensing image understanding.

Dengxin Dai is currently a Lecturer with the Computer Vision Lab at ETH Zurich. He is leading a research group in Zurich, working on the R&D project "TRACE: Toyota Research on Automated Cars in Europe". He obtained his Ph.D. degree in Computer Vision at ETH Zurich in 2016. Before that, he obtained his M.S. Degree and B.S. degree in Image Processing from Wuhan University, Wuhan, China, in 2008 and 2010, respectively. His current research interests include robust vision algorithms under adverse conditions, autonomous driving models, sensor fusion, and learning with limited/weak supervision.

Liangpei Zhang (Fellow, IEEE) received the B.S. degree in physics from Hunan Normal University, Changsha, China, in 1982, the M.S. degree in optics from the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China, in 1988, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1998.
He is a Chair Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. From 2011 to 2016, he was a Principal Scientist for the China State Key Basic Research Project appointed by the Ministry of National Science and Technology of China to lead the Remote Sensing Program in China. He has published more than 700 research papers and five books. He is the Institute for Scientific Information highly cited author. He is the holder of 30 patents. His research interests include hyperspectral remote sensing, high-resolution remote sensing, image processing, and artificial intelligence.
Dr. Zhang was a recipient of the 2010 Best Paper Boeing Award, the 2013 Best Paper ERDAS Award from the American Society of Photogrammetry and Remote Sensing, and the 2016 Best Paper Theoretical Innovation Award from the International Society for Optics and Photonics (SPIE). His research teams won the top three prizes of the IEEE GRSS 2014 Data Fusion Contest, and his students have been selected as the winners or finalists of the IEEE International Geoscience and Remote Sensing Symposium student paper contest in recent years. He is the Founding Chair of the IEEE Geoscience and Remote Sensing Society (GRSS) Wuhan Chapter. He also serves as an associate editor or an editor for more than ten international journals. He serves as an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He is a Fellow of the Institution of Engineering and Technology.

Two-Stream Graph Convolutional Network for Intra-oral Scanner Image Segmentation

Yue Zhao[1,2], Lingming Zhang[1,2], Yang Liu[3], Deyu Meng[4,8], Zhiming Cui[5,6], Chenqiang Gao[1,2], Xinbo Gao[1], Chunfeng Lian[4], Dinggang Shen[5,7]

[1]School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China.

[2]Chongqing Key Laboratory of Signal and Information Processing, Chongqing, China.

[3]Department of Orthodontics, Stomatological Hospital of Chongqing Medical University, Chongqing, China.

[4]School of Mathematics and Statistics, Xi'an Jiaotong University, Xian, China.

[5]School of Biomedical Engineering, ShanghaiTech University, Shanghai, China.

[6]School of Computer Science, The University of Hong Kong , Hong Kong , China.

[7]Shanghai United Imaging Intelligence Co., Ltd., Shanghai, China.

[8]Faculty of Information Technology, Macau University of Science and Technology, Macau, China.

Introduction

One essential task of an advanced computer-aided-design (CAD) system is to perform fully automatic tooth segmentation on the intra-oral scanner images reconstructed by the intra-oral scanners (IOS). In this task, accurate labeling of each tooth and the derived information from labeled teeth are critical for various subsequent tasks towards precise personalized treatment.

So far, there are two categories of conventional methods proposed to segment teeth from intra-oral scanner images. 1) projection-based methods [1] usually first project the 3D intra-oral scanner image onto a 2D space to perform image-wise segmentation, and then reconstruct the segmentation result back to the original 3D space. 2) geometry-based methods [2], [3], [4], [5] typically use pre-selected geometric attribute to separate mesh cells. However, these geometry-based methods are not fully automatic, as manual initialization relying on domain knowledge and experience is often required.

Encouraged by the successful applications of convolutional neural networks (CNNs), some CNN-based methods have also been proposed to segment teeth from intra oral scanner images. They typically organize hand-crafted feature vectors as 2D images [6] or voxelize unordered mesh vertices/cells as 3D grid volumes [7], which are then used as the input of a segmentation network. Such operations inevitably ignore the unordered nature of geometric data or may introduce additional computational costs. More recent works proposed to learn translation-invariant geometric features from the raw mesh data for vertex/cell-wise labeling on 3D dental surfaces. However, these methods often simply concatenate different raw attributes as the input vector to train a single-stream segmentation network, potentially resulting in isolated false predictions on the intra-oral scanner image.

In this paper, we propose a two-stream graph convolutional network (i.e., TSGCN) to learn discriminative geometric features from heterogenous multi-view inputs for end-to-end tooth segmentation. Our TSGCN starts with two parallel branches, namely C stream and N-stream, to learn independently multi-scale feature representations from coordinates and normal vectors, respectively. These complementary single-view representations are then combined by a fusion branch integrating a self-attention mechanism, which minimizes the inter-view confusion and adaptively balances the contributions of different inputs to learn high-level multi-view geometric representations for the segmentation task.

1. Related Work

Conventional tooth segmentation methods using preselected geometric properties can be roughly grouped as curvature-based methods [8,9,10], contour-line based methods [11, 12] and harmonic-field-based methods [13]. Due to the typical requirement of manual steps and domain knowledge, the efficacy of such semi-automated methods heavily depends on the operator experience. In recent years, several deep learning-based methods have been proposed for fully-automated tooth segmentation on dental models. Typically, Xu *et al*. [6] proposed to reshape handcrafted geometric features as 2D image patches to train CNNs for classifying the mesh cells. Tian *et al*. [7] proposed to voxelize the dental model with a sparse octree partitioning, after which 3D CNNs are applied for tooth segmentation. Zanjani *et al*. [14] proposed an end-to-end network that integrates PointCNN [15] with a discriminator to directly segment the raw dental surfaces acquired by IOS. Lian *et al*. [16] extended PointNet by adding a multi-scale graph-constrained module to extract fine-grained local geometric features from dental mesh data. Instead of using solely the 3D coordinates, Lian *et al*. [16] combined 3D coordinates and normal vectors as the network input to improve the segmentation performance.

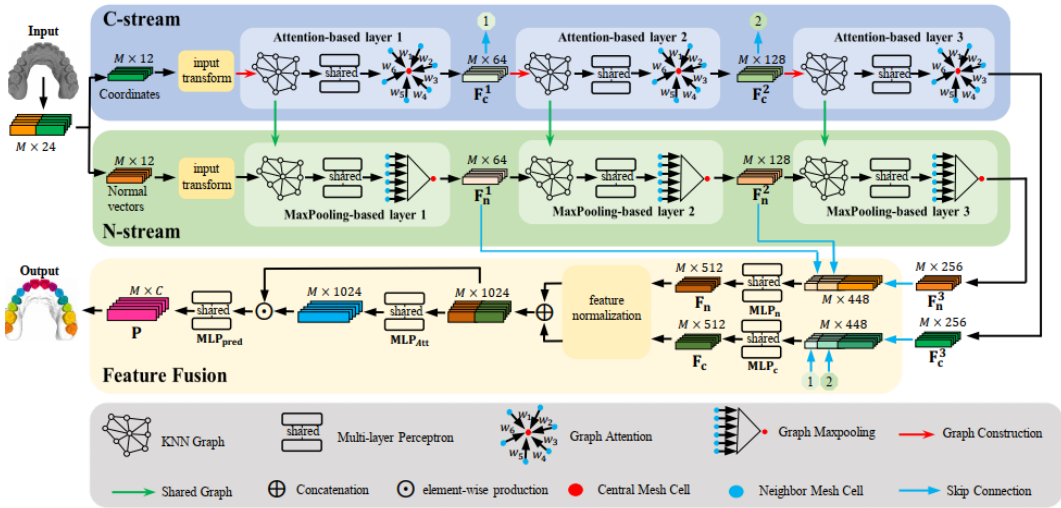


Fig. 1 Structure of our TSGCN.

2. Structure of TSGCN

As illustrated in Fig. 1, our TSGCN starts with two parallel streams, i.e., the C-stream and N-stream, which adopt input-specific graph-learning layers to extract high-level geometric representations from the coordinates and normal vectors, respectively. After that, these single-view features produced by these two complementary streams are further combined in the feature-fusion branch to learn more discriminative multi-view representations for teeth segmentation.

3.1 C-Stream

Given the input of an $M \times 12$ coordinate matrix $\mathbf{F}_c^0$, the C-stream first adopts an input transformer module to learn an affine transformation matrix $\boldsymbol{T}$, which updates $\mathbf{F}_c^0$ as:

$$\hat{\mathbf{F}}_c^0 = \mathbf{F}_c^0 \mathbf{T} \tag{4.1}$$

Following the input-transformer module, a series of graph attention layers are successively applied in the forward path of the C-stream to hierarchically extract multi-scale geometric features from the coordinate aspect. Specifically, given the feature matrix $\mathbf{F}_c^l \in \mathbf{R}^{N \times d}$ learned by (l - 1)-th graph attention layer. We construct a dynamic KNN graph $G(V, E)$. Notably, each node $m_i \in V$ only connects to its KNNs, which can be denoted as $N\ (i)$. Then, we calibrate the local information for each center m_i, such as:

$$\hat{\mathbf{f}}_{ij}^l = MLP^l(\mathbf{f}_i^l \oplus \mathbf{f}_{ij}^l), \forall m_{ij} \in N\ (i) \tag{2}$$

We estimate the attention weights for the neighborhood $N\ (i)$. A lightweight network shared across cells/nodes is used to learn attention weights. Specifically, the attention weight α_{ij}^l is defined as:

$$\boldsymbol{\alpha}_{ij}^l = \sigma(\Delta\mathbf{f}_{ij}^l \oplus \mathbf{f}_{ij}^l), \forall m_{ij} \in N\ (i) \tag{3}$$

where the function σ is implemented as a lightweight MLP, which adopts both $\Delta\mathbf{f}_{ij}^l = \mathbf{f}_i^l - \mathbf{f}_{ij}^l$ and $\mathbf{f}_{ij}^l$ as the inputs. *Finally*, we aggregate the neighborhood information to each center, which is formulated as:

$$\mathbf{f}_i^{l+1} = \sum_{m_{ij} \in N\ (i)} \boldsymbol{\alpha}_{ij}^l \square\ \hat{\mathbf{f}}_{ij}^l \tag{4}$$

3.2 N-Stream

As a complementary branch to the C-stream, we further design an N-stream to learn fine-grained boundary representations from the aspect of normal vectors. Our N-stream takes as inputs the normal vectors for all cells, the N-stream is restricted to share the same KNN graphs constructed in the C-stream. In contrast to the case of using the same KNN graphs, the N-stream adopts graph max-pooling layers different from the graph-attention layers in the C-stream for feature extraction, such as:

$$\mathbf{f}_i^{l+1} = maxpooling\{\hat{\mathbf{f}}_{ij}^l, \forall m_{ij} \in N\ (i)\} \tag{5}$$

It is worth mentioning that we use max-pooling (rather than graph attention) in the N-steam since the max operator can more sensitively capture the most distinctive features presented at the tooth boundaries.

3.3 Feature Fusion

After extracting single-view representations in the C-stream and N-stream, respectively, our TSGCN further fuses them to learn more discriminative multi-view representation for teeth segmentation. This operation can be formulated as:

$$\mathbf{F}_c = MLP_c(\mathbf{F}_c^1 \oplus \mathbf{F}_c^2 \oplus \mathbf{F}_c^3) \tag{6}$$

$$\mathbf{F}_n = MLP_n(\mathbf{F}_n^1 \oplus \mathbf{F}_n^2 \oplus \mathbf{F}_n^3) \tag{7}$$

The single-view representations are further harmonized by using a mesh-wise normalization operation. For the same row in $\mathbf{F}_c, \mathbf{F}_n$, e.g., $\hat{\mathbf{f}}_c^i \in \mathbf{F}_c, \hat{\mathbf{f}}_n^i \in \mathbf{F}_n$, we adopt a self-attention mechanism to enable the network to adaptively balance the contributions of $\hat{\mathbf{f}}_c^i$ and $\hat{\mathbf{f}}_n^i$, such as:

$$\boldsymbol{\beta}_i = MLP_{Att}(\hat{\mathbf{f}}_c^i \oplus \hat{\mathbf{f}}_n^i) \tag{8}$$

the multi-view feature geometric representation of m_i can be quantified as:

$$\hat{\mathbf{f}}^i = \boldsymbol{\beta}_i \square (\hat{\mathbf{f}}_c^i \oplus \hat{\mathbf{f}}_n^i) \tag{9}$$

3. Experimental Results

Our TSGCN is compared with five state-of-the-art methods for 3D shape segmentation (i.e., **PointNet** [18], **PointNet**++ [20], **PointCNN** [12], **DGCNN** [29]) and intra-oral scanner image segmentation (i.e., **MeshSegNet** [14]). The overall segmentation performance is quantitatively evaluated by two metrics, i.e., 1) Overall Accuracy (OA), and 2) mean Intersection-over-Union (mIoU). Besides, we also calculate the detailed IoU of each class.

The quantitative segmentation results obtained by all competing methods are summarized in Table 1. we can see that our TSGCN consistently obtained superior overall accuracy than all the competing methods in terms of OA and mIoU, demonstrating the state-of-the-art performance by our TSGCN in automatic teeth segmentation. Besides, Our TSGCN achieved better IoU values than other competing methods in segmenting each tooth, suggesting the generalization ability of our method in handling the varying teeth appearances.

Table1 The Segmentation result of all methods.

Method	All teeth		Each class (IoU)							
	OA	mIoU	T1	T2	T3	T4	T5	T6	T7	BG
PointNet [18]	84.95	66.86	55.31	65.31	69.35	75.47	72.21	66.18	74.71	84.86
PointCNN [12]	88.61	72.86	61.72	66.45	68.10	78.98	78.57	70.51	72.15	86.39
PointNet++ [20]	90.25	78.14	67.82	74.61	78.10	82.73	80.70	74.67	78.94	87.52
DGCNN [29]	91.93	84.30	82.18	79.95	82.09	87.88	86.24	80.14	84.26	91.65
MeshSegNet [14]	93.11	84.47	81.31	83.65	82.15	82.87	84.81	81.93	87.10	91.94
Ours	**96.96**	**91.69**	**83.47**	**91.29**	**93.53**	**94.84**	**93.14**	**90.26**	**91.32**	**95.67**

References

[1] K. T., F. K.W.C., and O. S.H. Tooth segmentation of dental study models using range images. *IEEE Transactions on Medical Imaging*, 23(3):350–362, 5 2004.

[2] T. Kronfeld, T. Kronfeld, D. Brunner, D. Brunner, G. Brunnett, and G. Brunnett. Snake-based segmentation of teeth from virtual dental casts. *Computer-Aided Design and Applications*, 7(2):221–233, 12 2010.

[3] Y. Kumar, R. Janardan, B. Larson, and J. Moon. Improved segmentation of teeth in dental models. *Computer-Aided Design and Applications*, 8(2):211–224, 2011.

[4] C. Sinthanayothin and W. Tharanont. Orthodontics treatment simulation by teeth segmentation and setup. In *5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2008), vol.1*, pages 81–84, Krabi, Thailand, 1 2008.

[5] K. Wu, L. Chen, J. Li, and Y. Zhou. Tooth segmentation on dental meshes using morphologic skeleton. *Computers & Graphics*, 38:199–211, 2014.

[6] X. Xu, C. Liu, and Y. Zheng. 3d tooth segmentation and labeling using deep convolutional neural networks. *IEEE transactions on visualization and computer graphics*, 25(7):2336–2348, 2018.

[7] S. Tian, N. Dai, B. Zhang, F. Yuan, Q. Yu, and X. Cheng. Automatic classification and segmentation of

teeth on 3d dental model using hierarchical deep learning networks. *IEEE Access*, 7:84817–84828, 2019.

[8] L. Zhang, Y. Zhao, D. Meng, Z. Cui, C. Gao, X. Gao, C. Lian, and D. Shen. Tsgcnet: Discriminative geometric feature learning with twostream graph convolutional network for 3d dental model segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6699–6708, 2021.

[9] M. Zhao, L. Ma, W. Tan, and D. Nie. Interactive tooth segmentation of dental models. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 654–657. IEEE, 2006.

[10] Y. Kumar, R. Janardan, B. Larson, and J. Moon. Improved segmentation of teeth in dental models. *Computer-Aided Design and Applications*, 8(2):211–224, 2011.

[11] C. Sinthanayothin and W. Tharanont. Orthodontics treatment simulation by teeth segmentation and setup. In *5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2008), vol.1*, pages 81–84, Krabi, Thailand, 1 2008.

[12] M. Yaqi and L. Zhongke. Computer aided orthodontics treatment by virtual segmentation and adjustment. In *2010 International Conference on Image Analysis and Signal Processing*, pages 336–339. IEEE, 2010.

[13] Bei Ji Zou A, Shi Jian Liu A, Sheng Hui Liao A, Xi Ding B, and Ye Liang C. Interactive tooth partition of dental mesh base on tooth-target harmonic field. *Computers in Biology and Medicine*, 56(Feb.):132–144, 4 2015.

[14] F. Ghazvinian Zanjani, D. Anssari Moin, B. Verheij, F. Claessen, T. Cherici, T. Tan, and P. de With. Deep learning approach to semantic segmentation in 3d point cloud intra-oral scans of teeth. volume 102, pages 557–571. PMLR, 2019.

[15] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, volume 31, pages 820–830. 2018.

[16] C. Lian, L. Wang, T. Wu, F. Wang, P. Yap, C. Ko, and D. Shen. Deep multi-scale mesh feature learning for automated labeling of raw dental surfaces from 3d intraoral scanners. *IEEE Transactions on Medical Imaging*, 39(7):2440–2450, 2020.

Yue Zhao received Ph.D. degree from Jilin University, Jilin, China, in 2017. She is currently a lecturer with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. Her research interests include medical image processing and analysis, pattern recognition.

Lingming Zhang is currently pursuing the Master degree with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include medical image processing and analysis, computer vision.

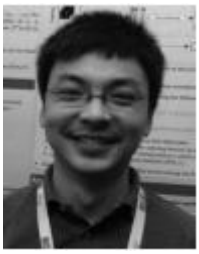
Chenqiang Gao received Ph.D. degree from Huazhong University of science and technology. He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include Infrared image analysis, target detection and recognition, action recognition.

Yang Liu received the D.M.D. degree from School of Stomatology, Peking University in 2014. He is currently a Orthodontist, Affiliated Stomatological Hospital of Chongqing Medical University, Chongqing, China. His research interests include biological mechanism of orthodontic tooth movement, tooth image processing and application.

Zhiming Cui is now a Ph.D. candicate in the computer vision and graphics lab at the University of Hong Kong, Hong Kong, China. His research interests include medical image analysis and 3D vision.

Deyu Meng (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2001, 2004, and 2008, respectively. From 2012 to 2014, he took his two-year sabbatical leave at Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Professor with the School of Mathematics and Statistics, Xi'an Jiaotong University, and an Adjunct Professor with the Faculty of Information Technology, Macau University of Science and Technology, Taipa, Macau. His research interests include self-paced learning, noise modeling, and tensor sparsity.

Xinbo Gao (Senior Member, IEEE) received the B.Eng., M.Sc., and Ph.D. His current research interests include multimedia analysis, computer vision, pattern recognition, machine learning, and wireless communications. He has published five books and around 200 technical articles in refereed journals and proceedings. He is on the Editorial Boards of several journals, including Signal Processing (Elsevier) and Neurocomputing (Elsevier). He has served as the General Chair/Co-Chair, the Program Committee Chair/Co-Chair, or a PC Member for around 30 major international conferences. He is also a fellow of the Institution of Engineering and Technology.



Chunfeng Lian (Member, IEEE) received the B.S. degree in electronic and information engineering from Xidian University, Xi'an, China, in 2010, and the Ph.D. degree in computer engineering from Univiersité de Technologie de Compiègne, CNRS, Heudiasyc (UMR 7253), Compiègne, France, in 2017. His research interests include medical image analysis, pattern recognition, and machine learning.

Dinggang Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1995. He has published more than 1100 peer-reviewed papers in the international journals and conference proceedings, with H-index 99. His research interests include medical image analysis, computer vision, and pattern recognition. Dr. Shen served on the Board of Directors, the Medical Image Computing and Computer Assisted Intervention (MICCAI) Society, in 2012–2015, and was the General Chair for MICCAI 2019. He is a fellow serves as an editorial board member for eight international journals. He is a Fellow of the American Institute for Medical and Biological Engineering and the International Association for Pattern Recognition.

BreathTrack: Tracking Indoor Human Breath Status via Commodity WiFi

Dongheng Zhang, Yang Hu, Yan Chen, Bing Zeng
University of Electronic Science and Technology of China, Chengdu, China

1. Introduction

Breath rate is an important vital sign for health monitoring and medical diagnosis. While we have witnessed, in the past few years with the development of Internet of Things (IoT), the increasing research interests and progresses in ubiquitous health monitoring [1][2], traditional methods are intrusive that require the physical contact between human and sensors [3][4], which affects the normal breath of human and cannot be applied to the long-term breath monitoring.

To resolve this challenge, radio frequency based monitoring schemes that can provide non-intrusive human sensing have been proposed, e.g., the vital-radio uses the frequency modulated continuous wave signal to estimate the breath and heart rate [5]. However, such a system is not only expensive, but also occupies a large wireless band, which limits its application. To reduce the cost, the WiFi-based methods have been proposed [6]. In [6], the UbiBreathe system has been proposed to utilize the Received Signal Strength (RSS) to monitor the breath signal. Since the RSS is not very sensitive to the minor displacements in the environment, it requires users to hold the WiFi devices close to their chests to achieve reasonable performance. The amplitude of Channel State Information (CSI) has also been utilized to estimate the breath rate [7]. However, similar to the RSS, the amplitude of CSI is also not very sensitive to the minor displacement in the environment, due to which the estimation performance is limited. To achieve reasonable performance, sophisticated subcarrier selection, denoising and filtering procedures are needed. Since the frequency of breath varies in a very narrow band, i.e., about 0.2Hz, it is difficult to determine whether the estimated frequency is corresponding to the breath frequency.

Compared with the RSS and the amplitude of CSI, the phase of CSI is much more sensitive to the minor displacement in the environment, i.e., the phase of CSI is more suitable for breath rate estimation. However, the hardware imperfection of the commodity WiFi chips will introduce different kinds of phase distortions, which makes it difficult to obtain accurate CSI phase information. It has been found that the phases of the measured CSIs across packets are not correlated even in very short time intervals [8]. Therefore, it is very difficult to estimate the breath rate directly from the phase variation of the measured CSI. To solve the phase distortion problem, the PhaseBeat system was proposed in [9], which utilizes the phase difference between receiver antennas to eliminate the phase distortion. However, since the CSI measured on the antennas is affected by the minor displacement caused by breath, the phase difference between antennas is actually the subtraction of two periodic signals, which makes the model in [9] inaccurate. Another challenge for the breath tracking in the indoor environment is the multipath effect. In a multipath indoor environment, the received signal is not periodic due to the aggregation of the multipath effect, and thus the breath rate cannot be directly obtained from the frequency components of the CSI. Moreover, the breath may be interrupted due to various factors such as talking, thinking or even some unconscious behaviors. In such a case, only the estimation of the breath rate may not be enough. Instead, it would be more significant if we could track the detailed breath status, i.e., the time domain breath waveform similar to the Electrocardiography (ECG) signals. Such detailed breath status could be an important indicator for disease diagnosis. However, the performance of the existing WiFi-based solutions is quite limited when utilized to track the detailed breath status.

To resolve the challenges, in this paper, we propose a contact-free breath tracking system using the off-the-shelf WiFi devices, BreathTrack, to track the human breath. BreathTrack exploits the phase variation of the CSI to track human breath. To avoid the phase distortions and obtain the accurate phase information, BreathTrack combines the hardware and software corrections. Specifically, the time-invariant PPL Phase Offset (PPO) is calibrated by the hardware correction using cables and splitters, while the time-varying Carrier Frequency Offset (CFO), Sampling Frequency Offset (SFO) and Packet Detection Delay (PDD) are removed by the software corrections using the phase difference between the CSI at the receiver antennas and that at the reference antenna connected from the transmitter antenna. To eliminate the multipath effect in the indoor environment, BreathTrack utilizes the sparse recovery method to find the dominant path in the environment and obtain the corresponding complex attenuation coefficient of this dominant path from the CSI. Then, the phase variation of the complex attenuation coefficient is utilized to extract the detailed breath status and the breath rate. Extensive experiments are conducted to show that BreathTrack could estimate the breath rate with the median accuracy of over 99\% in most scenarios. Besides the breath rate, BreathTrack could track the detailed status of breath directly using the raw phase variation.

2. System Model and BreathTrack System

The overview of the proposed BreathTrack system is shown in Fig. 1. We first measure the raw CSI on different

antennas and subcarriers. Then, we calibrate the CSI to remove random phase offsets which prevent current WiFi chips from extracting human breath signal. Then, we estimate the signal Angle of Arrival(AoA) and Time of Flight(ToF) through sparse recovery. Finally, we extract the human breath signal based on the AoA-ToF and output the breath rate and status.

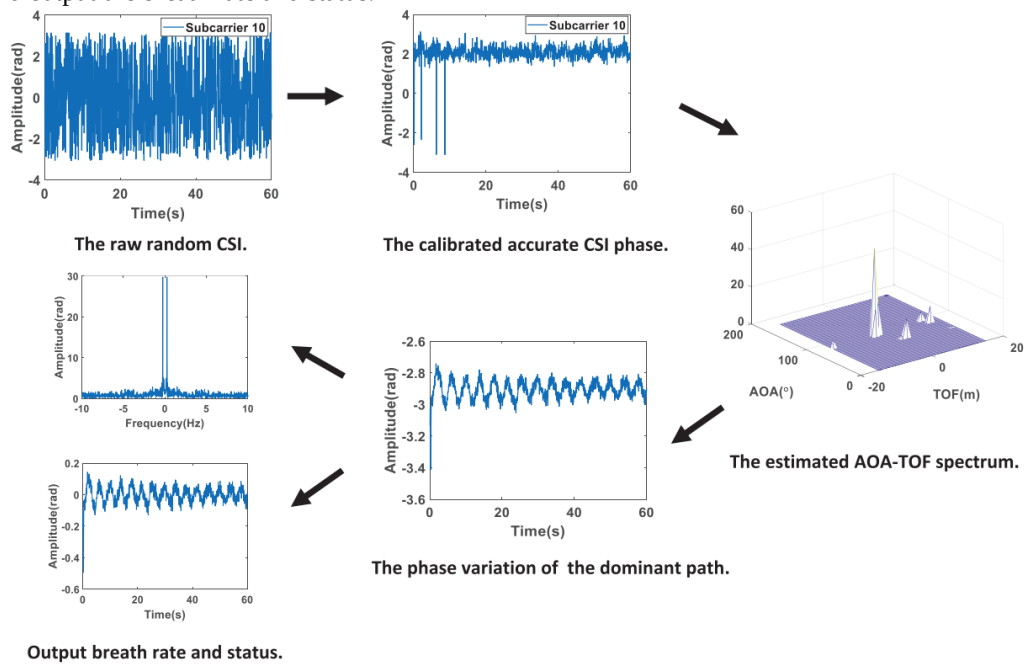


Fig. 1: System overview

To illustrate the contributions of this paper, consider a practical wireless channel with multi-path propagation, the measured CSI distorted by the random noise can be expressed as

$$y = \sum_{l=1}^{L} a(\theta_l, \tau_l)\alpha_l + e = A\alpha + e \quad (1)$$

where $a(\theta_l, \tau_l)$ denotes the steering vector of path l determined by the AoA and ToF of signal propagation, α and e denotes the complex signal attenuation and additive noise introduced by the receiver. With a person breathing in the wireless channel, the chest movement caused by breath would modulate the measured CSI. Hence, we could estimate the human breath rate based on the signal variation of y. Since off-the-shelf WiFi chips are equipped with multiple antennas and adopt OFDM for signal modulation, the CSI is measured on multiple antennas and subcarriers. Previous works treat signals on different antennas and subcarriers separately, which does not fully utilize the spatial resolution of WiFi chips to enhance the human breath signal. To solve this problem, we propose to estimate the AoA and ToF of human reflections and then extract the corresponding signal based on the AoA-ToF. To estimate the AoA and ToF, we cast the problem into a sparse recovery formulation and solve the following optimization problem [10-12]

$$\min \ ||\bar{\alpha}||_1$$
$$s.t. ||y - \bar{A}\bar{\alpha}||_2^2 \quad (2)$$

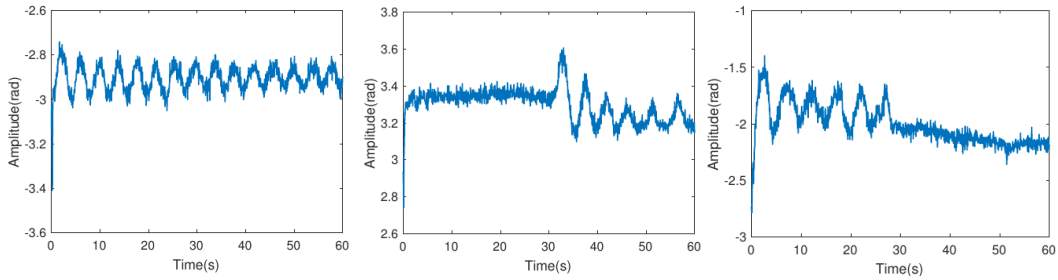
With (2), the AoA-ToF and complex signal attenuation can be extracted. Since the chest movement caused by the human breath would modulate the CSI phase, we could obtain the human breath variation from the phase of α. However, due to the hardware imperfection and timing offsets, the received signal phase is distorted by PPO, CFO, SFO and PDD[15]. In this case, the received signal on antenna m and subcarrier k can be expressed as

$$y_{m,k}(t) = e^{-j2\pi(f_{CFO}(t)t + k\Delta f(\tau_{SFO}(t) + \tau_{PDD}(t)))} e^{j\varphi_{PLL}} \sum_{l=1}^{L} \alpha_i e^{-j2\pi f_k \tau_l^m} \quad (3)$$

Since the PPO φ_{PLL} is time invariant, we can measure and remove it from the received signals. However, other phase offsets, CFO, SFO and PDD variant randomly with time. In this case, the received signal phase could not be utilized to extract human breath anymore. To solve this problem, we have noted that although these offsets are difficult to track, they are the same among different antennas. Hence, we first perform conjugation multiplication among different antennas to cancel these phase offsets. Then, we apply (2) to obtain the signal variation caused by human breath. Finally, we extract breath variation from the phase of the signal.

3. Experiments

We use two desktop computers equipped with the Intel 5300 NIC as the transmitter and the receiver. The Linux 802.11n CSI Tool [14] is installed on the Ubuntu desktop 14.04 LTS OS for both the transmitter and the receiver. We randomly choose channel 62, i.e., 5.31GHz center frequency with the 40MHz bandwidth, as our experimental band. The receiver operates in the ``monitor'' mode. It is equipped with a uniform linear array which is composed of three omnidirectional antennas, while we only use two of them since one of the port is connected with the transmitter directly. The space interval of the antennas is 2.6cm, which is about the half wavelength. The transmitter operates in the ``inject'' mode using one omnidirectional antenna. It injects 20 packets per second. The transmitted signal is first divided into two parts using a microwave power splitter: one is fed into the transmit antenna and the other is fed into an attenuator which is connected with the receiver via a coaxial cable.



(a) Breath normally (b) Hold and then breath (c) Breath and then hold

Fig. 2: Extracted human breath signal

The extracted human signals in three different scenarios are shown in Fig. 2. The participant is asked to breath normally, hold the breath and then breath normally, breath normally and then hold the breath in these three scenarios. According to Fig. 2, the proposed system is able to accurately capture such fast and tiny variation of human breath.

4. Conclusion

In this paper, we proposed BreathTrack, the first system that can track the detailed status of breath using the off-the-shelf WiFi devices. To achieve this, we proposed hardware and software correction methods to remove both the time-invariant and time-varying phase distortions introduced by the hardware imperfection of the commodity WiFi chips and thus obtain the accurate CSI. We also proposed a joint AOA-TOF sparse recovery method to eliminate the multipath effect in the indoor environment and extract the information of the dominant path to track the status of breath. Experimental results show that BreathTrack can achieve high accurate breath rate estimation and track the detailed status of breath. The proposed system is also applicable in other wireless sensing applications, such as activity recognition, events detection and speed estimation.

References

[1] M. Liu and M. Huang,``Asthma Pattern Identification via Continuous Diaphragm Motion Monitoring", *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 1, no. 2, pp. 76-84, Apr. 2015.

[2] I. Bisio, A. Delfino, A. Grattarola, F. Lavagetto and A. Sciarrone, ``Ultrasounds-based Context Sensing Method and Applications over the Internet of Things", *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3876-3890, Oct. 2018.

[3] iBabyGuard. http://www.ibabyguard.com/.

[4] VivoSense. Sensors: Equivital. https://www.vivosense.com/portfolio/equivital/

[5] F. Adib., H. Mao, Z. Kabelac., D. Katabi and R. C. Miller. ``Smart homes that monitor breathing and heart rate," *ACM CHI*, 2015.

[6] H. Abdelnasser, K. A. Harras and M. Youssef, ``UbiBreathe: A Ubiquitous non-Invasive WiFi-based Breathing Estimator", *ACM MobiHoc*, 2015.

[7] X. Liu, J. Cao, S. Tang, J. Wen, and P. Guo, ``Contactless Respiration Monitoring Via Off-the-Shelf WiFi Devices", *IEEE Trans. Mobile Comput*., vol. 15, no. 10, pp. 2466-2479, Oct. 2016.

[8] R. Nandakumar, B. Kellogg, and S. Gollakota. ``Wi-fi gesture recognition on existing devices." arXiv:1411.5394, 2014.

[9] X. Wang, C. Yang and S. Mao, ``PhaseBeat: Exploiting CSI phase data for vital sign monitoring with commodity WiFi devices," *IEEE ICDCS*, 2017.

[10] D. Malioutov, M. Cetin, and A. S. Willsky, ``A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process*., vol. 53, no.8, Aug. 2005.

[11] W. Gong and J. Liu, ``Robust indoor wireless localization using sparse recovery," *IEEE ICDCS*, 2017.

[12] E. J. Cand`es and M. B. Wakin, ``An introduction to compressive sampling," *IEEE Signal Processing Mag.*, vol. 25, no. 2, pp. 21-30, Mar. 2008.

[13] CVX solvers, http://cvxr.com/cvx/doc/solver.html.
[14] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, ``Tool release: gathering 802.11n traces with channel state information," *ACM SIGCOMM CCR*, 2011.
[15] Y. Zhuo, H. Zhu, H. Xue, and S. Chang. ``Perceiving Accurate CSI Phases with Commodity WiFi Devices," *IEEE INFOCOM*, 2017.

Survey of Graph Databases and Current Challenges in handling their Design and Computation

Sudip Dhakal
University of North Texas, Texas, USA
sudipdhakal@my.unt.edu

1. Introduction

The graph is a collection of nodes, edges and the relationship between them. Graph databases on the other hand is a multi-relational graph. This is different from the relational database which represents a collection of interlinked tables. Graph databases use structure of semantic queries with nodes, edges, and properties to represent and store data. Graph databases have many advantages over the traditional database systems hence its usage is increasing rapidly over the years. In this paper, we provide a qualitative survey of different types of graph databases, compare them with traditional database systems, present various challenges in handling them and overall provide a general overview of managing and modeling of graph database systems.

Graph database is a multi-relational graph. While traditional databases store data or information as a collection of interlinked tables, graph databases on the other hand use the concept of nodes, edges and properties or relationships to represent and store data. In the graph, nodes are terms as entities, and we can think of multiple ways where these entities are correlated in different types of applications. The relationship can be termed as the connection between these entities. In graph databases, data term "Attributes" related to entities and relationships are called labels. Also, data is stored into nodes, and these nodes have some properties, relationships consist of properties and connect one node to the other node. The graph database exhibits a single data structure known as a graph, and it has no combined operation, and hence each edge will be connected to another edge.

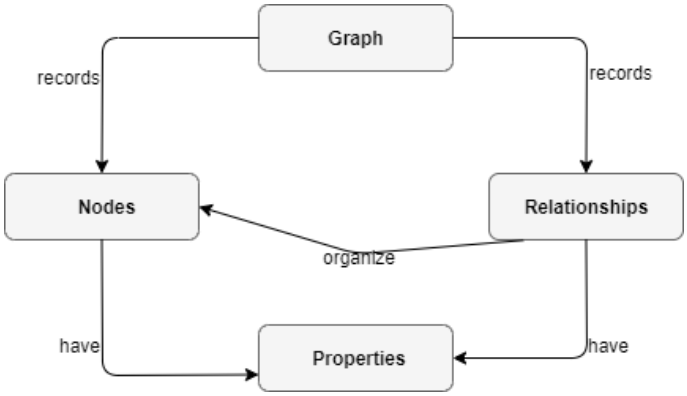


Figure 1: Basic Structure of Graph Database

While relational database systems are good for well-understood data structures that don't change too frequently, graph databases on the other hand are well suited for dynamic systems: where the data topology is changing overtime and is very difficult to predict. And with the current advancement in multiple dynamic systems in almost every single application, graph databases are most suited for managing data and information. And likely the use of graph databases has been increasing over the years. From being used in simple and uncomplicated modeling of data to huge application domains such as social networks, transportation networks, knowledge bases and many more, they have flourished over the years due to ease in programming, structure, processing and modeling. Some of the most familiar and common graph databases include Labeled graph, Property graph, RDF, Hypergraph, DEX, Trinity, Infinite Graph and so on. These database systems provide high performance and handling of data and information. However, with the increase in scalable systems, emergence of big data, and security and privacy terms of utmost concern, designing and handling graph databases has become one of the most challenging tasks. A precise study of all the prevailing graph database systems and their application along with the challenges in handling them is required to provide a clear understanding of graph databases. Our contribution in this paper includes the following:

- We provide a general overview and taxonomy of existing graph database systems including Labeled graph, Property graph, RDF, Hypergraph, DEX, Trinity, Infinite Graph and so on.

- We outline the existing framework for graph database management
- We discuss in detail the power and limitation of graph databases
- We outline basic requirements in modeling of graph databases
- Finally, we also discuss the challenges of handling and processing graph databases in present scalable and huge data computational applications

We have organized our paper into 8 different sections. In section 1 we will discuss the existing graph database models. In section 2, 3 and 4 we present the concept of Querying Graph Databases, Graph Database Modeling and Integrity Constraints which is very important in understanding the graph database system as a whole. Similarly, in section 5 we discuss the various graph processing platforms and in section 6 we present different experiments and results with respect to various graph processing platforms and models. Moreover, in section 7 we present a qualitative comparison of graph database and traditional database. We wrap up the paper discussing the limitations and challenges in designing and using graph databases.

2. Sections

2.1. Existing types of Graph Database Models and systems

All graph data models are changed as their formal basis on the basic mathematical definitions of graphs, such as directed or undirected graphs, labeled or unlabeled edges and nodes, nodes and properties on edges, hypergraphs and hypernodes. In this section, we will discuss various Graph Database Models, based on all the existing surveys. Some of the most common database models include following:

1. **Labeled Graph**: This is the most basic data structure of graph database models. Nodes and edges are marked with some vocabularies. Figure 2.1 shows an example of labeled graph data model, as you can see, at schema level, the common names are used to define entities and relationships, at instance level, the instance labels are created and to represent entities and the edges to represent the relationships between data and entities [10].

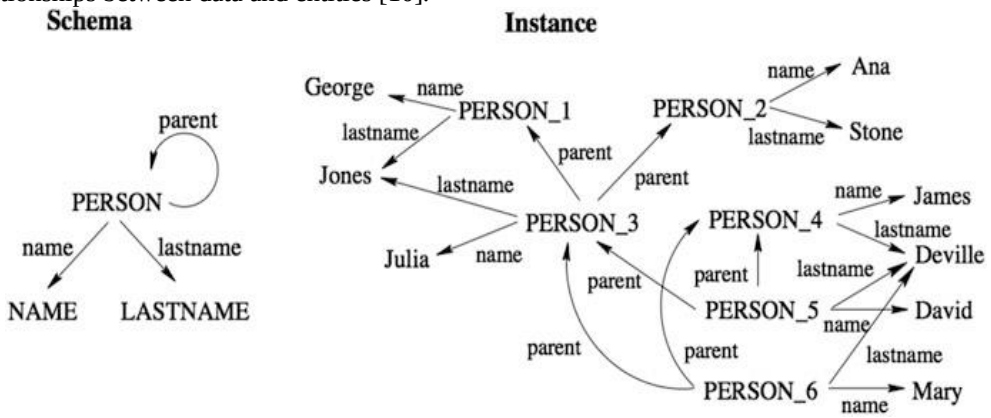


Fig. 2.1 Gram

2. **Property Graph Model**: The Property Graph is a graph where the nodes and edges are labelled, the edges are directed and there are many attributes [10]. Property Graphs are good at showing the connections between data scattered in different data structures and data patterns. They provide richer insights on how to model data on many different databases and how different types of metadata are related. The Property Graph also shows data dependencies that are not visible in relational database schema or other tools [13].

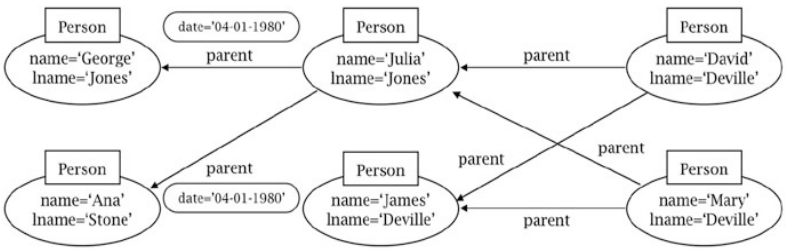


Fig. 2.2 Property graph data model

Figure 2.2 shows an example of a property graph. The main feature of this model is the appearance of properties in nodes and edges. Each property is represented as a pair property-name = “propertyvalue”[10].
The Property Graph module has become very popular through **Neo4J** and TINKERPOP and is now widely used and implemented by many others [7] [9] [13].

3. **RDF (Resource Description Framework) Model**: The RDF is a collection of specifications for representing information. It was a W3C-standardized data model to describe web resources and models data as an edge-labeled multigraph [13], the latest version of the RDF specification was published in 2014.The goal is to implement a simple format for easy data exchange between different data formats. It is particularly useful for describing irregularly connected data. The core part of the RDF model is the collection of triples. Each triple consists of a subject, a predicate and an object. Therefore, RDF databases are often called triple stres (or triplestores) [6].

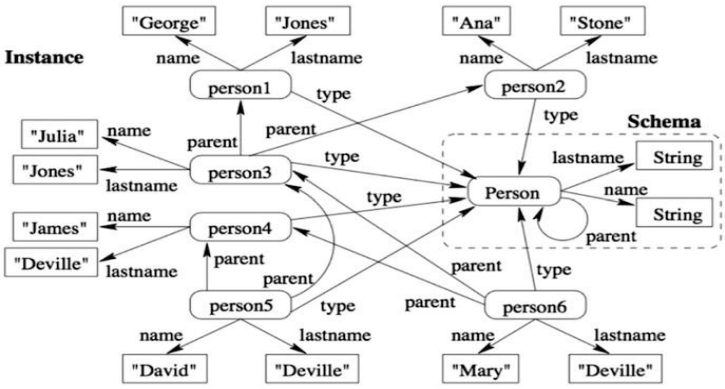


Fig. 2.3 RDF data model

Figure 2.3 is an example of RDF data mode; the schema and instance are mixed together. The edges labeled type disconnects the instance from the architecture. The instance is built by the subgraphs obtained by instantiating the nodes of the schema and establishing the corresponding parent edges between these subgraphs [6] [10] [13].

4. **Hypergraph Model**: Hypergraph is a generalization of graph, in which the concept of edge is extended to hyperedge, which is related to any node set [10]. It is a graph data model in which relationships can connect any number of given nodes. The property graph allows a relationship to have only one start node and one end node, while the hypergraph model allows any number of nodes at either end of the relationship.

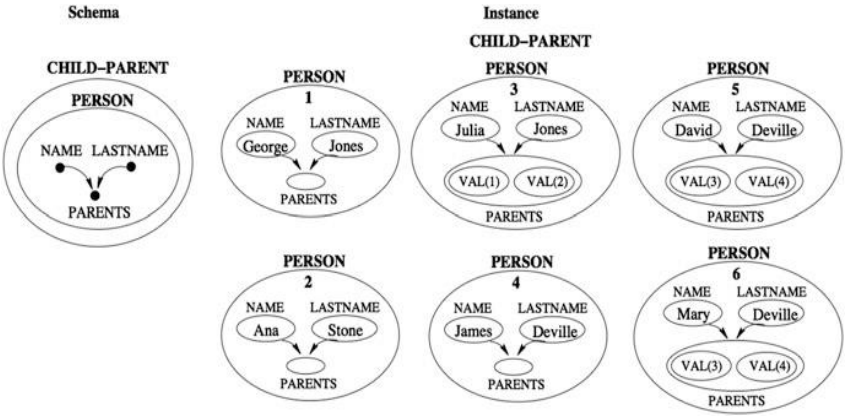


Fig. 2.4 GROOVY

Figure 2.4 shows a Hypergraph model example, this model shows that these structures can be defined in terms of hypergraphs [10]

5. **DEX**: DEX is a very effective bitmap-based graphics database, written in C++ language. First version released in 2008. It makes graph query possible in different networks, such as social network analysis and pattern recognition. For large graphs, it is also called an advanced graph database, which is useful for most NoSQL applications [7].
6. **Trinity**: Trinity is a distributed graphics system on the storage cloud [9]. The memory cloud is a memory key-value store that is globally addressed on a computer cluster. When we have large data sets, it provides fast data access capabilities. It is a large graphics processor. It provides fast graph exploration and parallel computing for larger data sets. It also provides high throughput for large graphs with one billion nodes [7].
7. **Infinite Graph**: The infinite graph is produced by an organization called objectivity. Infinite graph database is a distributed graph database in Java language, which is based on a structure similar to graphs. We can call the infinite graph as a graph database supporting the cloud [7].

2.2 Querying Graph Databases

This section includes an overview of the research on querying graph databases. We display a classification of queries, basically including pattern matching, adjacency and analytical queries [6][10][13].

1. **Pattern Matching Queries**: Basically, the purpose of graph pattern matching consists in finding the set of subgraphs of a database graph that "match" a given graph pattern. Graph pattern matching is usually defined according to subgraph isomorphism, that is to say, all subgraphs of database G are isomorphic to graph pattern P. Subgraph isomorphism is NP complete (Gallagher 2006) [10].
2. **Adjacency Queries**: The main concept of adjacency queries is node/edge adjacency. For example, two nodes are neighbors when there is an edge between them. Or two edges are neighbors when they share the same node. Several applications benefit from adjacency queries like a web ranking system (Chang and Chen 1998) and recommendation system (Dominguez-Sal et al. 2010a) [10].
3. **Analytical Queries**: The purpose of this type query is to quantitatively measure the topological characteristics of database graphs, usually in aggregate form [10]. Analytical queries can be supported by special operators that allow summary of query results or by the ability to hide complex algorithms.

2.3 Graph Database Modeling

In a native graph database model, both the schema and its instances are modelled as graphs. Nodes and edges are first-class citizens [5]. A data manipulation operator as well as users by data are equipped by the modeling system. This operator does not only return disconnected nodes but also handles entire graphs. Also, integrity constraints defined over the graph structure should be able to guarantee the consistency of the graph data and should be applied on whole subgraphs and not just the single nodes or edges in the graph. In the example given below we can observe that only the graph database schema may not be sufficient for application i.e. Obviously, each teacher can teach more languages and each teacher is born exactly in one town. For this, the integrity constraint should be defined at the conceptual level.

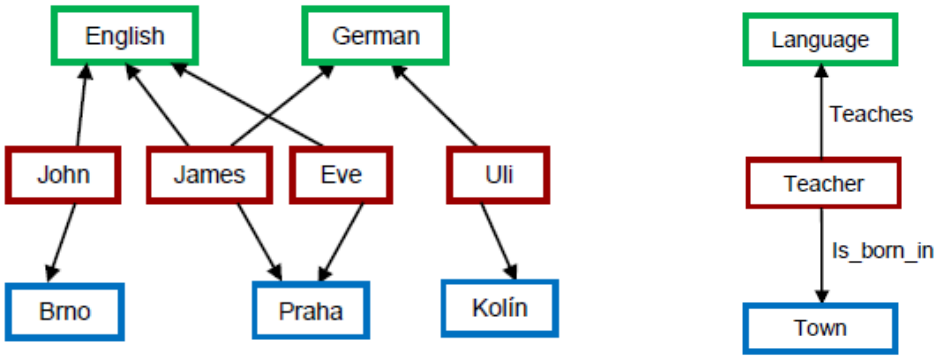


Fig. 2.5 Graph Database and schema

2.4 Graph-Processing Frameworks

With the increase in scalable systems, graphs with millions and billions of nodes have become very common nowadays. And with the increasing abundance of large graphs, handling, designing and processing of such

large graphs has become a very challenging and complicated task. Without a proper and reliable graph processing framework, it is almost impossible to comprehend and process those graphs. A comprehensive survey of the state-of-the-art of scalable graph processing platforms is presented by Omar Batarfi[4]. They have categorized current processing platforms for graph databases into 5 families namely: Hadoop Based System, Pregel Family, GraphLab Family, Other systems as seen in figure 2.6.

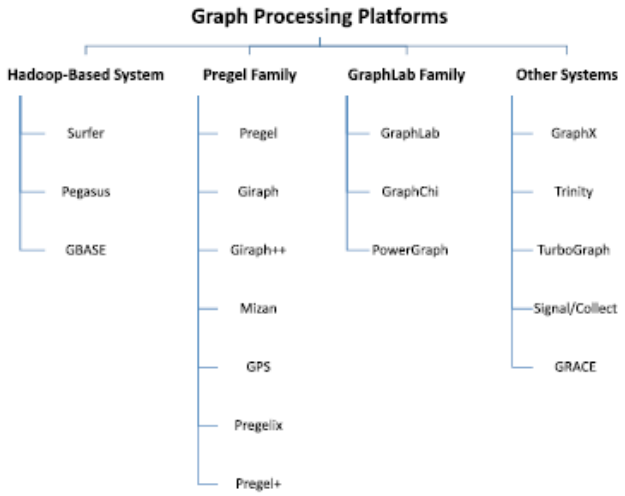


Fig. 2.6 Graph Processing Platforms

In general, graph processing platforms such as MapReduce can handle unstructured and tabular data. However, such frameworks are not efficient when it comes to analyzing or implementing iterative graph algorithms which requires multiple stages of complex joins. This leads to huge network traffic and optimization of graph systems becomes inefficient. In order to tackle such a problem, the Surfer system [4] has been presented which basically provides two functionalities namely: MapReduce and propagation. In this system, MapReduce processes different key-value pairs in parallel, and propagation is an iterative computational pattern that transfers information along the edges from a vertex to its neighbors in the graph [4]. GBASE is another Hadoop-based system which first partitions the input graph into a number of blocks and then reshuffles the nodes such that the nodes that belong to the same partitions are placed near to each other and then compresses the non-empty blocks. These compressed blocks are then stored into the graph storage. The various types of graph queries supported by GBASE include neighborhood, induced subgraph, egonet, K-core and cross-edges [4]. Similarly, PEGASUS supports graph mining tasks such as computing the diameter of the graph, computing radius of nodes and finding the connected components.

The Pragel system is developed by Google and falls under the Pragel Family. Such a system stores data locally and performs the computation on locally stored data which helps to avoid the communication overhead. In particular, Pregel distributes the graph vertices to the different machines of the cluster where each vertex and its associated set of neighbors are assigned to the same node. Graph processing algorithms are then represented as supersteps where each step defines what each participating vertex has to compute and edges between vertices represent communication channels for transmitting computation results from one vertex to another [4]. Apache Giraph is another open source project based on JAVA and was initially developed by YAHOO. Giraph runs graph processing jobs as map-only jobs on Hadoop and uses HDFS for data input and output. Graph also uses Apache ZooKeeper12 for coordination, checkpointing, and failure recovery schemes [4]. Similarly, GPS is an extension of Pragel platform and provides an additional function called master.compute(), which provides access to all the global aggregated values. Mizan's dynamic repartitioning strategy is based on monitoring the runtime characteristics of the graph vertices (e.g., their execution time, and incoming and outgoing messages) and uses this information, at the end of every superstep, to construct a migration plan with the aims of minimizing the variations across workers by identifying which vertices to migrate and where to migrate them to [4]. Pragel+ on the other hand reduces the number of exchanged messages between the worker nodes. This is done via a mirroring mechanism, which selects the vertices for mirroring based on a cost model. This cost model analyzes the tradeoff between mirroring and message combining. Similarly, in Pragelix the exchange of messages is treated as join operations. This join operation is followed by group-by operation which embeds functions that capture the semantics of the graph computation program.

GraphLab family includes graph processing platforms such as GraphLab, GraphChi and PowerGraph. GraphLab is an open-source graph processing platform, implemented in C++. While Pragel performs

computation locally on the stored data, GraphLab relies on shared memory abstraction. This abstraction consists of three parts namely: the data graph, the update function and the sync operation. The data graph is the user program, the update function represents the user computation. This is a distributed system or computational platform. GraphChi on the other hand is a centralized system that has the capability to process huge data and graphs that is present in the secondary storage in a given machine. While GraphChi can minimize the communication between the workers as it is a centralized system, it presents other challenges of distributed systems such as fault tolerance and management of the cluster.

So far we have discussed the Hadoop, Pragel and GraphLab processing platform. Other graph processing systems include GraphX, Trinity, TurboGraph, Signal/Collect and GRACE. Trinity is a cloud-based system and supports fast graph computation and exploration. This is suitable for large graphs. Similarly, TurboGraph is a disk-based graph processing platform which performs parallel computation on the graph. GraphX is a distributed graph and helps to minimize the communication and storage overhead.

2.5 Experiments and Results

In this section, we will outline the various experiments and corresponding results and compare the performance of different graph database systems in multiple dataset as presented by O. Batarfi [4] . For simplicity we have only considered 4 datasets namely: Wikitalk, Amazon, Citation and Friendster. The graph database systems that we have taken into account are Apache Giraph, GraphX, GPS, GraphLab.

Table 1. Dataset

Dataset name	Number of nodes	Number of edges	Size on disk
Wikitalk	2,394,385	5,021,410	1 GB
Amazon	21,365,698	140,015,189	18 GB
Citation	3,774,768	16,518,948	43 GB
Friendster	65,608,366	1,806,067,135	120 GB
LUBM 30K	12×10^8	3×10^9	700 GB
LUBM 40K	2×10^9	5×10^9	950 GB
LUBM 50K	28×10^8	7×10^9	1.2 TB

Table 2. Graph Processing Platform

System	Version	Hadoop version	Java version
Apache Giraph	1.1.0	2.6.0	7
GraphX	1.3.0	2.6.0	7
GPS	Revision 112	0.20.203.0	7
GraphLab	1.3	2.6.0	7

We can clearly see that apart from GraphChi the performance of all the graph systems are similar in given datasets and in almost all the cases the processing time is higher in comparison to the loading time. In terms of execution time below is the graphical representation of the performance of the given systems [4]:

We made our evaluation based on all the above papers and we can finally conclude that as the size of the graph or the overall complexity increases the computation time, reading time, writing time, CPU utilization, RAM usage and Network traffic also increases as seen in the graph below:

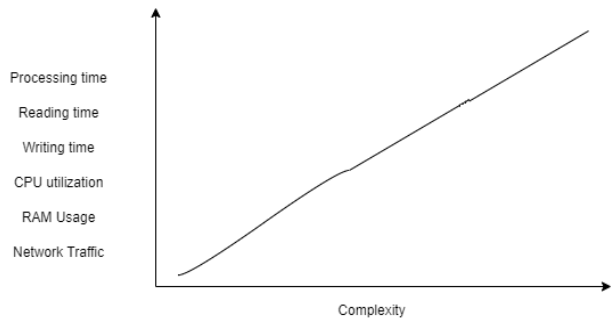


Fig 2.7 Time Complexity Comparison

2.6 Comparison with Relational Database

Surajit Medhi [12] presented a comparative analysis between graph databases and relational databases. Relational databases consist of tabular structure where each row and columns represent records and attributes respectively as shown in the figure x. In the given example, the first table consists of employee ID and name and the second table provides a relationship which is used to identify the data objects. The concept of primary key and foreign key is used to establish relationships between data objects. Graph databases on the other hand uses graph structure with nodes, edges and properties to represent and to store data as shown in the figure x. Here, nodes represent entities such as Users, source, Facebook, hashtag and so on. Edges are the relationship such as reply, contains, mentions and so on and properties are the information associated with nodes i.e. for users we have id, name and age. This is the structural difference between relational and graph databases. From the above properties of both databases, we can come to the conclusion that both databases are better than one another in different ways. As we know that the relational database shows results in the form of rows and columns this can be very easy to understand. Graph database, on the other hand, exhibits data or information in the form of any of the three different types of graphs. And sometimes it becomes very hard to understand graphs when there is huge data associated with it. The relational database is only concerned with data and not with a structure which can improve the performance of the model. On the other hand, Graph database is more flexible than Relational database. It exhibits higher performance for complex deep analytics while the Relational database has poor performance for deep analytics. Graph databases are more flexible than Relational databases in another manner, as it exhibits higher performance for complex transactions than Relational databases. As for the dynamic system where data are frequently changing over time, graph databases provide more support for computation and management, but relational databases suffer in performance in such dynamic systems.

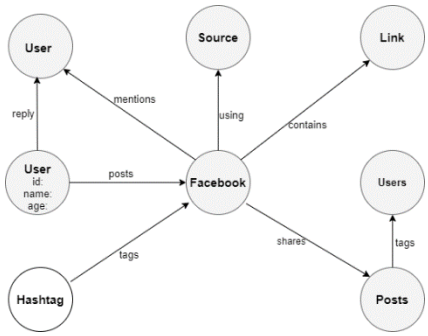


Fig 2.8 A simple Graph Database

Table 3. Example of relational database

Employees	
ID	Name
1	Sudip
2	Darren
3	Dhakal

Relationships	
ID	Rel_ID
1	R1
2	R3
2	R4
3	R3

2.7 Limitation of Graph Databases

Despite the long-term research in graph database systems, their modeling, computation and management, there are still many limitations that remain open in graph data management. In this section we quickly go through the limitations of graph databases based on all the papers that we have surveyed. The limitation of the graph databases are as follows:

Lack of high-performance concurrency: In graph databases there are multiple reader but single writer types of transactions. This mostly hinders the concurrency and performance of the system as a whole.

Lack of standard language: As we have discussed earlier there are multiple programming languages that are used for graph database system modeling. Some of the graph data processing platforms are based on Java while others are on C++ or Python, but a well-established and standard language is still not defined which creates serious optimization issues.

Lack of parallelism: In the case of very large graphs, the shared parallel queries are not possible. This creates a huge problem in overall partitioning of the graph as a whole.

High cost of some queries: Most of the real-world graphs such as social network graphs are highly dynamic and often generate large amounts of data at a very rapid speed. One challenge here is how to store the historical trace compactly while still enabling efficient execution of point queries and global or neighborhood-centric analysis tasks

Heterogenous and uncertain graph data: There is still a lot of work to be done in order to find automated methods for handling the heterogeneity, incompleteness and inconsistency between different data sets. There is no standard for dealing with such heterogeneous data and hence a lot of problems arise when dealing with them.

Similarly, based on the online survey performed in [1], which included 89 users of 22 different software products, with the goal of answering 4 high-level questions:

(i) What types of graph data do users have?

(ii) What computations do users run on their graphs?

(iii) Which software do users use to perform their computations?

(iv) What are the major challenges users face when processing their graph data?

The major findings of the survey include the following:

Variety: Graphs in practice represent a very wide variety of entities, traditional enterprise data comprised of products, orders, and transactions, which are typically seen as the perfect fit for relational systems

Ubiquity of Very Large Graphs: Many graphs in practice are very large, representing a large scale of entities for big organizations.

Challenge of Scalability: The ability to process very large graphs efficiently seems to be the biggest limitation of existing software.

Visualization: Participants indicated visualization as their second most pressing challenge, tied with challenges in graph query languages.

Prevalence of RDBMSes: Relational databases still play an important role in managing and processing graphs.

2.8 Other Challenges

M. Besta [6] presented various challenges in terms of graph database design which we will discuss in this section. With the evolvement of scalable data the challenges in designing graph databases has also increased over the years. First, establishing a single graph model for these systems is far from being complete Second, a clear identification of the most advantageous design choices for different use cases is yet to be determined Third, while there exists past research into the impact of the underlying network on the performance of a distributed graph analytics framework [14], little was done into investigating this performance relationship in the context of graph database workloads Moreover, contrarily to the general static graph processing and graph streaming, little research exists into accelerating graph databases using different types of hardware architectures, accelerators, and hardware-related designs [6].

2.9 Comparison

1. The [1] and [8] papers survey the current and future challenges of graph databases. In the [1] paper, they conducted an online survey across 89 users of 22 different software products and found present graphs use a variety of entities and are very large which is a major challenge. But in the [8] paper, they first survey the current system approaches for management and analysis of big graph data. Then, this paper discussed the distributed graph processing systems, such as Google Pregel and its variations. Next, they discussed the Graph data flow approaches based on Apache Spark and Flink. And they present a framework called GRADOOP, this mode supports analyzing not only single graphs but also collections of graphs. Finally, they discuss current and future research challenges which are similar in most cases, but the challenges put forward by paper one seems more comprehensive and through.

2. The [2] paper talks about a parallel system of graph computation They have verified that GRAPE achieves comparable performance to the state-of-the-art graph systems for various query classes, and that (bounded) IncEval reduces the cost of iterative graph computations. One of the challenges as put forward by the paper [1] and [8] is that graph databases have poor performance when it comes to handling parallel systems or data. This paper can provide a solution to this challenge as GRAPE is concerned with computation in parallel systems.

3. The [4] paper provides an experimental evaluation of performance of the existing graph systems such as Apache Giraph, Giraph, GPS and so on. GRAPE discussed in the [2] paper on the other hand outperforms all these existing systems in terms of performance.
4. The [5], [11] and [14] papers discuss the modelling of graph databases. In the [5] paper, they present current possibilities and challenges in graph database modelling. Also, a conceptual level of a graph database design is considered. In the [11] paper, they introduce a native and generic graph database model, which is GRAD. Unlike the traditional graph database models that support simple graph structures and constraints, GRAD supports advanced graph structures. And in the [14] paper, they propose a method for modeling graph databases, especially, creating a schema for graph databases based on a conceptual schema of the application domain. They use Entity-relationship diagram (ERD) as the schema, and it provides a two-step process for mapping the ERD to a graph database schema (GDBS). First step is adjusting the ERD, the second step is Mapping the Adjusted ERD to the graph database schema.
5. In The [9] paper, they provided an overview of the Graph Database Management systems and Graph Processing Systems. They explained all the systems (including Neo4j, HyperGraph DB, DEX, Trinity, Infinite Graph and Titan etc.) which they listed, this paper pretty similar as paper [7].
6. In the paper [12], due to the relational databases it is difficult to work with a large number of joining tables. To solve this problem, one of the best solutions is to use graph databases for storing data. In this paper, they did experiments to compare between relational databases and graph databases, the result shows graph databases got much better performance.
7. In the [13] paper, they have a comprehensive introduction of Graph Data Models, Query languages, Indexing and Engines. They also list the advantages of graph processing systems.

Some of the papers such as [1], [4], [6], [7], [10] are very well organized, structured and comprehensive. They describe the approach and also their survey in a thorough manner which facilitates a better understanding for a normal reader. However, other papers such as [2], [3], [11], [12] have many grammatical errors as well as unstructured components.

Table 4. Comparison of all the 14 papers

Papers	Well-Structured	Grammatical Errors	Topics Covered	Performance	Overall Contribution
1	Yes	Some	GDS,Challenges	High	Good
2	No	Many	GRAPE	Very High	Average
3	No	Some	Query Lang.	-	Average
4	Yes	No	GPS	High	Good
5	No	Some	DB Modeling	-	Average
6	Yes	No	All	-	Good
7	Yes	No	GDM Tech.	High	Very Good
8	Yes	No	GPS Challenges	High	Good
9	Yes	No	GDBMS, GPS	High	Good
10	Yes	No	GDM, Query, GDMS, GPS	High	Very Good
11	No	No	GRAD, Modelling	-	Average
12	Yes	No	Neo4j, Mysql, Query	High	Good
13	Yes	No	GDM, Query Lang.	High	Good
14	No	No	ERD, Modelling	-	Good

3. Conclusion

Although graph databases are still in its initial stage of development, we can already see its application in both the academia and industry for several purposes and domains. As more scalable systems come into existence this trend is expected to continue further. In this paper, we provided a general overview of the entire graph database family with reference to 14 different papers. From introduction to graph databases and comparing them with relational or traditional databases these papers discussed the prevalent graph database modes, graph processing frameworks, limitations and future challenges in handling them. Taking reference from all these papers we can come to the conclusion that there is still a lot to do in the graph database domain. Although graph databases can be applicable for large database systems such as computer networks, social media, medical, academia and so on, managing the ever-growing data and information is not as simple as we think. Choosing the right platform for graph database computation is very important and establishing a industry standard language for modeling and handling the graph database as a whole is a major challenge in this field. There is still a lot of research that needs to be done to achieve perfection in this field.

References

[1] S. Sahu, A. Mhedhbi, S. Salihoglu, J. Lin, and M. T. Ozsu. *The ubiquity of large graphs and surprising challenges of graph processing*. PVLDB, 11(4):420–431, 2017.

[2] W. Fan, J. Xu, Y. Wu, W. Yu, J. Jiang, Z. Zheng, B. Zhang, Y. Cao, C. Tian. *Parallelizing Sequential Graph Computations*. International Conference on Management of Data, pp.495 - 501, 2017.

[3] R. Angles, M. Arenas, P. Barcelo, A. Hogan, J. Reutter, D. Vrgoc. *Foundations of Modern Query Languages for Graph Databases*. ACM Computing Surveys, Vol.50, No.5, 2017.

[4] O. Batarfi, R. El Shawi, Ayman G. Fayoumi, R. Nouri, Seyed-Mehdi-Reza Beheshti, A. Barnawi, S. Sakr. *Large scale graph processing systems: survey and an experimental evaluation*. Cluster Computing, pp.1189 - 1213, 2015.

[5] Jaroslav Pokorný. *Conceptual and Database Modelling of Graph Databases*. International Database Engineering and Applications Symposium, pp.370 - 377, 2016.

[6] M. Besta, E.l Peter, R.t Gerstenberger, M. Fischer, M. Podstawski, C. Barthels, G. Alonso, T. Hoefler. *Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries*. Cornell University arXiv, 1910.09017, 2019.

[7] Patil N. S., Kiran P, Kiran N. P., Naresh Patel K. M. *A Survey on Graph Database Management Techniques for Huge Unstructured Data*. International Journal of Electrical and Computer Engineering, pp.1140 - 1149, 2018.

[8] M. Junghanns, A. Petermann, M. Neumann, E. Rahm. *Management and Analysis of Big Graph Data: Current Systems and Open Challenges.* Handbook of Big Data Technologies, pp.457 – 505, February 2017.

[9] M. Nolé , C. Sartiani. *Graph Management Systems: A Qualitative Survey*. APTIKOM Journal on Computer Science and Information Technologies, Volume:5, Issue:1, pp.66-76, 2018.

[10] R. Angles, C. Gutierrez. *An introduction to Graph Data Management*, Graph Data Management, pp.1 – 32, January 2018.

[11] A. Ghrab, O. Romero, S. Skhiri, A. Vaisman, E. Zimányi. *GRAD: On Graph Database Modeling*. arXiv: Databases, Volume abs/1602.00503, 2016.

[12] S. Medhi, H. K. Baruah. *Relational database and graph database: A comparative analysis*. Journal of Process Management – New Technologies, International, Vol. 5, No.2, 2017.

[13] M. Paradies, H. Voigt. *Big Graph Data Analytics on Single Machines – An Overview*. Datenbank-spektrum, Volume: 17, Issue:2, pp. 101 -112, 2017.

[14] Noa Roy-Hubara, L. Rokach, B. Shapira, P. Shoval. *Modeling Graph Database Schema*. IT Professional, Volume: 19, Issue: 6, pp. 34 – 43, 2017.

MMTC OFFICERS (Term 2020 — 2022)

CHAIR

Jun Wu
Fudan University
China

STEERING COMMITTEE CHAIR

Joel J. P. C. Rodrigues
Federal University of Piauí (UFPI)
Brazil

VICE CHAIRS

Shaoen Wu (North America)
Illinois State University
USA

Liang Zhou (Asia)
Nanjing University of Post and Telecommunications
China

Abderrahim Benslimane (Europe)
University of Avignon
France

Qing Yang (Letters & Member Communications)
University of North Texas
USA

SECRETARY

Han Hu
Beijing Institute of Technology
China

STANDARDS LIAISON

Weiyi Zhang
AT&T Research
USA

MMTC Communication-Frontier BOARD MEMBERS (Term 2016—2018)

Danda Rawat	Director	Howard University	USA
Sudip Misra	Co-Director	IIT Kharagpur	India
Guanyu Gao	Co-Director	Nanjing University of Science and Technology	China
Rui Wang	Co-Director	Tongji University	China
Lei Chen	Editor	Georgia Southern University	USA
Tasos Dagiuklas	Editor	London South Bank University	UK
ShuaiShuai Guo	Editor	King Abdullah University of Science and Technology	Saudi Arabia
Kejie Lu	Editor	University of Puerto Rico at Mayagüez	Puerto Rico
Nathalie Mitton	Editor	Inria Lille-Nord Europe	France
Zheng Chang	Editor	University of Jyväskylä	Finland
Dapeng Wu	Editor	Chongqing University of Posts & Telecommunications	China
Luca Foschini	Editor	University of Bologna	Italy
Mohamed Faten Zhani	Editor	University of Quebec	Canada
Armir Bujari	Editor	University of Padua	Italy
Kuan Zhang	Editor	University of Nebraska-Lincoln	USA
Bin Tan	Editor	Jinggangshan University	China