A Mobile Robot Generating Video Summaries of Seniors' Indoor Activities

Chih-Yuan Yang
National Taiwan University
Taipei, Taiwan
yangchihyuan@csie.ntu.edu.tw

Srenavis Varadaraj
Intel Labs, Intel Technologies India Pvt. Ltd.
Bangalore, India
srenivas.varadarajan@intel.com

Heeseung Yun
Seoul National University
Seoul, Korea
terry9772@snu.ac.kr

Jane Yung-jen Hsu
National Taiwan University
Taipei, Taiwan
yjhsu@csie.ntu.edu.tw

MobileHCI'19, October 1–4, 2019, Taipei, Taiwan

ACM ISBN 978-1-4503-6825-4/19/10...$15.00
https://doi.org/10.1145/3338286.3344401

Abstract
We develop a system which generates summaries from seniors' indoor-activity videos captured by a social robot to help remote family members know their seniors' daily activities at home. Unlike the traditional video summarization datasets, indoor videos captured from a moving robot poses additional challenges, namely, (i) the video sequences are very long (ii) a significant number of video-frames contain no-subject or with subjects at ill-posed locations and scales (iii) most of the well-posed frames contain highly redundant information. To address this problem, we propose to exploit pose estimation for detecting people in frames. This guides the robot to follow the user and capture effective videos. We use person identification to distinguish a target senior from other people. We also make use of action recognition to analyze seniors' major activities at different moments, and develop a video summarization method to select diverse and representative keyframes as summaries.

Author Keywords
Mobile Robot; Video Summary; Senior; Indoor Activity

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Figure 1: System architecture. Due to the limited computational resources available on a robot, we extend them by a high-performance computer. We transmit data between the robot and the computer via the wireless connection to ensure the robot's moving capability.

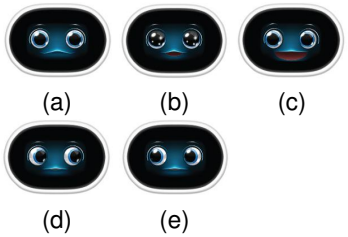


Figure 2: Facial expressions shown on our robot. (a) default_still, if no human is observed. (b) expecting, if the robot is seeing a user's back. (c) active, if the robot is seeing a user's eyes. (d)(e) aware_left and aware_right, when the robot is turning left or right to look for a user.

Introduction

With a large portion of the population becoming aged, we aim to investigate the feasibility of applying video summarization techniques using a social robot to help family members care about seniors living alone. Numerous video summarization methods and interactive robots have been independently developed, but they were never put together. Thus, we investigate their limitations and propose our solution to the widely growing demand.

Related Work

Our study covers the topics of mobile robots and video summarization because we use a mobile robot to capture videos and analyze the videos to generate summaries.

Autonomous Mobile Robots are capable of navigating an uncontrolled environment and moving around to carry out given tasks. If those robots interact and communicate with humans and their tasks are to improve the quality of users' life, they belong to social robots. Many studies have shown that social robots are good tools to meet the elderly individual needs and requirements [7]. With the advance of technology, social robots have been extended from zoomorphic robots to humanoid mobile robots equipped with multiple sensors and advanced intelligence [2, 11]. In this paper, we assign a new task of generating video summaries to social robots and develop a solution.

Video Summarization methods analyze videos to create summaries. It is part of machine learning and data mining. The main idea of summarization is to find a diverse and representative subset of the entire input data. Several summary formats are available including keyframes, skimmed videos, time-lapsed videos, and video synopsis. Among them, the keyframes are widely used because they are simple and ease to consume [3, 12].

System Architecture

We use a commercial robot Zenbo, as shown in Figure 1, to capture videos. It uses an ultra-low-voltage CPU to reduce power consumption, but the processing power of the CPU is insufficient to analyze videos in real time. Thus, we transmit captured frames via Wi-Fi to a powerful computer to analyze them and return results to the robot.

The robot has a touch screen on its head, which serves as an input UI and also shows facial expressions for human-computer interaction. The robot's camera is at the upper center boundary of the screen. The robot's OS is a modified Android system with a set of additional APIs to retrieve sensor data and control the robot's movements, neck gestures, and facial expressions. There are 24 built-in expressions rendered by OpenGL. We select five of them, as shown in Figure 2, to interact with users. Our software implementation includes three parts, an Android-based app running on the robot to transmit captured frames to a server and receive analyzed results from the server to control the robot's actions, a C++ program running on the server to analyze images, and an offline Python program to generate summaries by selecting keyframes from all captured frames.

Human Detection

Because our summaries focus on human activities, we utilize existing human detection algorithms to find our human targets. Figure 3 shows the relationship among those algorithms. Given an input video frame, we use a real-time pose estimation algorithm OpenPose [1] to find people. We use a pose estimation method rather than a pedestrian or object detection method because we need human body landmark coordinates to control our robot's camera view angle. We want our robot to capture well-posed frames so we can generate high-quality summaries. In contrast, bounding boxes reported by a pedestrian or object detection method do not

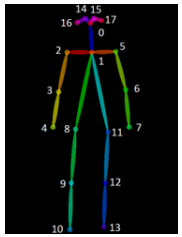


Figure 4: Landmark points.

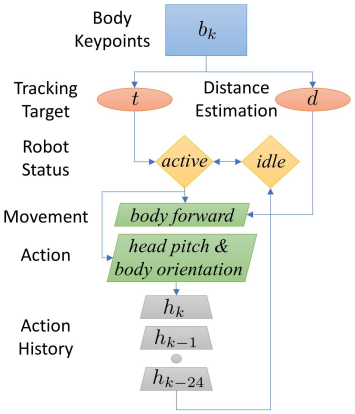


Figure 5: Robot-side data flow for movement control.

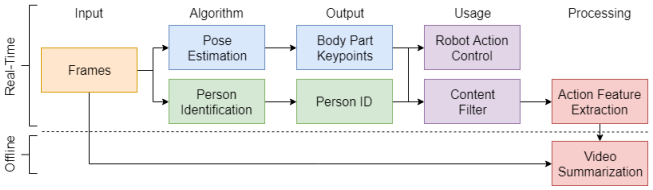


Figure 3: Server-side flowchart of data analysis

analyze human body parts. Because pose estimation methods do not discern people's identities, we use a person identification method PSE-ECN [9] to prevent false-positives and determine our target person among people on an image.

Rule-based Robot Movement Control

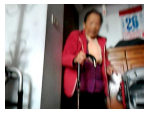
OpenPose reports 18 body landmark coordinates as shown in Figure 4, but some may be unavailable if the body parts are invisible. Regarding our target person's coordinates, if any facial landmark point is available, we adjust the robot's body orientation and neck pitch angle to change the robot's camera view angle and move the landmark point(s) towards the central upper part of the upcoming frames to create well-posed ones for summarization. If all facial landmark points are invisible but the neck point is visible, we raise the robot's head to look for a face. In addition, we also use landmark points to estimate the distance between our robot and the target user to control the robot's forward movement. We compute the average distances from the neck point to the two hip points because they are the longest connected distances and they distort less than other body parts do in different poses. We move our robot toward the target person until 2 meters apart, which is a proper distance to take images with adequate human size. When our robot cannot detect a person, it will turn left or right 30 degrees depending on the last position of a person visible in past frames. If the robot turns around but still cannot detect a person, we set the robot's neck pitch angle 15 degree vertically because this degree ensures the robot to detect distant people. If the robot turns around twice but still does not detect a person, we let the robot wait in an idle mode for 15 minutes to save energy until a person appears in front of its camera. Figure 5 shows the overall flow of movement control.

Content Filter

Our images are captured from a moving camera in an indoor space so many of them are blurred and improper to be selected as summaries. To remove them, we use the variance-of-Laplacian [8] method and set a threshold. We also use the aforementioned OpenPose and PSE-ECN methods to ignore ill-posed frames, including the ones without people or with people but too small, cropped, at corners, or whose faces are invisible. Figure 6 shows examples for 6 cases.

Video Summarization

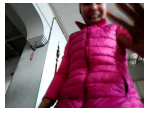
We use keyframes as our summary format due to its efficiency to consume information at a glance. We propose a method to select keyframes by temporally clustering well-posed frames and then selecting one representative frame out of a cluster in terms of human actions because we expect the summary to not only diversely cover seniors' daily activities but also show the representative ones. To do it, let $\{v_i\}$, $i \in (1,n)$, be the well-posed frames and $\{t_i\}$ be their timestamps, $t_i < t_{i+1}$. Let k be the number of keyframes in our summary. We group $\{v_i\}$ into at least k clusters $\{C_j\}$, $j \in (1,m)$ and $m \geq k$. Initially we let C_1 contain the first frame v_1. For any other frames v_i, $i \in (2,n)$, we assign their clusters by the temporal difference from its previous frame, i.e., assume $v_{i-1} \in C_j$, we assign $v_i \in \begin{cases} C_j & \text{if } t_i - t_{i-1} < h; \\ C_{j+1} & \text{else,} \end{cases}$ where h is a temporal gap threshold. In order to produce at least k clusters, We iteratively adjust $h = \begin{cases} 2h & \text{if } m \geq k; \\ \frac{h}{2} & \text{else,} \end{cases}$



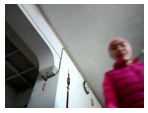
(a) Blurred

(b) Eyes invisible

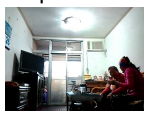
(c) People absent



(d) Forehead cropped



(e) People at corners



(f) People too small

Figure 6: Examples of ill-posed frames, which are removed by our content filter.

until $m(h) \geq k$ and $m(2h) < k$ where $m(h)$ is the number of clusters determined by h. If $m(h) > k$, we disregard the small $m(h) - k$ clusters and only use the k large ones.

We extract frame features as the probabilities of 157 predefined indoor actions generated by a SqueezeNet [5] model pre-trained on the Charades dataset [10], which aims to recognize human actions from a single frame. We compute the mean features of a cluster, and select the frame with the closest distance to the cluster mean, as the cluster's keyframe.

Experiments

We conduct experiments in three families, and their statics are shown in Table 1. We set the keyframe number k as 8 and the initial threshold h as 60 seconds. We compare the proposed video summarization method with three existing ones: VSUMM [3], DPP [4], and DR-DSN [13]. All of them are programmed in Python and run on a machine equipped with a 3.4GHz quadratic core CPU and their execution time is shown in Table 2. We use the publicly available code of OpenPose implemented in OpenVINO [6] but temporally disable PSE-ECN because we have not fully integrated it into our program.

Qualitative comparisons are enclosed in Figure 7. A set of 8 key frames (a-h) selected by the 4 different methods are shown on the right while the histogram of well-posed frames and the time of keyframes selection are shown on the left. The proposed method prevents the problem of consecutive similar keyframes arising in other methods because it exploits timestamps and a threshold h to partition frames into temporally disjoint clusters, which are separated from each other with a gap that tends to isolate different activities. In contrast, VSUMM and DPP ignore temporal information and their keyframes are not sufficiently diverse. DR-DSN generates summaries containing diverse keyframes but neglects representative events of eating a lunch box in video 1, and sitting on a sofa in videos 2 and 3. The proposed method generates diverse and representative summaries, but we have not fully implemented the person identification component, which results in a person on TV shown in video 3's summary, as the images 12(g)(h) of Figure 7. Our code is available at https://github.com/yangchihyuan/RobotVideoSummary.

Video	Subjects	Duration	#Frame (Total/well-posed)
1	male (79) female (74) female (41)	6h 46m	198689 / 8093
2	female (94) female (31)	1h 44m	50634 / 19971
3	female (70)	7h 42m	191183 / 11563

Table 1: Statistics of experimental videos. The subjects' ages are shown beside their genders.

Method	VSUMM	DPP	DR-DSN	Proposed
Time	11.05s	11m 32s	11m 27s	**0.18s**

Table 2: Execution time for video 2. The execution time for videos 1 and 3 is proportional to their frame numbers.

Conclusion and Future Study

The paper presents an effective method to generate video summaries using a social robot for family members to care about seniors living alone. We use a pose estimation method to detect humans to control the robot's movements to capture well-posed frames. We use human pose and image quality information to disregard ill-posed frames and develop a summarization method to generate diverse and representative summaries. Experimental results show that our summaries prevent the problems of redundancy and unrepresentative keyframes generated by existing methods.

Our immediate plan is to integrate a person identification algorithm into our system to ensure the robot to keep track on a target user. During our experiment, our users express strong demand for fall detection and immediate notification, which are important features. In addition, we plan to take preferences into account to create personalized summaries.

Acknowledgements

This research was supported in part by the Ministry of Science and Technology of Taiwan (MOST 108-2633-E-002-001, 107-2218-E-002-009, 107-2811-E-002-575), National Taiwan University (NTU-108L104039), Intel Corporation, Delta Electronics and Compal Electronics.

REFERENCES

1. Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *CVPR*.
2. Chaona Chen, Oliver G. B. Garrod, Jiayu Zhan, Jonas Beskow Philippe G. Schyns, and Rachael E. Jack. 2018. Reverse Engineering Psychologically Valid Facial Expressions of Emotion into Social Robots. In *FG*.
3. Sandra Eliza Fontes de Avila, Ana Paula Brand ao Lopes, Antonio da Luz Jr., and Arnaldo de Albuquerque Araújo. 2011. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters* 32, 1 (2011), 56–68.
4. Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. 2014. Diverse Sequential Subset Selection for Supervised Video Summarization. In *NIPS*.
5. Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *arXiv:1602.07360* (2016).
6. Intel. 2019. Distribution of OpenVINO Toolkit. `https://software.intel.com/en-us/openvino-toolkit`. (2019).
7. Tineke Klamer and Soumaya Ben Allouch. 2010. Acceptance and use of a social robot by elderly users in a domestic environment. In *Proceedings of IEEE International Conference on Pervasive Computing Technologies for Healthcare*.
8. José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-Martínez, and Joaquín Fernández-Valdivia. 2000. Diatom Autofocusing in Brightfield Microscopy: a Comparative Study. In *ICPR*.
9. M. Saquib Sarfraz, Arne Schumann, Andreas Eberle, and Rainer Stiefelhagen. 2018. A Pose-Sensitive Embedding for Person Re-Identification with Expanded Cross Neighborhood Re-Ranking. In *CVPR*.
10. Gunnar A. Sigurdsson, Göl Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *ECCV*.
11. Zheng-Hua Tan, Nicolai Bæk Thomsen, Xiaodong Duan, Evgenios Vlachos, Sven Ewan Shepstone, Morten Højfeldt Rasmussen, and Jesper Lisby Højvang. 2018. iSocioBot: A Multimodal Interactive Social Robot. *IJSR* 10, 1 (2018), 5–19.
12. Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video summarization with long short-term memory. In *ECCV*.
13. Kaiyang Zhou, Yu Qiao, and Tao Xiang. 2018. Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity-Representativeness Reward. In *AAAI*.

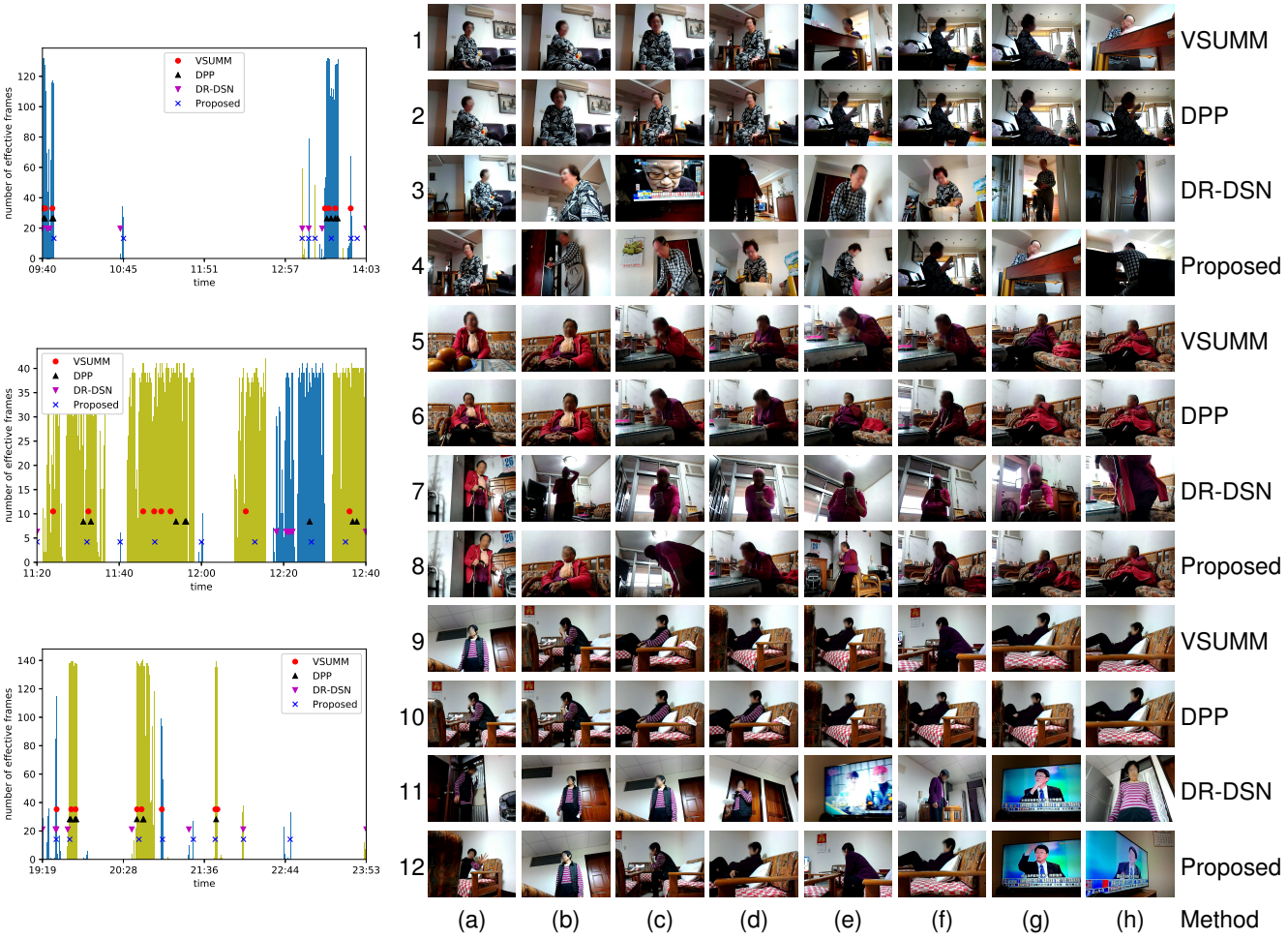


Figure 7: Distribution charts and keyframes. From top to bottom: Videos 1, 2, and 3. At left: Histograms of well-posed frames of the three videos in 1000 bins. We draw the bars in two alternative colors to show the clusters generated by the proposed method. The numbers of clusters of the three videos are 11, 8, and 12. Note that the compared methods are not affected by the clusters and some clusters are invisible because they contains too few frames to be drawn under the chart resolution. The marks indicate the time of the keyframes and we use evenly separated heights for the ease of observation. At right: selected keyframes arranged in temporal order. The proposed method does not generate repeating keyframes such as 1(ab), 1(fg), 2(efgh), 5(cd), 6(ef), 7(de), 7(gh), 10(de), 11(cd), and 11(efg). We blur participants' faces to protect their privacy.