Senbay: A Platform for Instantly Capturing, Integrating, and Restreaming of Synchronized Multiple Sensor-Data Stream

Yuuki Nishiyama
University of Oulu
90570 Oulu, Finland
yuuki.nishiyama@oulu.fi

Takuro Yonezawa
Keio University
Endo 5322, Fujisawa,
Kanagawa, Japan
takuro@ht.sfc.keio.ac.jp

Anind K. Dey
University of Washington
anind@uw.edu

Jin Nakazawa
Keio University
Endo 5322, Fujisawa,
Kanagawa, Japan
jin@ht.sfc.keio.ac.jp

Denzil Ferreira
University of Oulu
90570 Oulu, Finland
denzil.ferreira@oulu.fi

Abstract
The spread of smartphones allows us to freely capture video and diverse hardware sensors' data (e.g., accelerometer, gyroscope). While recording such data is relatively simple, it is often challenging to share and restream this data to other people and applications. Such capability is very valuable for a range of applications such as a context-aware prototyping/developing platform, an integrated data recording and analysis tool, and a sensor-data based video editing system. To enable such complex operations, we propose Senbay, a platform for instant sensing, integrating, and restreaming multiple-sensor data streams. The platform embeds collected sensor data into a video frame using an animated two-dimensional barcode via real-time video processing. The video-embedded sensor data, dubbed Senbay Video, can be easily restreamed to other people and reused by data rich, context-aware applications.

Author Keywords
Sensor-data embedded video; multiple sensor-data streams, extensible platform, data restreaming.

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

http://acm.org/about/class/1998 for the full list of ACM classifiers. This section is required.

Introduction

With the rapid spread of smartphones, user-produced videos have become a common "digital recording media" in our daily lives. We can instantly record a physical space in the form of video media, which is composed of multiple images and audio, using an off-the-shelf smartphone camera. In addition to the camera, a variety of sensors (e.g., motion, location, and environmental sensor) are built-in to the smartphone for recording the physical world in the form of sensor data. While the recording of such data is relatively simple, it is often tedious to share and restream sensor data as shown in Figure 1, to other people and applications, despite such capabilities being very valuable for a range of applications such as a context-aware system prototyping/developing platform, integrated data record/analysis tool, and sensor-data based video editing system.

Existing systems require users to perform tedious and often complex operations to replay sensor data with a synchronized video. In particular, while existing context-aware platforms [1, 2, 3, 4, 5] support the rapid collection/visualization of sensor data from smartphones, these systems do not support synchronization with captured video: users have to manually perform operations for using a video and time synchronize the data manually. Taking a different approach, research focused on augmenting video [6, 7] has embedded sensor data into a video. As these systems were designed for their specific use cases, they are neither extensible nor do they make it easy to restream-able. Finally, Integrated Development Environments (IDEs) have functions for developing applications using video input and sensor data at the same time on a personal computer [8]. However, importing a video and sensor data from smartphone, and restreaming the data requires users to perform manual data operations. To the best of our knowledge, a platform that supports instantly capturing, integrating, and restreaming a time-synchronous video and sensor data, does not exist.

We propose a novel platform for instantly capturing, integrating and restreaming multiple-sensor data stream, named Senbay (short for Sensors Bay). The platform is based our previous work regarding video and sensor-data federation system [7, 9, 10]. The previous system can generate a sensor-federated video on the smartphone and PC. However, we did not design for extending flexibly to expanding opportunities of the sensor data federated video. Here, we now allow others to generate a sensor-federated video on a smartphone. Furthermore, Senbay provides an extensible sensing and restreaming API for connecting other devices and append their data. Our performance evaluation and user study demonstrate that Senbay significantly reduces the time in operations for data collection, data-visualization with a synchronized video, and reuse of the data by supporting seamless restreaming of the data to other tools, and it provides new opportunities to easily and rapidly leverage multiple synchronized sensor streams augmented with video.

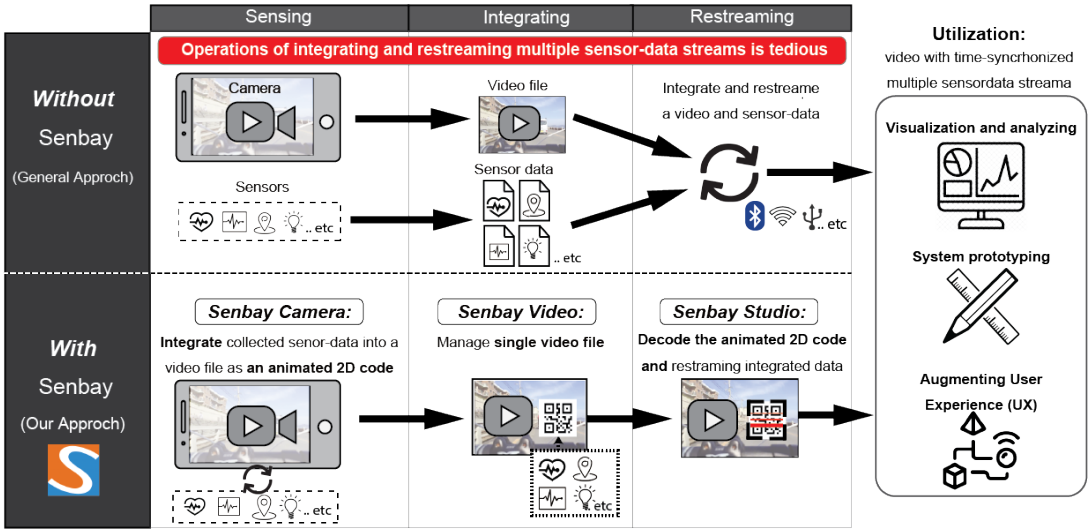


Figure 1. Overview of Senbay Platform

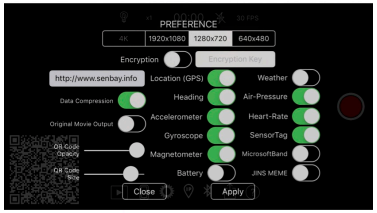
Design

To address these issues and satisfy the identified system requirements, we propose Senbay, a smartphone-based platform that supports rapid data capturing, sharing, and restreaming of sensor-data embedded in a video. Senbay embeds collected sensor data using an animated QR code added to a video. We refer to the generated video as Senbay Video, which has sensor data associated with each video frame, and can be stored as a simple video file (i.e., MOV format).

As shown in Figure 2, Senbay is composed of Senbay Video generator and reader modules. Senbay Camera generates the sensor data federated video (Senbay Video) in a flexible format (Senbay Format) that supports the embedding of data on the video. Users can share Senbay Videos via existing video integrating methods (e.g., email, social network services, and video sharing services). A user can decode the animated QR code by using Senbay Studio (on laptops/desktops/mobile phones).

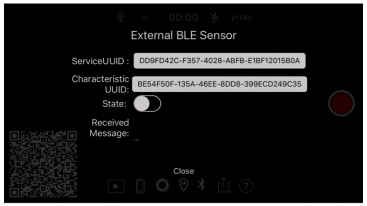
Sensing: Senbay Camera

Users start by pressing the record button to enable Senbay Camera to generate a sensor-data federated video on the iOS platform. This can be easily extended to leverage not only sensors built-in into the iOS device, but also external sensors. Figure 3 and 4 shows screenshots of Senbay Camera application. Senbay Camera encodes sensor data streams as an animated QR code into a video. As shown in Figure 2, the procedure of generating Senbay Video with Senbay Camera works as follows. Firstly, an image is captured by the camera of a smartphone. Then the captured images are



(a) General sensor and camera setting view

(b) Socket connection setting view



(c) Bluetooth LE setting view

Figure 4. Screenshots of Senbay Camera

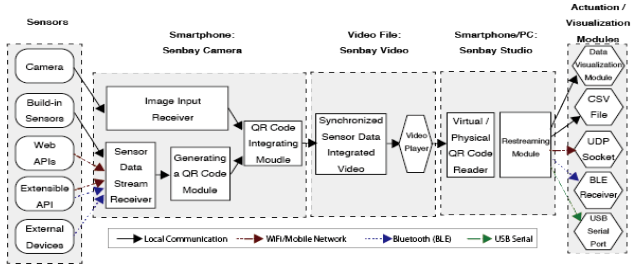


Figure 2. System Design of Senbay Platform

processed to video data. At the same time, a sensor data module receives sensor data using a Key-Value format and compresses it using the Senbay Format (see Figure 3). The QR code generator module generates a QR code from the sensor data. Finally, the mixing module superimposes the generated QR code onto a captured image using OpenGL ES, and saves all processed images to one video file.

Senbay Camera supports multiple types of data sources:

(1) **Built-in Smartphone Sensors**: On an iPhone 7, Senbay Camera can leverage motion sensors (accelerometer, gyroscope, and magnetometer), location sensors (latitude, longitude, and altitude from GPS), and environmental sensors (air-pressure, brightness).

(2) **Web APIs**: By using a network connection on the smartphone, Senbay Camera can obtain data from Web APIs. For example, Senbay Camera can collect current weather information based on a location data via OpenWeatherMap API.

Figure 3. Screenshot of Senbay Camera

(3) **External Bluetooth Sensors**: Bluetooth is a common nearfield wireless protocol for communicating between a smartphone and external devices. Senbay Camera can connect to external sensors via Bluetooth a Texas Instruments SensorTag, Microsoft Band2, Heart-Rate Sensor, JINS MEME.

(4) **Extensible API**: Senbay Camera has an extensible API. There are two ways to use the API: Bluetooth Low Energy (BLE) or WebSocket. BLE is able to advertise and subscribe to its own defined GATT services and characteristics with each Universally Unique Identifier (UUID). Senbay Camera subscribes to the specified GATT services. A Socket allows for the importing of a data stream if the device is in the same Wi-Fi network and has its own IP address via UDP (User Datagram Protocol). A user just sends a UDP packet to the Sebbay Camera's IP and port. The user can append any data stream by using the API as described above.

Integrating: Senbay Video

Senbay Format is a flexible data format for representing data. The format is extensible because it uses a JSON-like Key-Value format (Sample 1). Each video

V:3,TIME:1494064033.645251,LONG:139.425588,LATI:35.388033,ALTI:36.727589,SPEE:1.000000,ACCX:0.4050140380859375,ACCY:0.0925445556640625,ACCZ:0.8967437744140625,HEAD:79.162476,BATT:1.000000,AIRP:100.0032348632812,TEMP:18.843988,WEAT:'Rain', HUMI:90

Sample 1. Sample SenbayFormat

Figure 5. Senbay Studio macOS

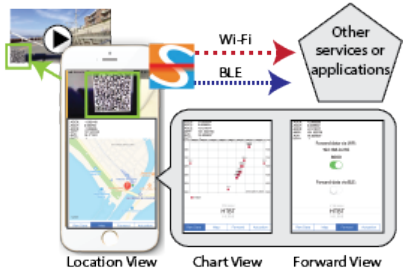


Figure 6. Senbay Studio iOS

frame contains data represented using the Senbay Format and each key-value element represents sensor data. As an example of a numerical value, "TIME" represents a UNIX timestamp and "ACCX" represents x-axis accelerometer data. For text values (e.g., "WEAT" for weather), the data is stored with single quotation marks ('). By using this format, a developer can add additional sensor data.

We embed a QR code into each video frame, taking the same approach as SENSeTREAM [7]. This method guarantees exportability even if the video codec is somewhat changed by other services. Commonly, existing video streaming services, such as Social Network and communication tools, change the video formatting to optimize video file size.

Restreaming: Senbay Studio

When playing a Senbay Video, a user can use Senbay Studio (for iOS and macOS) to re-stream the QR code in the video. They can visualize the decoded sensor data (e.g., using a map, directional arrow and images) on a window as shown in Figures 4 and 5. Senbay Studio on iOS captures the QR code using the camera of a mobile device, while Senbay Studio on macOS uses a virtual desktop camera (QuickTime.app) that can track a QR code automatically.

Flexible data visualization

As shown in Figure 5, Senbay Studio can visualize embedded sensor data in a variety of forms. By default, Senbay Studio supports raw data, maps (standard, satellite, and hybrid mode), direction (0-360), and line charts (to visualize all numerical data) in each view. The view can be created through the Senbay Studio interface.

Embedded data extracting and restreaming

In addition to the data visualizations, Senbay Studio can export embedded data from a Senbay Video. The exported data can either be saved to a CSV file for later user or be restreamed as a JSON object to another application or services enabling a variety of uses. The CSV file can be exported from a video stream on any video player, or the video file itself. Moreover, the restreaming function unicasts/broadcast the exported data as a JSON object to (a) a specified IP address and port via UDP, (b) BLE or (c) USB serial. By simply receiving the UDP packets, a user can utilize the video-synchronized data rapidly.

Evaluation

In addition to our previous evaluation of Senbay Camera/Reader (see [10] for battery consumption, decoding performance, and an usability study), here we measure the **generating speed of Senbay Video** by Senbay Camera on the latest devices and **data robustness of Senbay Video encoding**.

Senbay Video Generation Performance

Table 1 shows the results of the average speed (FPS) of Senbay Video generation for different levels of video resolution. This result shows that an iPhone7 (with resolutions of 1280*720 and 1920*1080) can generate Senbay Video at a rate of 30 FPS, consistent with standard videos.

Table 1. Senbay Video Generation Speed

Device	FPS by each video size		
	640 x480	1280 x720	1920 x1080
iPhone 7	26.07	30	30
iPhone 6s	24.12	29.96	19.98
iPhone 6	29.62	28.03	14.77

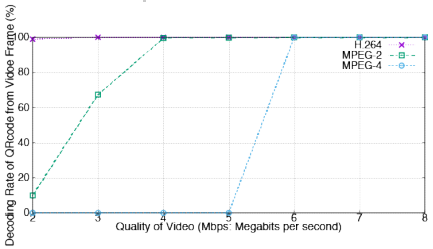


Figure 7. Robustness of Senbay Video by different resolutions and codecs

Figure 8. Overview of *Tacvie*

Robustness of Senbay Video

Figure 7 shows the robustness of Senbay Video under different video resolutions (2-8Mbps) and CODECs (H.264, MPEG-4, and MPEG-2). When the resolution is greater than 6Mbps, Senbay Studio can successfully export 100% of the sensor data from videos with all formats. When the format is H.264, the sensor data can be exported when the resolution is greater than 3Mbps. However, the exporting rate for MPEG-2 rapidly decreases when the resolution is under 6Mbps, and MPEG-4 when under 4Mbps.

Use Case

As an actual use-case, we applied *Senbay* Platform for ***UX media prototyping***. As a class project, a student developed an augmented video player, *Tacvie*, that enhances users' experience by leveraging multiple physical actuators. The student is a 4th-year undergraduate student at our university and has 2-years experience with python and physical computing (using Arduino). The motivation of this project is to reproduce humans have the sense (sight, hearing, touch, smell, and taste) during watching a video.

As an example, a user records a video while riding a bike, and then shares the recorded video. Audiences can view that video and receive the collected physical feedback including vibrations and accelerations. In the system, Senbay Platform sense, integrate data into a video and restream the implemented the data via USB-serial and BLE to Arduino. The student can just force to develop actuation modules. The student was very positive about Senbay Platform and agreed that it was useful for developing his application ("I could sense and use the data very quickly and easily."), and the platform improved the workflow of sensing, integrating, and restreaming ("I could freely replay the video and sensor data without caring about time-synchronization during the development and test phases because the sensor data was encoded into each video frame."). The restreaming function was seen as an important service of Senbay. The student described it well: 'The UDP forwarding function is a very impressive function for developing the UX medias. By using the function, we can connect various types of devices via a network."

Conclusion

This paper has presented a new platform which provides a workflow for integrally capturing, integrating, and restreaming of multiple synchronized sensor-data streams using a data-embedded video, and provided Senbay as an implementation of this workflow. Senbay focuses on reducing the tedious operations of sensing, integrating and restreaming data with a synchronized video to generate new opportunities using sensor data and video. Our platform provides extensible sensing and restreaming APIs for connecting other devices and their data, and for integrating the data streams with other services (e.g., visualizations, analytics, and development) using off-the-shelf devices.

References

1. 2017. HASC Hub. (5 2017). http://hub.hasc.jp/
2. Denzil Ferreira, Vassilis Kostakos, and Anind K. Dey. 2015. Aware: mobile context instrumentation framework. FRONTIERS IN ICT (2015).
3. K. Katevas, H. Haddadi, and L. Tokarchuk. 2016. SensingKit: Evaluating the Sensor Pow-er Consumption in iOS Devices. In 2016 12th International Conference on Intelligent Envi-ronments (IE). 222–225.

4. Kiran K. Rachuri, Mirco Musolesi, Cecilia Mascolo, Peter J. Rentfrow, Chris Longworth, and Andrius Aucinas. 2010. EmotionSense: A Mobile Phones Based Adaptive Platform for Experimental Social Psychology Research. In Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp '10). ACM, New York, NY, USA, 281–290.
5. Haoyi Xiong, Yu Huang, Laura E. Barnes, and Matthew S. Gerber. 2016. Sensus: A Cross-Platform, General-Purpose System for Mobile Crowdsensing in Human-Subject Studies. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '16. ACM Press, New York, New York, USA, 415–426.
6. Ryohei Suzuki, Daisuke Sakamoto, and Takeo Igarashi. 2015. AnnoTone: Record-time Audio Watermarking for Context- aware Video Editing. Proceedings of the ACM CHI'15 Conference on Human Factors in Computing Systems 1 (2015),
7. Takuro Yonezawa, Masaki Ogawa,Yutaro Kyono, Hiroki Nozaki, Jin, Osamu Nakamura, and Hideyuki Tokuda. 2014. SENSeTREAM: Enhancing Online Live Experience with Sensor-federated Video Stream Using Animated Two-dimensional Code. In Proceed- ings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14). ACM, New York, NY, USA, 301–305.
8. Björn Hartmann, Scott R. Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. 2006. Re ective Physical Prototyping Through Integrated Design, Test, and Analysis. In Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06). ACM, New York, NY, USA, 299–308. https://doi.org/10.1145/1166253.1166300
9. M.Ogawa, T.Yonezawa, Y.Nishiyama, J.Nakazawa, and H.Tokuda. 2015. Arobotcontrol system for video streaming services by using dynamic encoded QR codes. In Mobile Computing and Ubiquitous Networking (ICMU), 2015 Eighth International Conference on. 86–87.
10. Yuuki Nishiyama, Tadashi Okoshi, Yin Chen, Takuro Yonezawa, Jin Nakazawa, Hideyuki Tokuda, "Senbay: Smartphone-Based Activity Capturing and Sharing Using Sensor-Federated Video", ACM Ubicomp2015, poster session. Sep.2015