Smartphone-Based Objects Tracking for Motion Detection and Analysis Services

Nikita A. Bazhenov, Dmitry G. Korzun
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{bazhenov, dkorzun}@cs.karelia.ru

***Abstract*—Many digital services can use on well-elaborated methods of video processing for motion detection and analysis. The spectrum of possible application domains is diverse, to name a few: medical environments for patient monitoring and assistance at home, user activity detection and analysis in collaborative work environments, and monitoring of production equipment by nearby personnel. This demo presents how such services are constructed using the smart spaces concept and Internet of Things (IoT) technology. As a basic video recording resource, we use a smartphone (our reference case is Xiaomi Mi 5). Online video stream from such a smartphone is processed to recognize recorded objects. Our example service shows how video capture resources of an edge-centric IoT-environment can be effectively used within the smart space and its construction of advanced digital services.**

Many IoT-enabled application domains need motion detection for constructing advanced digital services: medical environments for patient monitoring and assistance at home [1], user activity detection and analysis in collaborative work environments [2], and monitoring of production equipment by nearby personnel [3]. There exist many methods of pattern recognition in video. They are successfully applied in motion detection based on object tracking, e.g., see [4], [5].

Nevertheless, such existing methods are oriented to specific digital environments with professional video capture equipment. Since IoT is actively comes into our life we need more flexible solutions, which are oriented to everyday devices and personal mobile computing based on smartphones. Our concept vision on the smartphone-based system of object tracking for motion detection is shown in Fig. 1. The system uses ontology-oriented representation of the semantics for interaction and management support.

Service clients (users) with their personal mobile devices (smartphone, tablet, etc.) connect to cameras (e.g., IP cameras). In order to find the IP address of the camera on the network, the video server is requested. After that, the client connects to the camera and access its resources. For the video server, the whole state information (available cameras, connected clients, ongoing information processing on cameras) is needed to be properly updated. Using this reason we introduce an additional entity that would perform all these operations, based on the smart spaces concept [6]. The Semantic Information Broker (SIB) acts as a mediator providing a simple way to interact remotely with other devices and to realize non-trivial processing and data models.

In our implemented demo prototype, we use one smartphone. In particular, we use Smartphone Xiaomi Mi 5 with a recording video camera (aspect ratio 16:9, resolution 1920x1080). Video records are for 5..6 seconds without the HDR function (High Dynamic Range). Each video record is about 20 Mb.

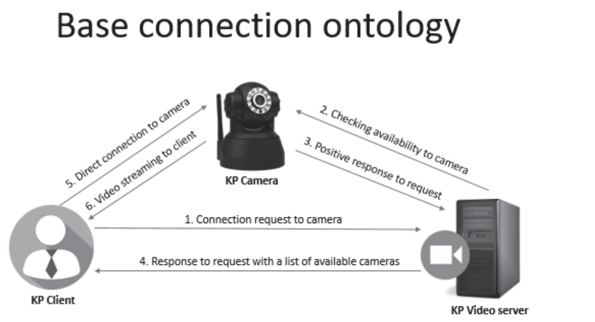

Fig. 1. Vision on smartphone-based object tracking for motion detection

The following scenarios are implemented whereby the user (client) connects to the camera and then uses it for motion detection and analysis services.

1) Scenario 0. Connecting to the camera in the local network using its IP address or to the smartphone:
    - Start the application and choose how to connect to the camera or video.
    - Connecting to the camera.
2) Scenario 1. Receiving a video stream with the ability to record and snapshot and then saving to the device:
    - View live streaming.
    - Recording video from camera.
    - Snapshot from the video camera.
3) Scenario 2. Video analysis and delivery of the result:
    - Uploading video to the app.
    - Processing and analysis of video for some time.
    - Drawing charts and models on video.

Let us consider the interrelationships that should be transmitted and stored.

- Camera—Server: transfer of information about beginning and ending the video stream.
- Server—Camera: start a stream for translation to a specific client.

Fig. 2. Motion detection based on object tracking

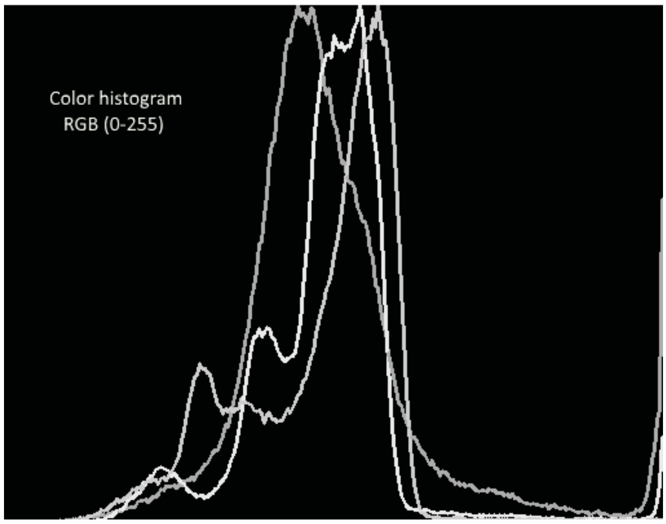

Fig. 3. Observed movement trajectory

- Camera—Client: streaming to the client (e.g., to smartphone).
- Client—Camera: control requests (e.g., rotation of the IP camera).
- Server—Client: information about available cameras and possible connections.
- Client—Server: requests to connect to a given camera.

Our demo prototype development uses Qt Creator. It provides multi-function and cross-platform capability, the ability to develop for several target platforms: Android, iOS, Sailfish OS (in this case, Android). Components of Android (Android SDK, Android NDK, Java and Apache Ant) are connected separately to the development environment, some of them are sent to the CMake compiler, which takes as source code, implemented functions and OpenCV libraries. Then Qt can use the generated libraries for a specific platform, version and internal compiler (MinGW).

In the implementation, all architectural components of the system must be compatible with each other. The application from Android is not easy to transfer to the environment due to the presence of compilation errors and linking files. This interaction of components (Qt + Android + MinGW) is not widespread, so there are not many standard solutions. In OpenCV itself, some algorithms are difficult or even impossible to implement because of full or partial incompatibility and lack of specifically implemented functions.

The client connects to the camera using the personal device (desktop, tablet, smartphone) by entering the IP address, selects the resolution quality, connection type and service. When the camera is connected then the picture becomes displayed in real time. Depending on the selected service, either motion (fact) is detected or faces in the image are recognized.

Examples are shown in Figs. 2 and 3. The height of reflective element (Fig. 2) is evolved in time (measured in frames). Trajectory of object movement (Fig. 3) shows the observation in the width and height directions.

ACKNOWLEDGMENT

The research was financially supported by the Ministry of Education and Science of Russia within project # 2.5124.2017/8.9 of the basic part of state research assignment for 2017–2019. The results are implemented by the Government Program of Flagship University Development for Petrozavodsk State University in 2017–2021.

REFERENCES

[1] A. Meigal, K. Prokhorov, N. Bazhenov, L. Gerasimova-Meigal, and D. Korzun, "Towards a personal at-home lab for motion video tracking in patients with Parkinson's disease," in *Proc. 21st Conf. Open Innovations Association FRUCT*, Nov. 2017, pp. 231–237.

[2] D. Korzun, "Designing smart space based information systems: The case study of services for iot-enabled collaborative work and cultural heritage environments," in *Frontiers in Artificial Intelligence and Applications. Volume 291: Databases and Information Systems IX*. IOS Press, 2016, pp. 183–196.

[3] A. Gurtov, M. Liyanage, and D. Korzun, "Secure communication and data processing challenges in the Industrial Internet," *Baltic Journal of Modern Computing*, vol. 4, no. 4, pp. 1058–1073, 2016.

[4] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 126–128, 2016.

[5] M. Patacchiola and A. Cangelosi, "Comparative efficiency of different regimens of locomotor training in prolonged space flights as estimated from the data on biomechanical and electromyographic parameters of walking," *Pattern Recognition*, vol. 71, pp. 135–137, 2017.

[6] S. Balandin and H. Waris, "Key properties in the development of smart spaces," in *Proc. 5th Int'l Conf. Universal Access in Human-Computer Interaction (UAHCI '09). Part II: Intelligent and Ubiquitous Interaction Environments, LNCS 5615*, C. Stephanidis, Ed. Springer-Verlag, Jul. 2009, pp. 3–12.