ELEKTROTEHNIŠKI VESTNIK 80(1-2): 13-18, 2013
ORIGINAL SCIENTIFIC PAPER

Mobile Vision for Intelligent Commuting: "Interactive Bus Stop" Case Study

Jaka Krivic, Matjaž Jogan[2], Aleš Leonardis

Faculty of Computer and Information Science at University of Ljubljana
[2] Computational Perception and Cognition Laboratory at University of Pennsylvania
E-pošta: jaka.krivic@fri.uni-lj.si

Abstract. In this paper we present a case study of context aware, on-site information retrieval using a computer-vision-based interaction on mobile phones with the goal of facilitating information access for urban commuters that use public transport. Our focus is on intuitive touchless interaction using contextual recognition of existing visual objects, such as signs and information plates. Object detection and localization are based on fast matching of local image features represented as Histogrammed Intensity Patches. We show that low-level image features can be learned, organized, and optimized for discrimination between similar, poorly textured object images. The system is integrated with existing software for information retrieval and experiments show that it achieves good performance in real-life situations.

Keywords: mobile computer vision, user interaction with mobile device, object detection, object tracking

Mobilni računalniški vid za inteligentni javni prevoz: primer "Interaktivne avtobusne postaje"

Članek predstavi primer uporabe računalniškega vida za lajšanje dostopanja do informacij v vsakdanjem življenju potnika javnega prometa. V središču je intuitivna interakcija brez tipkanja in dotikanja mobilne naprave z uporabo razpoznave konteksta obstoječih vizualnih objektov, kot so znaki in informacijske table. Detekcija in lokalizacija objektov temeljita na hitrem iskanju ujemanj lokalnih značilk slike, predstavljenih z intenzitetnimi histogrami. V članku pokažemo, da se lahko nizkonivojskih značilk slik naučimo in jih razvrstimo tako, da je omogočeno razločevanje med slikami podobnih, slabo teksturiranih objektov. Sistem smo integrirali z obstoječo aplikacijo za dostopanje do prometnih informacij in z eksperimenti pokazali dobro delovanje v realnem okolju.

1 INTRODUCTION

With the increased number of urban commuters and a growing environmental impact of traffic, it is of major importance to increase the comfort level and attraction of public transport. Modern communication technology already provides passengers with reliable information about their commute, and real time traffic information allows for intelligent management of transport systems with optimized scheduling and occupancy. Passenger information systems that target commuters in public transport usually gather data from vehicle location systems and combine them with planned schedule information and historical statistics to deliver predictions for arrivals and departures of vehicles in the transport network. Passengers can access this information at various locations (e.g. at home, on the way to the station, at the station or in the vehicle) and use it for planning a commute and to get an insight on various possibilities the network offers. Most of the dynamic information is accessible either actively, through websites, information kiosks, mobile applications, or passively, e.g. through LED information panels at stations and in vehicles. It is therefore important not only to serve quality information, but also to design the information displays and interaction patterns in a way that is convenient and easy for daily passengers, out-of-towners, and especially for the technologically challenged population that would be otherwise reluctant about using public transport.

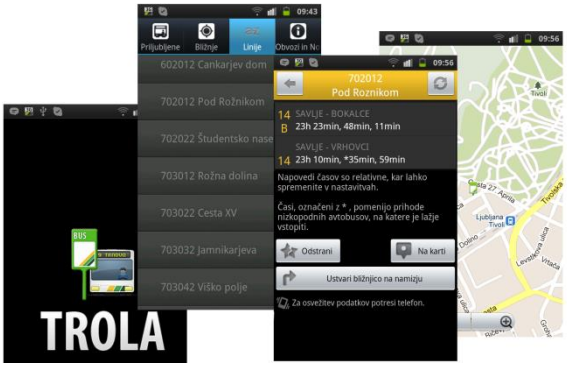


Figure 1. TROLA mobile application [1] for Android phones is serving passengers in Ljubljana, Slovenia

From the emergence of smartphones on, mobile applications have been serving an increasing quantity of

Received 8 March 2013
Accepted 31 March 2013

information, and passenger information systems have quickly adopted mobile applications to serve the public transport information. Fancier applications use sensory data such as GPS, compass, and accelerometer to facilitate user interaction and queries about nearby locations, stations, and routes. Figure 1 shows sample screens from the TROLA application for Android smartphones [1] that was developed by Studio314. The application serves commuting information to passengers in Ljubljana, Slovenia by querying the public transport provider's information system.

The existing applications such as the abovementioned one depend heavily on the interaction using keyboard and touch screen. However, the smartphone cameras offer an interesting alternative for touchless interaction using computer vision. While some public transport operators use QR code mobile tagging of stops and other assets such as busses as entry points to their passenger information systems, we envision a more seamless vision-based interaction requiring no invasive modifications of the existing infrastructure. Our approach is to use the smartphone camera as the primary mode to determine the context of the passenger's information request using a computer vision analysis of contextual features and visual objects in the immediate environment.

2 INTERACTIVE BUS STOP: CONTEXT-AWARE ON-SITE INFORMATION RETRIEVAL USING VISION

The TROLA application for Android smartphones does a great job at servicing information to passengers familiar with the supported public transport network (buses in Ljubljana, Slovenia). Using the available features such as shortcuts and favorites, one can customize the steps to get the required information. For the passengers new to the area and public transport, however, the application can be hard to use as one has to manage unfamiliar application and unfamiliar area or transport mode at the same time. The interaction can be also difficult for the elderly and other technologically challenged persons that are not used or able to navigate through selectors on a mobile screen. By using vision to recognize the context, we aim to develop an interaction design pattern that requires minimal user input.

Ideally, the phone would automatically recognize the bus stop, the location and the user's intentions, and automatically provide the contextually relevant data. To achieve this, we can use GPS, vision and inertial sensors.

Inertial sensors can detect the user's movement and initiate the appropriate action. When using one of our simplest interaction patterns the user lifts the smartphone as if taking a picture, the computer vision module is activated, starting the context recognition process. This works by comparing the inertial sensor data to the values for upright position of the smartphone and enabling the camera and detection processes appropriately, thus saving power.

While the location context can be constrained by the GPS data, it is often not accurate enough to discriminate between the nearby stops, mostly because of urban canyons interfering with reception of the GPS signals. The nearby stops usually have the most information in common, as stops across the street, for example, usually share all but route direction. For a passenger in an unfamiliar area it can be very hard to tell the exact stop. Visual recognition is therefore a vital part of the system.

Figure 2. Bus stop with information plate

Figure 2 shows a typical bus stop of the Ljubljana bus network. Using a smartphone camera directed to the stop's information plate, upon successful detection user requests information by simply touching the augmented frame of the recognized target. Information retrieval is based on the detected object with a single touch on the detected object's frame. To implement an immersed and intuitive interaction the targets should be tracked visually. Images captured by the camera are fed to both the detection module and the screen, so the user gets feedback on what the camera is looking at. When an object, in this case the information plate, is detected, the image is augmented with the learned plate frame and identity. In our application we use images of information plates located at each bus stop as target objects detectable and trackable by the mobile vision system. Figure 3 depicts seven exemplar plates used in this paper.

3 OBJECT DETECTION AND LOCALIZATION

Our first concern in searching for an appropriate object detection and localization method was speed: we should be able to detect and localize an object in near real-time on mobile platform. There are several proposed methods that work in real time on desktop systems (e.g. [2]), some (e.g. [3]) near real-time on a mobile platform. Our second concern was the ability to discriminate between similar objects, which also implies that multiple objects can be handled. While most of the available methods handle multiple objects, none seems to be able to discriminate between very similar objects. As depicted

Figure 3. Seven target images of bus stop plates. All targets share the logo (upper left part) and the route number (bottom left part). Two directions (bottom part) and four stop names (center part) are shared between several targets. Stop ID (upper right part) is the only distinct part.

in Figure 3, our targets are poor in texture and similar to each other, sharing the same parts.

Our third concern was to be able to detect low-texture objects. Authors in [2] propose a method that works on texture-less objects, is fast and works for multiple objects, but it is not clear how it can handle very similar objects. Our final concern was that object descriptions should be compact, thus enabling fast and imperceptible transfer to a mobile device.

The Histogrammed Intensity Patches (HIP) in [3] provide a fast method for feature detection and tracking. However, their ability to discriminate between similar objects is poor. Figure 4 depicts the issue: detection of the correct target (a) has 58 matches, while the detection of a very similar target (b) has 60 matches. Detections are unstable as to which of the similar targets is detected, as the process is depends on the extracted features (FAST-9 corners, see [6]). Also, no other measure can be used to discriminate between such targets on the method as-is.

a b

Figure 4. Detection of two targets on the same image using HIPs. Correct detection (a) has 58 matches, while wrong detection (b) has 60 matches.

The HIP method of [3] was therefore extended with a learning step that maximizes the number of distinct features for each target and labels them with the targets they support. In detection and localization, this information is used to efficiently discriminate between similar targets.

3.1 Learning Target Models

Target models are learned from target images. The first step in our method is registration of target images, to enable sharing similar model features between different targets in further steps. For this purpose the implementation with RANSAC and Levenberg-Marquardt optimization from [7] is used. After registering all image pairs, the ones that make no sense (subjective to operator, of course) are manually.

3.1.1 Collecting Subfeatures

After target images are registered, the learning of model features commences by cycling through targets and taking target's image until enough *subfeatures* are collected (see below).

As in [3], the view of the target is simulated by applying shear, rotation, moving, scaling, blur and noise to the target image, but only a single viewpoint bin is covered, and views are taken randomly.

Corners are then detected with FAST-9 detector in the transformed image, and patches from 8x8 sparse grids around the corners, aligned with corner orientation, are collected as *subfeatures* for the target, together with feature position and orientation in the reference frame.

Selecting repeatable features in [3] is done for each target independently, which would hinder the ability to discriminate similar targets further on. We therefore include target data in the subfeature data, and generate sets of subfeatures in the joined target space.

A subfeature is added to feature set of the closest feature in any target frames (which were registered to the subfeature's target frame), for which localization and orientation error lie below some threshold. We chose the same values as in [3] for those, i.e. 2 pixels for localization, and 10 degrees for orientation error. When a subfeature is not close enough to any features, a new feature is created.

The procedure of collecting subfeatures ends when there are has enough features for each target (at least 20 in this paper), each of which is supported by enough subfeatures (at least 10 in this paper) for that particular target. Such features are called valid features. Features that are in common to several targets are therefore supported by a large number of subfeatures.

Valid features are used to create model feature database. Quantized patches of each feature's subfeatures are combined to produce HIP description of the feature. Each HIP model contains 64 histograms (for each of 8x8 patch samples) of 5 quantized intensity levels. See [3] and [4] for details.

Differently to [3] our features are (possibly) supporting multiple targets, so our database includes feature positions in the frame of each supported target.

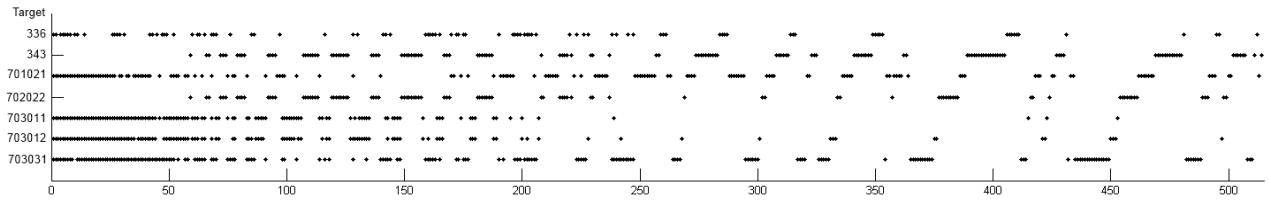


Figure 5. Learned features target support. A dot is drawn where feature (horizontal axis) supports target (vertical axis). Some features support multiple targets, while others are distinct to a specific target.

3.2 Target Detection and Localization

The images from the camera video stream are treated in the first few steps in most the same manner as in [3]: FAST-9 detector extracts highest scoring corners, 8x8 samples are extracted for each feature from patches rotated to the feature orientation, and normalized for mean and standard deviation. Using the same 5 histogram bins as in learning, each sample pixel produces a 1 in the bin it falls in and 0 in other 4 bins, thus enabling a fast comparison to learned models with bitwise operations. Match error scores are computed for all the matches between the learned features in database and image features, and the matches with the error score $e \leq 2$ are labeled *primary*, while those with $2 < e \leq 4$ *secondary*. See [3] and [4] for details on all the above steps.

From all the matches, sets of consistent matches for each target were built. A primary match is consistent with another match if the distance d_m between its two image features is in accordance to the expected distance d_e (from model features) $0.4d_e < d_m < 1.5d_e$, and the angle between the vector connecting the two matches and the primary match orientation is in accordance to the expected angle to 30 degrees. Same values were used as in [3]. Sets are built for all primary matches and all targets, and then sorted by the number of matches they contain. Sets with less than 10 matches are discarded in this paper.

Starting with largest match set, homography for the set with PROSAC is computed next. After a valid homography is obtained, the support of the homography's inliers I for the target is assessed to get the detected targets. The *target support* of the set S is defined as the percentage of inlier matches that support target T_i

$$s(T_i) = \frac{|I(T_i)|}{|I|}$$

where I is the set of inliers of the computed homography, and $I(T_i)$ is a subset of inliers I whose matches support target T_i. Set S supports targets for which $s(T_i) > 0.8$ in this paper. The value was set by detecting all targets on all training images. For detections that support multiple targets the system should act as it is not able to distinguish which is the correct interpretation of the input image and consider all supported targets as possible interpretations.

Any indexing and tree based search [3] was omitted from the implementation of the method, as our goal was to tackle the similar targets problem. However, the two can be included in a more or less straight forward manner if processing speed is an issue.

4 EVALUATION

We implemented learning part of the method described in the previous section in MATLAB. Figure 3 depicts seven target images that were taken from the Ljubljana public-transport bus-stop information plates. The respective bus stops were in the vicinity of each other and they shared several parts: the logo (upper left part), bus stop name (middle part), bus numbers and direction, etc. One target is blurry, due to bad and degraded print on the bus stop information plate. Grayscale images of 480x800 pixels were captured using Samsung Galaxy S smartphone camera and manually cropped to the region of interest. Training images of targets *343* and *702022* were registered to each other, as were all remaining five.

In the learning phase around 7000 distinct features were detected in all transformed target images. From those, 514 features (that were supported by at least 10 subfeatures) were selected and added to the database. Figure 5 shows features' target support with a dot where feature (horizontal) supports target (vertical). Note two distinct groups, one with targets *343* and *702022*, and the other with the remaining five targets. Most features are shared within each group, while some are distinct to particular targets. The two groups do not share any features as the training images were not registered to each other. Database covering the seven targets with 514 features was 66kB in size, which can be compressed to 24kB with standard zip.

4.1 Results

We tested the method on seven short sequences, each showing a single target, with 167 images altogether. Figure 6 provides detection rates for different targets. The targets were detected correctly in 69% of images. Figure 7 shows examples of correctly detected targets, with overlaid training image in white frame. There is enough size and tilt variation tolerance to cover the

Figure 7. Examples of correct detections for different targets. The detected target frames are white framed and semi-transparently augmented over the input image.

application needing no additional viewpoint bins for both the size and tilt, nor image pyramids for the size.

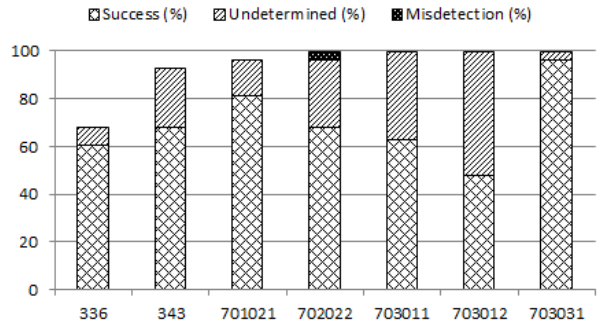
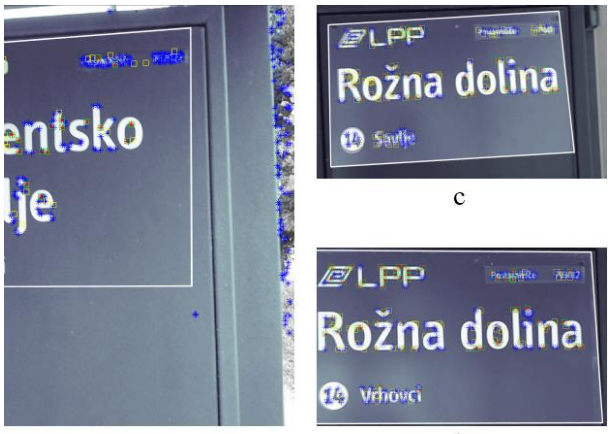


Figure 6. Detection rates for different sequences (labeled with target ids).

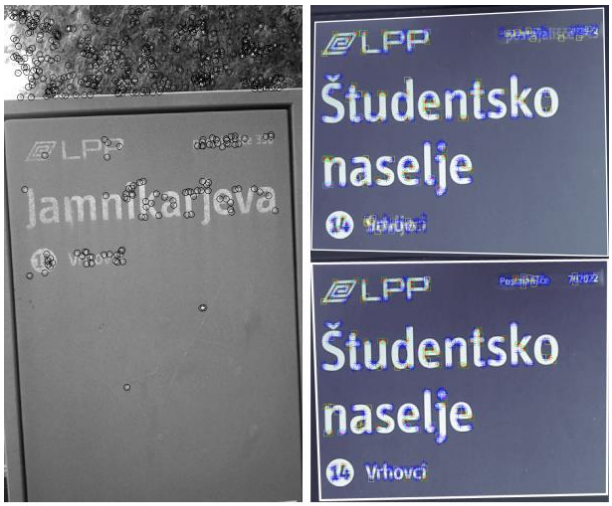
In 24% of the images detection did not determine fully the target; two or more targets were possibly supported by the computed homography. Figure 8 shows three reasons for such detections: a) only partially visible targets with distinct parts were outside the view or covered, b) homography for the best target was inaccurate and included mainly matches from non-distinct parts, and c) target support was above the threshold for two or more targets.



a b c

Figure 8. Examples of detections supporting multiple targets: a) partially visible target, b) inaccurately computed homography, c) too few distinct inlier matches in homography.

In 6% of images, no targets were detected. Target *336* had the most such cases (33%), mostly because very few corners were detected on the plate itself, like in Figure 9a. Print on the plate was degraded and the corner detection was poor on the printed area, while many corners were detected in the background on bushes in the upper part of the image. As largest match sets found had less than 10 matches, the detection process had stopped before any homographies were computed.



a b

Figure 9. a) Example of the input image with no targets detected. The extracted features are circled in black. b) The case where the incorrect target was detected (top), with better homography than the correct one (bottom).

In one image an incorrect target was detected. Figure 9b shows the case. Target 343 was detected (top), with 66 supporting inliers, while the correct target *702022* (bottom) had only 61 supporting inliers. Similarly as in undetermined detections, homography computed for the incorrect target *343* was slightly better than the one for *702022*. The target support of the former was just below the threshold for the correct target.

5 MOBILE APPLICATION

We designed the interaction with bus stop to be as simple as possible in mobile application. User starts the application, directs smartphone to get the information plate on screen, confirms the request for information by touching the augmented frame of the detected plate, and is presented with available routes and their respective predicted arrival times.

The core of the application is object detection and localization method described in Section 3. It is implemented in Android NDK to ensure optimal execution speed. 480x800 pixel images are captured using the smartphone camera and processed by the method to give identity and location in the image frame of the detected object.

When a successful detection is made, a green frame is augmented at the computed location with detected bus stop id in its center, inviting the user to touch it. Upon interaction, the bus stop id is fed to a raised TROLA application, which retrieves and presents the relevant route and time information.

If detection is such that the object cannot be determined well (i.e. there are two or more objects possible), the frame is augmented in red with an hourglass icon in its center. The user waits for the next frame(s) to get successful detection.

The method performs at around 3 frames per second on Samsung Galaxy S with Android 2.3, which is enough to enable smooth interaction. To improve the performance of our method, we reduce the set of the known objects by using GPS location. When initializing, the database set including the bus stops nearest to current location is chosen.

6 CONCLUSION

We argue that the interaction using mobile computer vision can make public transport passengers more comfortable in their commutes. In this paper we described a prototype application that uses vision to complement other sensors in a seamless interaction pattern that can be used to access contextually relevant data with minimal user intervention. Interaction is based on visual context and object recognition. We have extended an existing object detection and localization method with a learning step that learns a feature based representation capable of discriminating between similar targets. Detection was successful for a set of seven targets captured from varying viewpoints. Our tests show that the proposed interaction is a feasible alternative to screen based interaction.

REFERENCES

[1] TROLA mobile app, http://studio314.si/trola.

[2] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. “Dominant orientation templates for real-time detection of texture-less objects”, In Proc. IEEE CVPR, 2010.

[3] S. Taylor and T. Drummond, “Multiple Target Localisation at over 100 FPS”, In Proc. of BMVC, September 2009.

[4] S. Taylor, E. Rosten and T. Drummond, “Robust Feature Matching in 2.3µs”, In Proc. of IEEE CVPR Workshop on Feature Detectors and Descriptors, June 2009.

[5] T. Lee and S. Soatto, “Learning and Matching Multiscale Template Descriptors for Real-Time Detection, Localization and Tracking”, In Proc. IEEE CVPR, June 2011.

[6] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection”, In Proc. ECCV, May 2006.

[7] P. D. Kovesi. MATLAB and Octave Functions for Computer Vision and Image Processing, http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/.

Jaka Krivic received his B.Sc. degree in 1997, M.Sc. degree in 2000 and Ph.D. in 2006, all at the Faculty of Computer and Information Science, University of Ljubljana. He is currently working as a researcher at the Faculty of Computer and Information Science, University of Ljubljana. His research interests include computer vision based interaction and mobile processing.

Matjaž Jogan received his B.Sc. degree in 1997, M.Sc. degree in 2002 and Ph.D. in 2008, all at the Faculty of Computer and Information Science, University of Ljubljana. He joined Computational Perception and Cognition Laboratory at University of Pennsylvania. His research interests include visual categorization, spatial cognition and computational perception.

Aleš Leonardis is a full professor and the head of the Visual Cognitive Systems Laboratory with the Faculty of Computer and Information Science, University of Ljubljana. His research interests include robust and adaptive methods for computer vision, object and scene recognition and categorization, statistical visual learning, 3D object modeling, and biologically motivated vision.