

What is That?

Object Recognition from Natural Features on a Mobile Phone

Niels Henze
OFFIS - Institute for
Information Technology
Escherweg 2
Oldenburg, Germany
henze@offis.de

Torben Schinke
University of Oldenburg
Escherweg 2
Oldenburg, Germany
torben.schinke@uni-
oldenburg.de

Susanne Boll
University of Oldenburg
Escherweg 2
Oldenburg, Germany
susanne.boll@uni-
oldenburg.de

ABSTRACT

Connecting the physical and the digital world is an upcoming trend that enables numberless use-cases. The mobile phone as the most pervasive digital device is often used to establish such a connection. The phone enables users to retrieve, use, and share digital information and services connected to physical objects. Recognizing physical objects is thus a fundamental precondition for such applications. Object recognition is usually enabled using visual marker (e.g. Qr Codes) or electronic marker (e.g. RFID). Marker based approaches are not feasible for a large range of objects such as sights, photos, and persons. Markerless approaches that use the image stream from the mobile phone's camera are commonly server-based which dramatically limits the interactivity. Recent work on image processing shows that interactive object recognition on mobile phones is at hand. In this paper we present a markerless object recognition that processes multiple camera images per second on recent mobile phones. The algorithm combines a stripped down SIFT with a scalable vocabulary tree and a simple feature matching. Based on this algorithm we implemented a simple application which recognizes poster segments and conducted an initial user study to get an understanding of the implications that accompany markerless interaction.

Keywords

natural features, mobile phone, object recognition, markerless, camera

1. INTRODUCTION

Mobile devices with the mobile phone on its forefront are a ubiquitous part of our daily life. Not only because of their limited in and output capabilities there is an increasing interest in extending the interaction between the user and her phone to an interaction between the user, the phone and real world objects. Typical application are mobile tour guides which enables the user to point at sights [17, 1] to get further information, access related services for advertisements [10], or go shopping in physical stores [16]. To implement such an interaction with a real world object it is necessary that the mobile phone senses objects in its surrounding in some way.

Approaches to sense real world objects are usually based on visual markers (e.g. QR-Codes or other 2D barcodes) or dig-



Figure 1: User interacting with an interactive poster.

ital markers (e.g. RFID tags). For certain types of objects, such as sights, buildings, and living beings marker based approaches are often not sensible or considerably restrict the interaction radius. Markerless approaches, for instance based on natural features, can overcome some of these limitations. However, they suffer from high demands on the available processing power. Thus, markerless object recognition is usually performed on a remote server (see e.g. [10]) or preformed on the mobile device with a delay of up to several seconds. Either way this delay clearly restricts the interaction.

The work by Wagner et al. [14] showed that estimating the 3D pose of a 2D object from natural features with a high frame rate is feasible on recent mobile phones. In this paper we build up on this approach and describe an algorithm, which combines it with a vocabulary tree to recognize a number of objects. Based on this algorithm we implemented a prototype (see Figure 1) to conduct an evaluation which offers first insight into the implications of real-time markerless object recognition that provides direct feedback to the user.

In Section 2, we present work related to object recognition, in particular, on a mobile phone. The developed algorithm is described in Section 3. We present a first user test in Section 4 and close this paper with a conclusion and outlook to future work in Section 5.

2. RELATED WORK

Fitzmaurice was one of the first who predicted the use of mobile devices for pointing based interactions [3]. He described for instance an application with which the user could point onto certain locations on a map in order to get additional information [3]. In recent years systems developed for mobile phones emerged which provide information related to physical objects. A common and commercially successful way to implement such systems is to mark objects with visual markers [11] or electronic markers [15]. However, not all types of objects are suitable for equipment with markers. Sights and buildings are simply too large or out of range to be reasonably equipped with either type of markers. It is questionable if objects whose visual appeal is important, can in general be sensibly equipped with visual markers (see e.g. [4]). In addition, markers restrict the interaction radius in a specific way.

Another approach uses images from a camera for the recognition of digital images to find the corresponding digital information using content based image analysis. Lowe presented the influential Scale Invariant Feature Transform approach (SIFT) [7] which allows the recognition of arbitrary physical objects. SIFT is invariant against rotation by design and robust against scale, light changes, partial occlusion, and perspective changes. A survey of local feature descriptors in general can be found in [8].

In recent years the size of the database whose content can successfully recognized increased dramatically. While Lowe reported to successfully recognize around 50 objects Nister et al. presented the recognition using a vocabulary tree [9] which enables to find an image out of two million images in two seconds on a high-end computer. Schindler et al. refined the vocabulary tree [13] and improved its performance by a factor of more than ten.

Recognition of a large number of objects is feasible if enough processing power and memory is available. However, even computing the widely used SIFT descriptor alone, without any matching or storage issues, overcharges mobile phones today, which is the reason why remote server processing is so popular. Liu et al., for example, use the camera of a mobile phone to select documents displayed on computer screens [5]. Erol and Hull use mobile phones' cameras to enable the user to select presentation slides by taking images of the slides [2]. Zhou et al. developed a system to acquire information related to sights by taking a photo [18]. However, recently Wagner et al. [14] adopted the SIFT and FERNS algorithms for mobile devices and estimate the pose of a single image with high frame rates.

3. OBJECT RECOGNITION ALGORITHM

Widely used object recognition approaches such as SIFT are too expensive in terms of processing power. SIFT consist of three steps: keypoint detection, feature description, and feature matching. Wagner et al. describe a simplified SIFT algorithm to estimate the 3D pose of a 2D object [14]. Their approach is capable to process camera frames with a size of 320x240 pixels at a rate up to 20Hz. However, only results from processing a single image are reported and it was not analyzed how the algorithm performs with an increasing number of objects. In the following we describe the

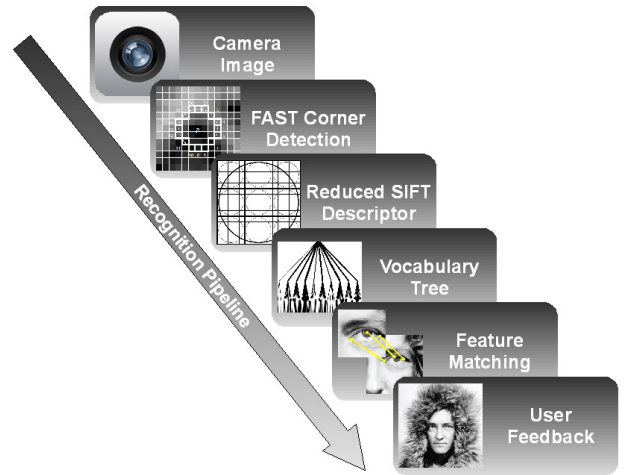


Figure 2: Overview of the recognition pipeline.

extension of the approach developed by Wagner et al. using a scalable vocabulary tree [9]. Our recognition pipeline is outlined in Figure 2.

3.1 Keypoint detection and feature description

For the keypoint detection we adopt the simplified SIFT algorithm [14]. In the pre-processing phase images are successively downscaled with a factor of $\sqrt{2}$ to achieve scale invariance. For each scale level the image is smoothed with a 5x5 Gaussian kernel. Afterwards a FAST corner detector [12] detects keypoint candidates. During the pre-processing we detect up to 300 features per scale step. During the on-line phase the image is not downscaled detecting keypoints from a single image plane.

The keypoint candidates are feed into the creation of feature descriptors. Image patches around the origin of the candidate with a size of 15x15 pixels are used to derive the descriptor. First the patch's main orientation is derived from the pixels' of the patch. The pixel's gradients are weighted by a Gaussian function and quantized to a natural number between 0 and 36. The result is inserted into an orientation histogram. For each peak in this histogram one feature is created.

The feature's according descriptor is again derived from the 15x15 pixel image patch. The patch is subdivided in 3x3 sub-regions. For each pixel the orientation is weighted by their distance to the patch centre as well as to the sub-region centre and quantized to a natural number between 0 and 4. The 3x3 sub regions and the 4 quantization steps form a descriptor with 3x3x4=36 entries. The descriptor is further normalized and each of its values is cropped to 80% of the descriptor's overall length.

3.2 Feature matching

Wagner et al. employs a "Spill Forest" (a combination of a number of Spill Trees [6]) to match features extracted from the camera image with features from all scale steps of the reference image. Since our aim is to recognize a number of images we employ a different approach. Vocabulary trees [9]

are able to reduce the problem to find the matching object by multiple magnitudes. Nister and Stewenius trained a vocabulary tree which reduced the problem to find an image out of two million to a problem to find an image out of a hundred candidates. Unfortunately the vocabulary tree described by Nister and Stewenius has a size of hundreds of megabytes and must be stored in RAM for performance reasons. We downsize the tree by reducing its level to five instead of six and a branching factor of eight instead of ten. In addition, our descriptor has only 36 entries instead of 128. Through this our empty tree needs only two megabyte. We trained our vocabulary tree with 10000 images, mainly high quality photos.

Reference images are inserted into the vocabulary tree by extracting the features from each of the image’s scale steps. Each scale step is then treated individually and inserted into the tree. By treating the scale steps individually we obtain not only object candidates from the vocabulary tree but scale step candidates. During the online-phase three scale-step candidates are retrieved from the vocabulary tree.

Since we do not aim at fine grained pose estimation no sophisticated feature matching is necessary. Thus, we rely on simple brute-force matching to compare the 100 features from the camera image with the 300 features from each of the three scale-step candidates using the sum of squared difference. To further reject potential outliers we compute a difference of orientations histogram for each candidate’s matches. If this histogram shows a consistent rotation and the respective candidate’s number of matches is above a certain threshold in two consecutive camera images the according image is considered as a match.

3.3 Performance

The algorithm described above was implemented for Windows Mobile 6 devices using *C*. We tested the speed using an ASUS P535 Smartphone equipped with an Intel XScale PXA270 processor running at 520 MHz and 64MB built-in RAM (26MB RAM available for applications). The respective durations are averaged over a short test sequence with a resolution of 320x240 pixels. Initial smoothing of the camera image takes 10ms. Afterwards keypoints candidates are detected using 13ms and the descriptors are computed in 27ms. Finally the features are matched against the vocabulary tree containing 343 scale levels corresponding to 57 images in 6ms. The three best candidates are matches with brute-force in 44ms. The overall time to process an image from the camera accordingly takes 100ms.

4. USER STUDY

In order to get a first impression of the implications that accompany markerless interaction with physical objects we conducted an early user study. Our aim was to observe how the participants interact if no visual marker highlight interactive areas.

4.1 Developed prototype

In order to conduct the user study we developed a simple prototype shown in Figure 3. The prototype displays the camera image in full screen. The camera image is constantly delivered into the recognition algorithm described in Section



Figure 3: ASUS smartphone running the used prototype.

3. If an image is recognized a small thumbnail of the recognized image overlays the camera image. The user can get details about the object by clicking the thumbnail with her finger.

4.2 Method

Six male colleagues from the lab participated in the study. All were between 25 and 35 years old. The evaluation consisted of three tasks described in the following. The sequence of the second and the third task was randomized. We asked the participants to fill a NASA TLX questionnaire after the second and third task.

In the first task the 45x55 cm large poster shown in Figure 4 was used. The poster sketches a street setting and contains seven interactive regions. If a participant selects one of these regions the phone displays advises about how to behave in the respective traffic situation. The poster lay flat on a table. The participants were asked to find all interactive areas without knowing its number. It was up to the respective participant to decide when to end this task.

In the second and third task two very similar posters hanging on a wall were used. Each poster contained 24 clearly identifiable interactive regions. Figure 4 shows a cut-out of the poster. For some of these regions the thumbnail that was displayed, when a region was recognized, contained a question mark. If the participant clicked the thumbnail the phone displayed either a happy green or a sad red emoticon but each poster contained only one happy emoticon. So the participants’ task was to find the happy one. One aim of these two tasks was to get an understanding of the participant’s behaviour if the recognition fails. Therefore, three regions were deactivated in the third task.

4.3 Results and Discussion

In the first task the participants found between three and six interactive regions ($\phi = 4.66$ $\sigma = 1.21$). No participant was able to find the region located in the upper right of the poster. All but one participant started by systematically scanning the poster in zigzag. After scanning the whole poster once some started to scan specific regions of the poster. All but one participant permanently aligned the phone with the orientation of the poster. Three participants held the phone in an almost constant height. Two participants mentioned that additional hints to surrounding interactive regions would be helpful and one participant said that it is difficult to remember the parts of the poster that were already scanned.



Figure 4: The poster used in the first task of the evaluation (left) and an extract of the poster used in the second task.

All participants managed to complete the second and third task. However, probably due to the small number of participants the NASA TLX showed no significant difference between the two tasks. Some participants rushed through these tasks and two did not even noticed the three deactivated regions. The longest time a participant tried to select one of these regions was around 20s. All but two participants permanently aligned the phone with the orientation of the poster. One participant rotated the phone by 90° and one participant did not show a consistent behaviour. All but one participant focused most of the time on one region after the other so that the respective region approximately filled the phone's screen.

Because of the used methodology and the selected participant the study can obviously not be generalised. However, the results indicate that users intentionally align the phone with the object. This is consistent with the observation we made in earlier work. It could imply that the recognition pipeline can be simplified by removing orientation invariance in tasks such as ours. Unsurprisingly the participants had problems to find all interactive regions if these are not clearly distinguishable. When marking interactive objects is not feasible additional hints displayed by the phone could ease finding nearby objects.

5. CONCLUSIONS AND FUTURE WORK

In this paper we described an algorithm which enables to recognize hundreds of objects on a mobile phone. We employ a FAST corner detection, stripped down SIFT descriptors, and a vocabulary tree combined with brute-force matching. The implementation is able to process about ten images per second on an Asus P535 Smartphone. The algorithm is used to implement a basic prototype to evaluate the implications of markerless image recognition on a mobile phone.

The implemented algorithm is far from being optimized and all stages can probably be improved in terms of speed and accuracy. In particular, the brute-force matching can be replaced by more sophisticated techniques. Furthermore, more detailed user studies are necessary to get a deeper understanding of the implications that accompany markerless object recognition. This is especially true if going beyond 2D objects by enabling interaction with 3D objects.

6. ACKNOWLEDGMENTS

This paper is supported by the European Community within the InterMedia project (project No. 038419).

7. REFERENCES

- [1] N. Davies, K. Cheverst, A. Dix, and A. Hesse. Understanding the role of image recognition in mobile tour guides. *Proc. of MobileHCI*, 2005.
- [2] B. Erol and J. J. Hull. Linking presentation documents using image analysis. *Proc. of Signals, Systems, and Computers*, 2003.
- [3] G. W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36(7), 1993.
- [4] N. Henze and S. Boll. Snap and share your photobooks. *Proc. of ACM Multimedia*, 2008.
- [5] Q. Liu, P. McEvoy, and C.-J. Lai. Mobile camera supported document redirection. *Proc. of ACM Multimedia*, 2006.
- [6] T. Liu, A. Moore, A. Gray, and K. Yang. An investigation of practical approximate nearest neighbor algorithms. *Advances in neural information processing systems*, 2004.
- [7] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.
- [8] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.
- [9] D. Nister and H. Stewenius. Scalable Recognition with a Vocabulary Tree. *Proc. of Computer Vision and Pattern Recognition*, 2006.
- [10] M. Pielot, N. Henze, C. Nickel, C. Menke, S. Samadi, and S. Boll. Evaluation of Camera Phone Based Interaction to Access Information Related to Posters. *Proc. of Mobile Interaction with the Real World*, 2008.
- [11] M. Rohs and B. Gfeller. Using camera-equipped mobile phones for interacting with real-world objects. *Proc. of Pervasive Computing*, 2004.
- [12] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. *Proc. of European Conference on Computer Vision*, 1, 2006.
- [13] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. *Proc. of Computer Vision and Pattern Recognition*, 2007.
- [14] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. *Proc. of ISMAR*, 2008.
- [15] R. Want, K. P. Fishkin, A. Gujar, and B. L. Harrison. Bridging physical and virtual worlds with electronic tags. *Proc. of CHI*, 1999.
- [16] Y. Xu, M. Spasojevic, J. Gao, and M. Jacob. Designing a vision-based mobile interface for in-store shopping. *Proc. of NordiCHI*, 2008.
- [17] T. Zeidler, B. Brombach, E. Bruns, O. Bimber, and P. Föckler. Phoneguide: museum guidance supported by on-device object recognition on mobile phones. *Proc. of Mobile and ubiquitous multimedia*, 2005.
- [18] Y. Zhou, X. Fan, X. Xie, Y. Gong, and W.-Y. Ma. Inquiring of the Sights from the Web via Camera Mobiles. *Proc. of Multimedia and Expo*, 2006.