Practical Issues in Physical Sign Recognition with Mobile Devices

Christof Roduner[1] and Michael Rohs[2]

[1] Institute for Pervasive Computing, Department of Computer Science,
ETH Zurich, 8092 Zurich, Switzerland
roduner@inf.ethz.ch

[2] Deutsche Telekom Laboratories and TU Berlin,
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
michael.rohs@telekom.de

Abstract This paper explores the use of physical signs as anchors for digital annotations and other information. In our prototype system, the user carries a camera-equipped handheld device with pen-based input to capture street signs, restaurant signs, and shop signs. Image matching is supported by interactively established point correspondences. The captured image along with context information is transferred to a back-end server, which performs image matching and returns the results to the user. We present a comparison of four different algorithms for the sign matching task. We found that the SIFT algorithm performs best. Moreover, we discovered that lighting conditions – especially glare – have a crucial impact on the recognition rate.

1 Introduction

Building on the idea and extending the work presented in [10], we evaluate how physical signs – and other areas with four clearly distinguishable corners – can be used as anchors and entry points to digital information. Recognition based on unmodified visual appearance is especially desirable in situations in which it is not practicable to attach a visual marker or RFID tag to an object. In urban areas, many objects, like street signs, shop signs, restaurant signs, indication panels, and even facades of buildings have clear borders against the background. Users can perceive and handle them as separable entities to which information is attachable.

There is a large range of application possibilities for linking information to signs in outdoor environments. Users may access food ratings via restaurant signs, access city maps via street signs, or attach information to company logos. Especially user-generated forms of content, such as digital annotations, are promising. In order to be used, such systems have to enable structured feedback that can be entered with minimal effort in mobile situations. The feedback can be structured according to an ontology or taxonomy that depends on the type of physical object, thus suggesting specific feedback or rating scales. Mobile annotation systems also have to enable collaborative experience sharing between users, in order to be useful. This is achievable through data exchange via the mobile phone network.

Many projects have investigated linking online information and services to physical media [1,3,4,8,9,11,12,13]. Whereas these projects have relied on some labelling technology, like visual markers or RFID tags, we try to achieve physical hyperlinking without any modification of the object itself and just rely on its visual appearance. To this end, we built a system that consists of a camera-equipped mobile device for capturing physical signs and a back-end system for matching captured signs against a database of template images.

Our prototype device – an MDA III running the Windows Smartphone operating system – has a touch-screen. This allows users to interactively highlight areas of interest, which greatly simplifies the image matching task. It enables reliable matching, even if the images in the template database are visually very similar to each other, which is often the case with street signs. Moreover, pen-based input enables rich interaction with captured images, like drawing arrows to give directions or putting predefined icons onto the captured image. Wireless connectivity is required for sending captured images to the back-end server, for getting up-to-date information, and for sharing content with others.

In the next section we review the general idea that was presented before in [10]. In Section 3 we outline the design and implementation of our prototype system. In Section 4 we discuss the image matching algorithms that we used in our experiments. In Section 5 we present the recognition performance measured in our experiments along with a discussion of practical problems encountered. Finally, in Section 6, we draw a conclusion and give directions for future work.

2 Physical Sign Recognition with Mobile Devices

Many sign-like objects in urban space have sufficiently characteristic visual features to be recognizable by machines. They are also recognizable by humans as separate entities to which information is linkable, since sign-like objects have clear-cut borders relative to the background. In order to attach or retrieve information using their camera phone, users take a photo of the sign including any background. They then tap the four corners of the sign on the device screen with the stylus. This corner marking method solves two problems. First, if multiple candidate objects are present in the image, the one of interest to the user is selected. Second, the image segmentation process becomes trivial.

To enable simple matching of the marked part of the image (the actual sign) against a set of templates, the marked part is projected into a square of fixed size ("warped"). Depending on the orientation of the user towards the sign when taking the photo, the sign may appear perspectively distorted. The distortion can be removed and the marked part projected into the square by treating the four corners of the sign as correspondences to the corners of the square. Since the corners are coplanar, there exists a unique homography (projective transformation) between the pixel coordinates of the marked area and the square. We can thus produce a square request image of fixed size, which is sent to a server for matching against a set of template images.

3 Prototype System

Our application consists of two main parts: A client application running on the MDA III mobile device and a server part running on a webserver. After the user has taken a picture using the client application, he or she can use the mobile device's pen to mark the sign that is to be annotated by pointing to its four corners. It is important that the corners are marked in the correct order required by the system (clockwise, starting with the upper-left corner). This ensures correct orientation of the image, which is essential when comparing it to other images already stored in the system. The image is compressed using JPEG and, together with the coordinates of the four corner points, transmitted to the servlet via a HTTP POST request. It is then warped by the servlet to a square of 128x128 pixels. We found this size to offer a good trade-off between recognition rate and processing time. Using each of the image matching algorithms (see Section 4), a list of the five best matches against all images stored in the MySQL database is composed. After that, a web page containing the picture taken by the user and a table containing the candidate images from the database is generated and displayed in the MDA III's web browser (Figure 1). The user then manually selects the sign corresponding to his or her original snapshot. If it cannot be found among the suggestions, he or she can request the new image to be added to the database.

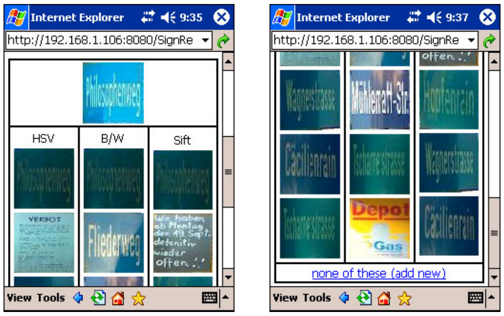


Figure 1. Sign photographed by user and the first two matches for each algorithm (left). The rest of the five best matches with the option to add the submitted image as a new sign (right).

After the user has selected one of the existing images or added the snapshot as a new sign, another web page is opened in the browser. It contains the annotations for the sign, provided there exist any, and a link to add a new annotation. Currently, our prototype only supports text annotations.

4 Image Matching Algorithms

Our application uses four different algorithms for image matching: HSV, Black / White, Wavelet, and SIFT. Each of these algorithms compares two images and offers a measure of their similarity. The *HSV algorithm* is based on comparing pixel-by-pixel hue values between two images. It calculates the absolute difference between hue values that are summed up to express the two images's similarity. If a hue value is undefined for a pixel, the gray value difference is used. The *BW algorithm* converts the original RGB image to grayscale in a first step and to monochrome in a second step using Otsu's method [7]. As a measure for similarity, the BW algorithm calculates the correlation coefficient between the two black and white images. The *Wavelet algorithm* creates a signature for each image that is used for matching. We used Francl's Eikon engine that is based on [2]. Finally, *SIFT* extracts local features from images that can be used to perform matching between different views of an object [5]. Similarity between two images is determined based on the Euclidian distance of their feature vectors. As SIFT does not match images pixel-wise (see Figure 4i), the marking and warping steps outlined above could be omitted. However, we did not test this in our prototype.

5 Results and Discussion

In order to empirically test the suitability of the four algorithms for our application, we prepared a number of sample images. These images, 95 in total, show different objects, such as street signs, building facades, company logos, posters, etc., and were stored in the database. For every object represented in the database, a second picture was taken that varies in perspective, lighting, or distance. This second image was then matched against the 95 images in the database. All pictures were taken with the MDA III's built-in camera, which has a maximum resolution of 640x480 pixel and offers a relatively poor image quality. As it does not offer optical zoom, distant objects result in rather small images. Of the 95 objects used for the purpose of our test, 11 were located indoors, while 84 were located outdoors. We consider an image successfully recognized if the corresponding object is among the top five hits returned by the image matching algorithm. The results of our tests are summarized in Table 1. The data indicate that SIFT's recognition rate is superior to the matching performance of other algorithms. The very simple BW and HSV algorithms still perform relatively well, whereas the Wavelet approach seems to be less suitable for our application. If an image is recognized correctly, it is usually ranked first or second in the candidate list. The last column of Table 1 also shows the recognition rate that can be achieved if we consider both the B/W and SIFT algorithms simultaneously, i.e. if we count an object as successfully recognized if it is among the top five hits returned by either algorithm.

Street signs are a special case in the context of sign annotation. On the one hand, they are omnipresent and thus allow for interesting new applications. On

Algorithm	**HSV**	**Wavelet**	**BW**	**SIFT**	**B/W and SIFT**
# Recognized (of 95)	81	69	86	89	94
% Recognized	85.26%	72.63%	90.53%	93.68%	98.95%
Mean Rank	1.543	1.623	1.372	1.371	1.096
% Rank 1	74.07%	76.81%	86.05%	77.53%	92.55%

Table 1. Results of image matching tests with all types of objects.

the other hand, we expected them to be difficult to handle for the matching algorithms as they all look very similar. On top of that, the material is often very susceptible to reflections and glare. We therefore did two more test runs: In the first one we did not consider any street signs, while the second one consisted only of pictures of street signs. The results are shown in Table 2 and Table 3, respectively. Again, the SIFT and BW algorithms perform relatively well with only one street sign being unrecognizable.

Algorithm	**HSV**	**Wavelet**	**BW**	**SIFT**	**B/W and SIFT**
# Recognized (of 83)	72	65	75	78	82
% Recognized	86.75%	78.31%	90.36%	93.98%	98.80%
% Gained / Lost	+1.48%	+5.68%	−0.16%	+0.29%	+1.05%

Table 2. Results of image matching tests with street signs excluded.

Algorithm	**HSV**	**Wavelet**	**BW**	**SIFT**	**B/W and SIFT**
# Recognized (of 12)	9	4	11	11	11
% Recognized	75.00%	33.33%	91.67%	91.67%	91.67%
% Gained / Lost	−10.26%	−39.30%	+1.14%	−2.02%	−7.28%

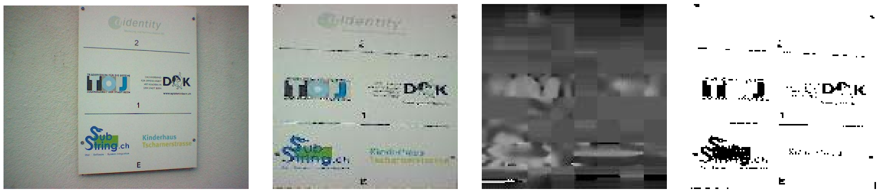
Table 3. Results of image matching tests with street signs only.

Although the overall recognition rate is surprisingly good even with very simple algorithms, there are a number of issues that are difficult to handle for the system:

Subtle features Signs that exhibit subtle features, such as the one shown in Figure 2, lead to very few distinguishable characteristics after warping. Even small translations may render image matching nearly impossible.

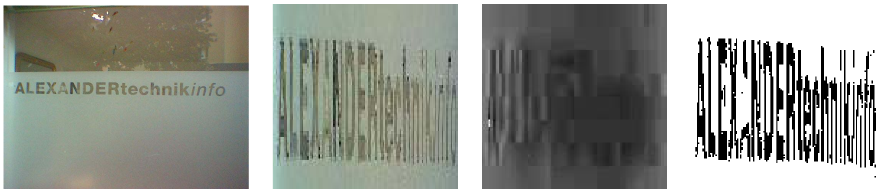
Large width and small height If a sign is wide but not very high (Figure 3), warping changes the aspect ratio considerably. In many cases, such as with text, the compressed image has few clear features.

Perspective Pictures of the same free-standing sign (e.g., of an arrow-shaped sign) can look quite different if taken from a different perspective. While pixel-by-pixel based matching algorithms cannot deal with this situation,



2a: Original 2b: Warped 2c: Hue 2d: Black / White

Figure 2. Object with few and subtle features.



3a: Original 3b: Warped 3c: Hue 3d: Black / White

Figure 3. Sign with large width and low height.

SIFT is still able to recognize the corresponding features. Even in cases where a sign looks the same from different perspectives, warping may introduce varying artifacts if a picture is taken from different angles. Moreover, reflections that would be tolerable per se can change heavily if perspective is changed.

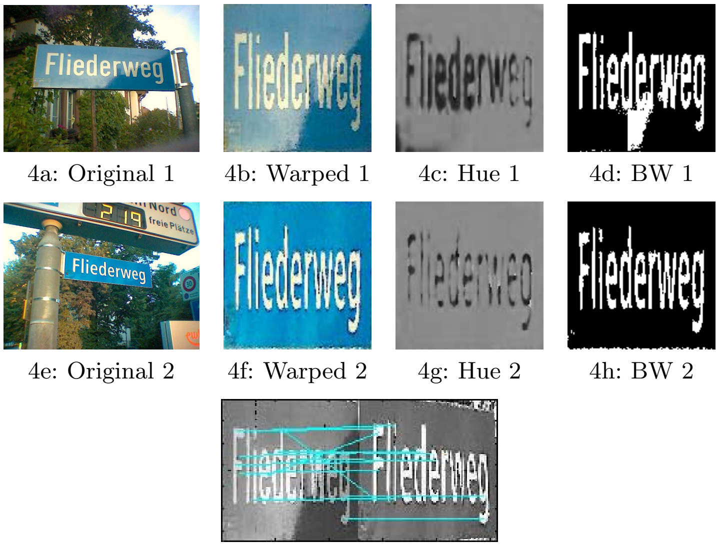
Reflections Reflecting surfaces can introduce spurious features into an image that can disturb the matching process significantly.

Blurred images Blurred images are mainly problematic for the SIFT algorithm, while the pixel-by-pixel based algorithms are somewhat more resistant to this effect. However, blurred images are a minor issue for the usability of our application since users can easily take another picture that is in focus.

Small objects In the case of small objects, magnifying may introduce artifacts and marking the area accurately can become difficult for users.

Non-rectangular objects Most signs and objects are easy to mark because they are rectangular and have clear-cut corners. However, if objects are of a different shape (e.g., round), our approach becomes impracticable for the algorithms used with the exception of SIFT. As SIFT is feature-based, it does not require the user to accurately mark the sign.

Lighting conditions If the same object is photographed under different lighting conditions, this may lead not just to reflections but also to the object appearing in different colors. Figure 4 illustrates some of the problems that can arise for the matching algorithm. For example, the color difference leads to a thicker writing in the darker image (Figure 4c), which impairs the HSV algorithm that can handle street signs well otherwise. The BW algorithm doesn't suffer from this problem. However, the reflection causes a white area



4a: Original 1 4b: Warped 1 4c: Hue 1 4d: BW 1

4e: Original 2 4f: Warped 2 4g: Hue 2 4h: BW 2

4i: SIFT 1 & 2

Figure 4. Influence of lighting conditions.

to appear, which, in this particular case, is not a serious problem due to to its small extent (Figure 4d). The SIFT algorithm, on the contrary, is not affected easily by changing lighting conditions. In the example shown in Figure 4i, 15 corresponding keypoints were identified by SIFT, while the next best match had only 4 corresponding keypoints.

6 Conclusion and Future Work

We demonstrated the feasibility of a system to annotate signs with camera phones that does not require any object markers. We showed some of the practical issues arising from the use of image matching algorithms that lie at the core of our approach. Although we did not consider runtime performance in our current implementation, fast response times are crucial. We will therefore focus on improving performance by the use of data structures that are more suitable for this type of application. While letting users mark the corners of the object of interest allowed us to use very simple pixel-based matching algorithms, SIFT's performance turned out superior. In future work we will therefore investigate more deeply algorithms that are based on local features [6], which will also allow us to relax corner marking requirements.

Acknowledgments

We thank Sandro Olibet for implementing the prototype system and evaluating the image matching algorithms in his semester project.

References

1. J. Barton, P. Goddi, and M. Spasojevic. Creating and experiencing ubimedia. HP Labs Technical Report HPL-2003-38, 2003.
2. C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. *Computer Graphics*, 29(Annual Conference Series):277–286, 1995.
3. T. Kindberg, E. Tallyn, R. Rajani, and M. Spasojevic. Active photos. In *DIS '04: Proceedings of the 2004 conference on Designing interactive systems*, pages 337–340. ACM Press, 2004.
4. P. Ljungstrand, J. Redström, and L. E. Holmquist. Webstickers: using physical tokens to access, manage and share bookmarks to the web. In *DARE '00: Proceedings of DARE 2000 on Designing augmented reality environments*, pages 23–31, New York, NY, USA, 2000. ACM Press.
5. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
6. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, Jan. 2006.
7. N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems Man Cybernet*, 9(1):62–69, 1979.
8. J. Rekimoto, Y. Ayatsuka, and K. Hayashi. Augment-able reality: Situated communication through physical and digital spaces. In *ISWC '98: Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, pages 68–75. IEEE Computer Society, 1998.
9. M. Rohs and J. Bohn. Entry points into a smart campus environment – overview of the ETHOC system. In *IWSAWC '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*, pages 260–266. IEEE Computer Society, 2003.
10. M. Rohs and C. Roduner. Camera phones with pen input as annotation devices. In *Pervasive 2005 Workshop on Pervasive Mobile Interaction Devices (PERMID)*, Munich, Germany, May 2005.
11. M. Smith, D. Davenport, H. Hwa, and L. Mui. The annotated planet: A mobile platform for object and location annotation. In *1st Int. Workshop on Ubiquitous Systems for Supporting Social Interaction and Face-to-Face Communication in Public Spaces at UbiComp 2003*, October 2003.
12. M. S. Smith, D. Davenport, H. Hwa, and T. C. Turner. Object auras: A mobile retail and product annotation system. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 240–241. ACM Press, 2004.
13. R. Want, K. P. Fishkin, A. Gujar, and B. L. Harrison. Bridging physical and virtual worlds with electronic tags. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 370–377. ACM Press, 1999.