Object Recognition for the Internet of Things

Till Quack[1], Herbert Bay[1], and Luc Van Gool[2]

[1] ETH Zurich, Switzerland
{quack,bay}@vision.ee.ethz.ch
[2] KU Leuven, Belgium
luc.vangool@esat.kuleuven.be

Abstract. We present a system which allows to request information on physical objects by taking a picture of them. This way, using a mobile phone with integrated camera, users can interact with objects or "things" in a very simple manner. A further advantage is that the objects themselves don't have to be tagged with any kind of markers. At the core of our system lies an object recognition method, which identifies an object from a query image through multiple recognition stages, including local visual features, global geometry, and optionally also metadata such as GPS location. We present two applications for our system, namely a slide tagging application for presentation screens in smart meeting rooms and a cityguide on a mobile phone. Both systems are fully functional, including an application on the mobile phone, which allows simplest point-and-shoot interaction with objects. Experiments evaluate the performance of our approach in both application scenarios and show good recognition results under challenging conditions.

1 Introduction

Extending the Internet to physical objects - the Internet of Things - promises humans to live in a smart, highly networked world, which allows for a wide range of interactions with this environment. One of the most convenient interactions is the request of information about physical objects. For this purpose several methods are currently being discussed. Most of them rely on some kind of unique marker integrated in or attached to the object. Some of these markers can be analyzed using different kinds of wireless near field communication (for instance RFID tags [24] or Bluetooth beacons [11]), others are visual markers and can be analyzed using cameras, for instance standard 1D-barcodes [2] or their modern counterparts, the 2D codes [21].

A second development concerns the input devices for interaction with physical objects. In recent years mobile phones have become sophisticated multimedia computers that can be used as flexible interaction devices with the user's environment. Besides the obvious telephone capabilities, current devices offer integrated cameras and a wide range of additional communication channels such as Bluetooth, WLAN or access to the Internet. People are used to the device they own and usually carry it with them all day. Furthermore, with the phone-number,

C. Floerkemeier et al. (Eds.): IOT 2008, LNCS 4952, pp. 230–246, 2008.

a device is already tied to a specific person. Thus it is only natural to use the mobile phone as a personal input device for the Internet of things.

Indeed, some of the technologies mentioned above have already been integrated in mobile phones, for instance barcode readers or RFID readers. The ultimate system, however, would not rely on markers to recognize the object, but rather identify it by its looks, i.e. using visual object recognition from a mobile phone's camera image. Since the large majority of mobile phones contain an integrated camera, a significant user base can be addressed at once. With such a system, snapping a picture of an object would be sufficient to request all the desired information on it. While this vision is far from being reality for arbitrary types of objects, recent advances in the computer vision field have led to methods which allow to recognize certain types of objects very reliably and "hyperlink" them to digital information.

Using object recognition methods to hyperlink physical objects with the digital world brings several advantages. For instance, certain types of objects are not well suited to attach markers. This includes tourist sights, which are often large buildings and a marker might only be attached at one or few locations at the building, an experiment which has been attempted with the Semapedia project [1]. Furthermore, a user might want to request information from a distance, for instance for a church tower which is up to several hundred meters away. But even if the object is close, markers can be impractical. A barcode or RFID attached to the label of an object displayed in the museum would be difficult to access if the room is very crowded. Taking a picture of the item can be done from any position where it is visible. Furthermore, consistent tagging the objects is often difficult to achieve. One example are outdoor advertising posters. If a poster company wanted to "hyperlink" all their poster locations, they would have to install an RFID or bluetooth beacon in each advertising panel or attach a barcode to each of them, which requires a standardized system and results in costs for installation and maintenance. Another field of application are presentation screens in smart meeting rooms or information screens in public areas. The content displayed on the screen is constantly changing and it would be a involved process to add markers to all displayed content.

Using object recognition to interact with these objects requires only a database of images. That being said, object recognition does not come without restrictions, either. For instance, it is currently (and maybe always) impossible to discriminate highly similar objects, such as two slightly different versions of the same product in a store. Furthermore, efficient indexing and searching visual features for millions or billions of items is still a largely unsolved problem.

In this paper we present a method and system enabling the Internet of Things using object recognition for certain types of objects or "things". At the core of our server-side system lies a retrieval engine which indexes objects using scale invariant visual features. Users can take a picture of an object of interest, which is sent to the retrieval engine. The corresponding object is recognized and and an associated action is executed, e.g. a web-site about the object is opened. The

[1] http://www.semapedia.org

Fig. 1. The user "tags" a presented slide using our mobile application by taking a picture (left), which is automatically transmitted to the server and recognized (middle), a response is given in an automatically opened WAP browser (right).

system is completed with a client-side application which can be installed on a mobile handset and allows true point-and-shoot interaction with a single click.

We present two fully functional applications, which demonstrate the flexibility of the suggested approach. The first one is slide tagging in smart meeting rooms. Users have the ability to "click" on slides or sections of slides that are being presented to record them for their notes or add tags. The second application is a cityguide on the mobile phone. Users have the possibility to take a picture of a sight, send it to a recognition service, and receive the corresponding Wikipedia article as an answer. For this application, the search space is limited by integrating location information, namely cell-tower ids or GPS.

Both systems are experimentally evaluated in different dimensions, including different phone models with different camera qualities, for the trade-offs using different kinds of search space restriction (geographic location etc.), and with and without projective geometry verification stage.

The remainder of this paper is organized as follows: we start with an overview over related work in section 2. The main body of the paper is built around the two applications presented, namely hyperlinked slides for interactive meeting rooms in section 3 and hyperlinked buildings for a cityguide in section 4. Each of these sections discusses method and implementation, followed by an experimental evaluation of the respective system. Finally, conclusions and outlook are presented in section 5.

2 Related Work

Our method can be related to other works in several aspects. One aspect covers work related to our smart meeting room application, for instance the use of camera-equipped mobile phones as an interaction device for large screens. Here, Ballagas et al. have suggested a system [4] which allows users to select objects on large displays using the mobile phone. However, their method relies on additional 2D barcodes to determine the position of the camera and is meant to use the mobile phone like a computer mouse in order to drag and drop elements on the screen. Very recently, in [7] a system similar to ours has been proposed for recognizing icons on displays. While the screens are conceptually similar to the ones used in meeting rooms, we are not aware of any other work that has proposed using camera-equipped mobile phones for tagging or retrieval of slides in smart meeting rooms. The most similar works in that respect deal with slide retrieval from stationary devices. For instance, Vinciarelli et al. have proposed a system [23] which applies optical character recognition (OCR) to slides captured from the presentation beamer. Retrieval and browsing is done with the extracted text, i.e. the method cannot deal with illustrations or pictures in the slides. SlideFinder [18] is a system which extracts text and image data from the original slide data. Image retrieval is based on global color histograms and thus limited to recognize graphical elements or to some extent the global layout of the slide. Using only the stored original presentation files instead of using the captured image data does not allow to synchronize the slides to other meeting data such as recorded speech or video. Both systems are only meant for query-by-keyword retrieval and browsing from a desktop PC. While our system could also be used for off-line retrieval with query-by-example, we focus on tagging from mobile phones. This requires the identification of the correct slide reliably from varying viewpoints, which would not be possible with the cited approaches.

Another aspect that relates to our work are guiding applications on mobile devices. Bay et al. have suggested a museum guide on a tablet PC [5]. The system showed good performance in recognizing 3D exhibition objects using scale invariant local features. However, in their system the whole database resisted on the client device, which is generally not possible for smaller devices such as mobile phones and larger databases. A similar system on a mobile phone, but with somewhat simpler object recognition is the one proposed in [12]. The suggested recognition relies on simple color histograms, which turns out not to be very robust to lighting changes in museum environments. Discriminating instances of the objects in our applications, namely slides or outdoor images of touristic sights, is even less reliable with global color histograms.

The work most similar to our city guide application is maybe [20]. Similar to the cityguide application presented in this paper, the authors also suggest a cityguide on a mobile phone using local features. However, their focus is on improving recognition capabilities using informative and compact iSift features instead of SIFT features. Our work differs significantly in several points: we use multiple view geometry to improve recognition, we rely on SURF features (which are also more compact and faster than SIFT features), and we also investigate

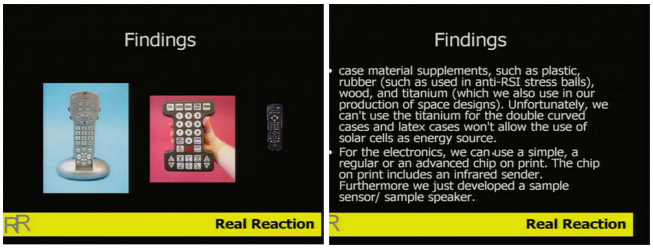

Fig. 2. Typical presentation slides from the AMI corpus [8] database

numerically the effects of restriction by GPS or cell ids on the recognition rate and matching speed. That is, instead of improving the features themselves, we add a global geometry filter as a final verification stage to the recognition system. Finally, the test databases we propose contains images taken from viewpoints with much larger variation than the databases used in [20].

The main contributions of this work are thus: a full object recognition system pipeline, including a server side recognition server and a client side software for single-click interaction with the environment; a complete object recognition pipeline for the Internet of Things, which starts with local feature correspondences, verification with projective geometry, and search space restriction by multimodal constraints, such as GPS location; the implementation evaluation for two sample applications, namely slide tagging and bookmarking in smart meeting rooms, as well as a cityguide application for the mobile phone; last but not least, for both cases the evaluation on challenging test datasets.

3 Hyperlinked Slides: Interactive Meeting Rooms

Today's meeting rooms are being equipped with an increasing number of electronic capturing devices, which allow recording of meetings across modalities [1,3]. They often include audio recording, video recording, whiteboard capturing and, last but not least, framegrabbing from the slide projector. These installations are usually deployed to facilitate two tasks: allowing off-line retrieval and browsing in the recorded meeting corpus and turning the meeting rooms into smart interactive environments. In the work at hand, we focus on the captured presentation slides which are a central part of today's presentations. As shown in figure 2, the slides usually contain the speaker's main statements in written form, accompanied by illustrations and pictures, which facilitate understanding and memorizing the presentation. Indeed, the slides can be seen as the "glue" between all the recorded modalities. Thus, they make a natural entry point to a database of recorded presentations.

A typical usage scenario for our system is as follows: Using the integrated camera of her mobile phone, an attendee to a meeting takes a picture of a slide which is of interest to her. The picture is transmitted to a recognition server over a mobile Internet connection (UMTS, GPRS etc.). On the server, features are

extracted from the picture and matched to the database of captured slides. The correct slide is recognized, added to the users personal "bookmarks", and she receives a confirmation in a WAP browser on her mobile phone. Note that the messaging from the phone can be done using standard MMS or using a custom client-side application which we programmed in C++ on the Symbian platform. Figure 1 shows screenshots of our mobile application for a typical usage scenario.

Back at her PC, the user has access to all her bookmarked slides at any time, using a web-frontend which allows easy browsing of the slides she bookmarked. From each bookmarked slide she has the possibility to open a meeting browser which plays the other modalities, such as video and audio recordings, starting at the timepoint the slide was displayed. By photographing only a section of a slide, the user has also the possibility to highlight certain elements (both text or figures) - in other words, the mobile phone becomes a digital marker tool.

Please note that one could assume that a very simple slide bookmarking method could be designed, which relies only on timestamping. The client-side would simply transmit the current time, which would be synchronized with the timestamped slides. Our system does not only allow more flexible applications (the beforementioned "hightlighting" of slide elements) but is robust towards synchronization errors in time. In fact, using a "soft" time restriction of some minutes up to even several hours, would make our system more scalable and unite the best of both worlds.

The basic functionality of the proposed slide recognition system on the server is as follows: for incoming queries, scale invariant local features are extracted. For each feature a nearest neighbor search in the reference database of slides is executed. The resulting putative matches are verified using projective geometry constraints. The next two subsections describe these steps in more detail.

3.1 Slide Capturing and Feature Extraction

We start from a collection of presentation slides which are stored as images. This output can be easily obtained using a screen capture mechanism connected to the presentation beamer. From the image files, we extract scale invariant features around localized interest points. In recent years significant progress has been made in this field and has led to a diverse set of feature extraction and description methods [16,6,17], which have been successfully applied in domains such as video retrieval [22], object class recognition [15] etc. It turns out that such local features cannot only be used to describe and match objects and scenery, but work also reliably for text such as license plates [9]. Thus, this class of features is a good choice for description of the slide content which contains both text and visual data such as pictures and charts. Furthermore, as opposed to global features proposed in [18,12] they also allow the user to photograph specific sections or elements of a slide as a query to our system. In our implementation we use the publicly available SURF [6] detector and descriptor combination. This choice was motivated by the fast computation times and competitive recognition performance shown in [6]. The output of the SURF detector consists of 64-dimensional feature vector for each detected interest point in an image.

3.2 Slide Recognition System

The slide recognition approach consists of two steps: feature matching and global geometric verification. For the feature matching we compare the feature vectors from the query image to those of the images in the database. More precisely, for each 64-dimensional query vector, we calculate the Euclidean distance to the database vectors. A match is declared if the distance to the nearest neighbor is smaller than 0.7 times the distance to the second nearest neighbor. This matching strategy was successfully applied in [16,6,5,17].

Finding the best result could now be done by just selecting the query-database pair, which receives the highest number of matches. However, without verification of the geometric arrangement of the matched interest points, the wrong query-database pair may be selected. This is particularly true in our case, where we have a high number of matches stemming from letters in text parts of the slides. These matches are all "correct" on the feature level, but only their consistent arrangement to full letters and words is correct on the semantic level.

To solve this problem, we resort to projective geometry. Since the objects (the slides) in the database are planar, we can rely on a 2D homography mapping [13] from the query image to a selected candidate from the database in order to verify the suggested matching. That is, the set of point correspondences between the matched interest points from query image $\mathbf{x}_i^q$ and database image $\mathbf{x}_i^d$ must fulfill

$$H\mathbf{x}_i^q = \mathbf{x}_i^d \; i \in 1 \ldots 4 \tag{1}$$

where H is the $3x3$ homography matrix whose 8 degrees of freedom can be solved with four point correspondences $i \in 1 \ldots 4$. To be robust against the beforementioned outliers we estimate H using RANSAC [10]. The quality of several estimated models is measured by the number of inliers, where an inlier is defined by a threshold on the residual error. The residual error for the model are determined by the distance of the true points from the points generated by the estimated H. The result of such a geometric verification with a homography is shown in Figure 6.

3.3 Experiments

For our experiments we used data from the AMI meeting room corpus [8]. This set contains the images of slides which have been collected over a extended period using a screen-capture card in a PC connected to the beamer in the presentation room. Slides are captured at regular time intervals and stored as JPEG files. To be able to synchronize with the other modalities (e.g. speech and video recordings), each captured slide is timestamped.

To create the ground truth data, we projected the slides obtained from the AMI corpus in our own meeting room setting and took pictures with the integrated camera of two different mobile phone models. Namely, we used a Nokia N70, which is a high-end model with a 2 megapixel camera, and a Nokia 6230, which is an older model with a low quality VGA camera. We took 61 pictures

Fig. 3. Examples of query images, from left to right: with compositions of text and image, taken from varying viewpoints, at different camera zoom levels or may contain clutter, example which select a specific region of a slide, or contain large amounts of text.

with the N70 and 44 images with the Nokia 6230 [2]. Figure 3 shows some examples of query images. The reference database consists of the AMI corpus subset for the IDIAP scenario meetings, which contains 1098 captured slide images.

We extracted SURF features from the reference slides in the database at two resolutions, 800x600 pixels and 640x480 pixels. For the 1098 slides this resulted in $1.02 * 10^6$ and $0.72 * 10^6$ features, respectively. For the SURF feature extraction we used the standard settings of the detector which we downloaded from the author's website.

The resolutions of the query images were left unchanged as received from the mobile phone camera. We ran experiments with and without homography check, and the query images were matched to the database images at both resolutions. A homography was only calculated if at least 10 features matched between two slides. If there were less matches or if no consistent homography model could be found with RANSAC, the pair was declared unmatched. If there were multiple matching slides, only the best was used to evaluate precision. Since the corpus contains some duplicate slides, a true match was declared if at least one of the duplicates was recognized.

Table 1 shows the recognition rates, for the different phone models, different resolutions and with and without homography filter. At 800x600 resolution, the homography filter gives an improvement of about 2% or 4% for each both phone type, respectively. The recognition rate with a modern phone reaches 100%, the lower quality camera in the older 6230 model results in lower recognition rates. The results for the 640x480 database confirm the results of the 800x600 case, but achieve overall lower recognition scores. This is due to the fact, that at lower resolution fewer features are extracted.

4 Hyperlinked Buildings: A Cityguide on a Mobile Phone

The second scenario we present in this paper deals with a very different kind of "things". We "hyperlink" buildings (tourist sights etc.) to digital content. Users

[2] The query images with groundtruth are made available for download under http://www.vision.ee.ethz.ch/datasets/.

Table 1. Summary of recognition rates for slide database

	Prec. with Geometry Filter		Prec. without Geometry Filter	
	800x600	640x480	800x600	640x480
Nokia N70	100%	98,3%	98,3%	96,7%
Nokia 6230	97,7%	93,2%	91%	86,3%

can request information using an application on their mobile phone. The interaction process, the software and user interface are very similar to the meeting room scenario. However, this time the number of objects is nearly unlimited, if the application is to be deployed on a worldwide basis. To overcome the resulting scalability problems, we restrict the search space geographically. That is, we restrict the visual search to objects in the database, which lie in the geographic surroundings of the user's position.

In the following sections we describe this approach in more detail and evaluate its performance.

4.1 Visual Data and Geographic Location

From the user perspective, the interaction process remains the same as in the meeting room scenario: by the click of a button on the mobile phone, a picture is taken and transmitted to the server. However, unlike in the meeting room application, the guide client-side application adds location information to the request. This information consists of the current position read from an integrated or external (bluetooth) GPS device and of the current celltower id the so called CGI (Cell Global Identity).

This combination of a picture and location data forms a perfect query to search for information on static, physical objects. As mentioned before, location information alone would in general not be sufficient to access the relevant information: the object of interest could be several hundred meters away (e.g. a church tower), or there could be a lot of objects of interest in the same area (e.g. the St. Mark's square in Venice is sourrounded by a large number of objects of interest). Furthermore, in urban areas with tall buildings and narrow roads, GPS data is often imprecise. On the other hand, relying on the picture only would not be feasible either: the size of the database would make real-time queries and precise results very difficult to achieve.

After the query has been processed, the user receives the requested information directly on the screen of her mobile phone. In our demo application we open a web browser with the Wikipedia page corresponding to the object. This is illustrated in Figure 4.

4.2 System Design

The cityguide system consists of a server side software and a client-side software on the mobile phone.

Fig. 4. Client software for the cityguide application: the user snaps a picture, waits a few seconds, and is redirected to the corresponding Wikipedia page

The server side elements consist of a relational database for storage of image metadata (GPS locations, cell information etc.) and information about the stored sights. We used mySQL for this purpose. The image recognition is implemented as a server in C++ which can be accessed via HTTP.

Queries from the client-software are transmitted to the server as HTTP POST requests. A middleware written in PHP and Ruby restricts the search by location if needed and passes this pre-processed query to the recognition server. The associated content for the best match is sent back to the client software and displayed in an automatically opened browser, as shown in figure 4.

Client software on the mobile phone was implemented both in Symbian C++ and Java[3]. Note that the feature extraction of the query happens on the server side, i.e. the full query image is transmitted to the server. It is also possible to extract SURF features on the mobile phone and then transmit them as a query to the server. An implementation of this method showed, that SURF feature extraction on the phone is currently too slow: our un-optimized version in Symbian C++ on a Nokia 6630 required about 10 seconds to calculate the query features. In contrast, on a modern PC SURF feature extraction takes a few hundred ms [6]. Since the SURF features are not much more compact than the original image (several hundred 64 dimensional feature vectors per image), the main advantages of feature extraction on the phone would be increased privacy (only features transmitted instead of image) and the possibility to give a user

[3] Unfortunately, only the Symbian version allows access to the celltower ids.

instant feedback if a query image contained too few features, for instance due to blur, lack of texture, or low contrast due to back light.

Alternatively our system can also be accessed using the Multimedia Message Service MMS. A picture is transmitted to the server by sending it as an MMS message to an e-mail address. The response (Wikipedia URL) is returned as an SMS message.

4.3 Object Recognition Method

The data from the client-side application are transmitted to the recognition server, where a visual search restricted by the transmitted location data is initiated. If GPS data is used, all database objects in a preset radius are searched (different radii are evaluated in the experimental section of this paper). If only cell-tower information is used, the search is restricted to the objects annotated with the same CGI string.

The object recognition approach is very similar to the method discussed for the meeting room slides. That is, putative matches between pairs of query and databases images are found by nearest neighbor search for their SURF [6] descriptors. These putative matches are validated with a geometry filter. However, since we deal with 3-dimensional objects in the cityguide application, the precise model is now the $3x3$ Fundamental matrix F instead of the Homography matrix H [13]. The Fundamental matrix maps points in one image to epipolar lines another view. Residual errors for the models are thus determined by the distance of the true points from the epipolar lines generated by the estimated F [13].

From a practitioners point of view, for objects such as buildings which consist basically of multiple planes (facades) one can approximate the results by using a homography nevertheless, which requires less point correspondences. The estimation of the model from putative point correspondences can be done with RANSAC [10] in both cases.

Note that the model is particularly important to filter out false positive recognitions: Especially on structures on buildings, there are a lot of repeated patterns which match between different buildings. Only their correct arrangement in space or the image plane, respectively allow for a robust decision if an object was truly detected. Simply setting a threshold on the number matches is dangerous particularly, since discriminating a false positive recognition (e.g. a query image of an building which is not even in the database) from a query with few matches due to challenging conditions (e.g. image taken from a distance) is infeasible.

4.4 Experiments

To evaluate the proposed method, we collected a database of 147 photos covering 9 touristic sights and their locations. The 147 images cover the 9 objects from multiple sides, at least 3 per object. The database images were taken with a regular point-and shoot camera. To determine their GPS location and CGIs (cell tower ids) we developed a tracker application in Symbian C++ which runs

on a mobile phone and stores the current GPS data (as obtained from an external bluetooth GPS device) and CGI cell information at regular time intervals. This log is synchronized by timestamps with the database photos.

We collected another 126 test (query) images, taken with different mobile phones (Nokia N70 and Nokia 6280, both with 2 Megapixel camera) at different days and times of day, by different users and from random viewpoints. Of the 126 query images 91 contain objects in the database and 35 contain images of other buildings or background (also annotated with GPS and cellid). This is an important to test the system with negative queries, an experiment which has been neglected in several other works. Compared to the MPG-20 database [4] we have fewer object but from multiple sides (in total about 30 unique representations), more challenging viewpoints for each side (distance up to 500 meters), full annotation with both GPS data and celltower ids, and more than 4 times as many query images. The database with all annotations (GPS, cellids, objects Wikipedia pages etc.) is available for download under [5]. Both database and query images were re-scaled to $500x375$ pixels. (Sample images from the database are visible in Figure 7 and are discussed a few paragraphs below).

Note that the CGI (Cell Global Identity) depends on the network operator, since each operator defines its own set of cell ids. If the operator does not release the locations of the cells (which is common practice in many countries for privacy reasons), we have to find a mapping between the cellids of different operators. We achieved such an experimental mapping by using our tracker application: tracks obtained with SIM cards of different mobile network operators were synchronized by their GPS locations: if GPS points were closer than 50m a correspondence between the respective cell-ids was established. This mapping is far from complete, but it simulates an approach which is currently followed by several initiatives on the Web.

We present experiments for three scenarios: linear search over the whole database without location restriction, restriction by GPS with different search radii, and restriction by cellid. For all cases we compare the trade-off between search time and recognition rate. A pair of images was considered matched, if at least

Table 2. Summary of recognition rates for cityguide

	Prec. with Geometry Filter		Prec. without Geometry Filter	
	Rec. rate	Avg. Matching Time	Rec. rate	Avg. Matching Time
Full database linear	88%	5.43s	67.4%	2.75s
GPS 300m Radius	89.6%	3.15s	76.1%	1.62s
Cell id	74.6%	2.78s	73%	1.34s

20 features matched. From the images which fullfiled this criterion the one with the most matches was returned as a response. Table 2 summarizes the results.

[4] http://dib.joanneum.at/cape/MPG-20/
[5] http://www.vision.ee.ethz.ch/datasets/

For the baseline, linear search over the whole database without geometry filter we achieve 67.4% recognition rate. This value is outperformed by over 20% with the introduction of the geometry filter, resulting in 88% recognition rate. This is due to the removal of false positive matches. However, the improved precision comes at a price in speed.

Restricting search by GPS position with a radius of 300 meters is about 40% faster while increasing precision slightly for the case with geometry filter and more substantially for the case without filter. Restriction by celltower CGI is slightly faster but significantly worse in precision. This seems mostly due to the fact, that our CGI correspondences for different operators might be incomplete. For a real world application where an operator would hopefully contribute the cell-id information or a search radius bound by GPS coordinates we would thus expect better results.

Overall the best results are achieved with GPS and a rather large radius of several hundred meters. In figure 5 we plot the precision versus time for different radii. At 100 meters we retrieve most of the of the objects correctly, but only between 300 and 500 meters we achieve the same recognition rates as for linear search, however at significantly higher speed. In fact, this speed-up over linear search will obviously be even larger, the more items are in the database. The recognition times can be further sped up with a suitable indexing structure such as [14,19]. We have compared several methods, however the results are preliminary and beyond the scope of this paper.

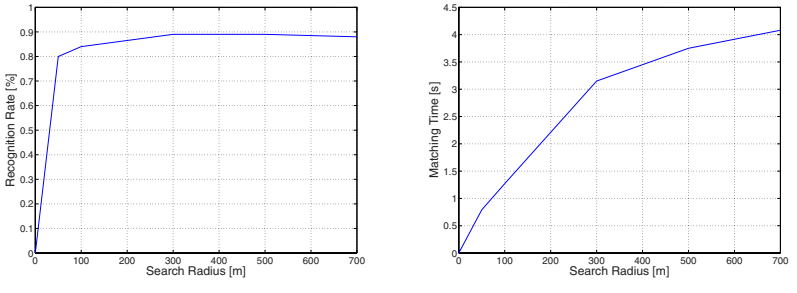

Fig. 5. Recognition rate (left) and matching time (right) depending on radius around query location

Visual results are shown in Figure 7. Section (a) shows query images in the left column and best matching database images for each query in the right column. Note the distance of the query image to the database image in the first row and the zoom and low contrast of the query in the second row. Section (b) contains a query image at the top and the best database match at the bottom. Besides the viewpoint change and occlusion through the lamp and railing, note that query and database image have very different clouds and lighting since they were taken several weeks apart. Section (c) shows an other query database pair, this time for a facade with strong cropping and change of angle. The last image in section (d)

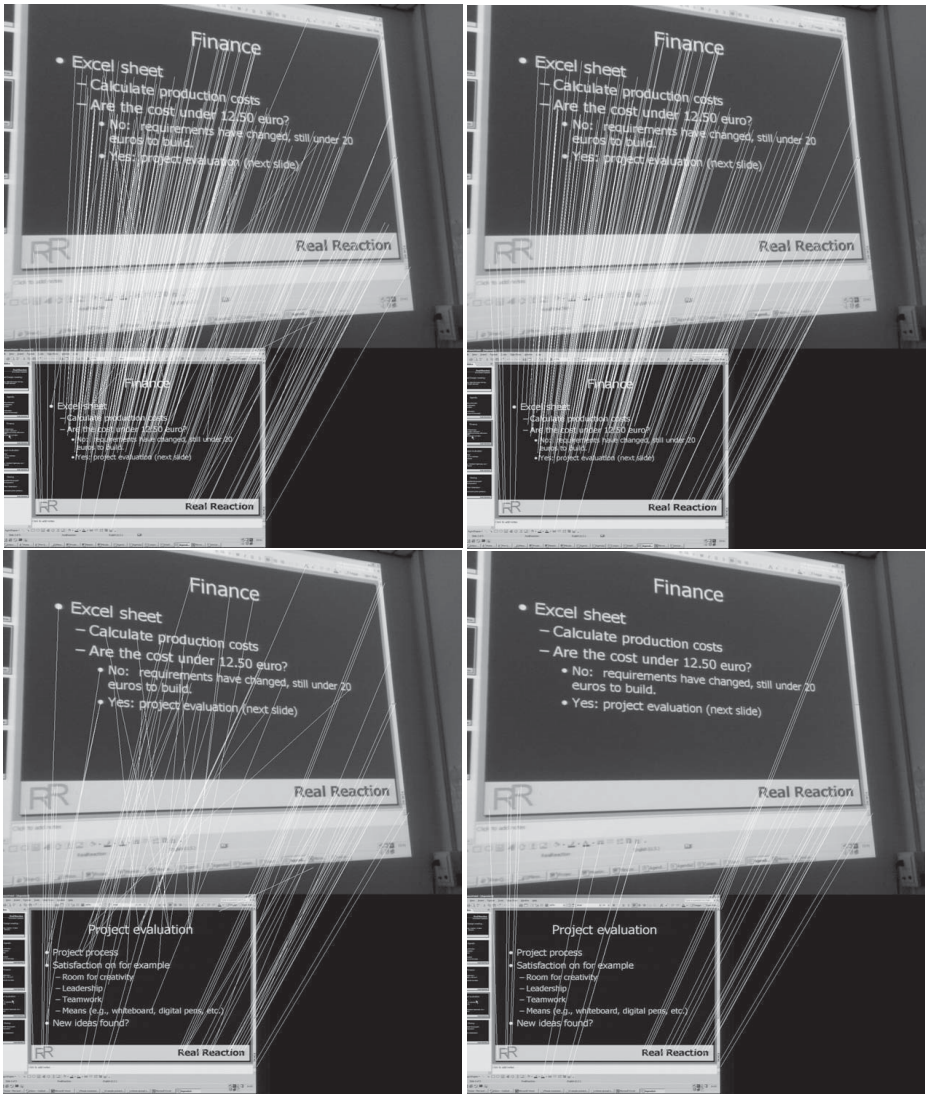

Fig. 6. Geometric verification with a homography. Top rows: matches for a query image with the correct database image. Top left: before homography filter, top right: after homography filter. As the match between the slides is correct most of the putative feature matches survive the homography filter. At the bottom rows we match the same image to a false database image. As can be seen at the bottom left, a lot of false putative matches would arise without geometric verification, in extreme cases their count can be similar to or higher than for the correct image pair. At the bottom right all the false matches are removed, only features from the (correctly) matching frame survive and the discriminance to the correct pair is drastically increased.

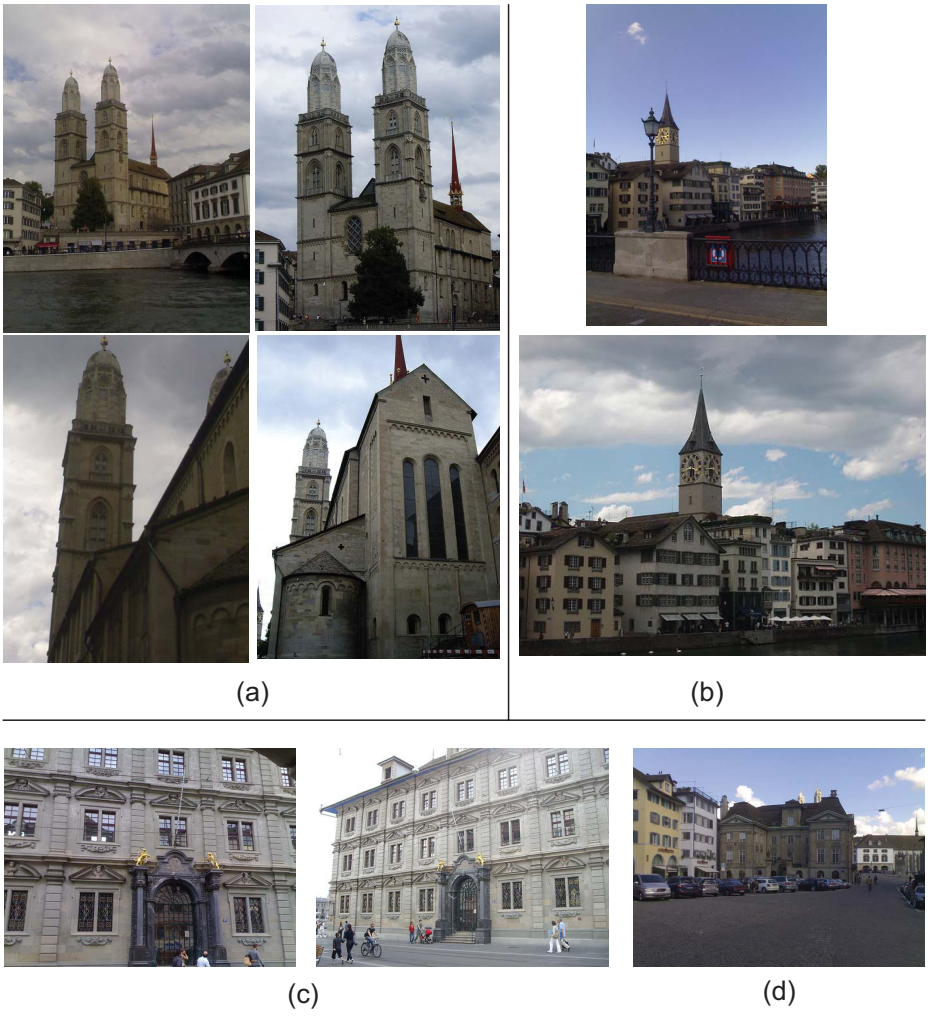

Fig. 7. Result images for the city-guide application, see text for details

contains a typical "negative" query image, which should not return any matching object.

The results highlight the qualities of the suggested approach: the geometry filter improves recognition rates drastically. Restricting search to a geographic radius of a few hundred meters increases speed significantly even in our test database and will be essential for large-scale real world applications. At the same time, the results show that relying only on GPS information (objects up to several dozen meters away) would not be suitable for a real-world guiding application. Being able to "select" the objects with their mobile phone brings significant usability benefits to the user.

5 Conclusions and Outlook

We have presented an approach for object recognition for the Internet of Things, which allows users to request information on objects by taking a picture of them. We have implemented and demonstrated a full system and evaluated its capabilities in two challenging scenarios: slide tagging and bookmarking from screens in smart meeting rooms and a cityguide on a mobile phone. For both applications a server side object recognition system executes the following pipeline: local features are extracted from an incoming image. The features are matched to a database, where the search space is optionally restricted by metadata delivered with the request, for instance by geographic location from GPS coordinates or celltower ids. The resulting candidate matches are verified with a global geometry filter. The system is completed with a client-side software, which transmits query image and metadata such as GPS locations to the server with a single click.

We have demonstrated the flexibility of the suggested approach with an experimental evaluation for both sample applications. To that end, the system was evaluated on two very challenging test datasets. Building on local features and boosting the recognition rate with a geometry filter we achieved very high recognition rates. This approach worked well for both matching of slides with large amounts of text and images of tourist sights from strongly varying viewpoints which underlines the flexibility of the proposed approach. For the especially challenging cityguide application we could find a good balance between performance and recognition rate by restricting the search space using GPS location information.

The results showed, that the Internet of Things by object recognition can be realized already today for certain types of objects. In fact, the system can be seen as a visual search engine for the Internet of Things. Relying just on an image sent from a mobile phone, the system can be easily adopted by both end-users and system providers. With the advance of computer vision methods, we expect a wealth of additional possibilities in the coming years.

Acknowledgements. We acknowledge support by the Swiss project IM2 as well as the "Schweizerische Volkswirtschaftsstiftung". We also thank Fabio Magagna for his help with the implementation of the client-side software.

References

1. Abowd, G.: Classroom 2000: An experiment with the instrumentation of a living educational environment. In: IBM Systems Journal (1999)
2. Adelmann, R., Langheinrich, M., Floerkemeier, C.: A toolkit for bar-code-recognition and -resolving on camera phones – jump starting the internet of things. In: Workshop Mobile and Embedded Interactive Systems (MEIS 2006) at Informatik (2006)
3. Amir, A., Ashour, G., Srinivasan, S.: Toward automatic real time preparation of online video proceedings for conference talks and presentations. In: Hawaii Int. Conf. on System Sciences (2001)
4. Ballagas, R., Rohs, M., Sheridan, J.G.: Mobile phones as pointing devices. In: PERMID 2005 (2005)

5. Bay, H., Fasel, B., Van Gool, L.: Interactive museum guide: Fast and robust recognition of museum objects. In: Proc. Intern. Workshop on Mobile Vision (2006)
6. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
7. Boring, S., Altendorfer, M., Broll, G., Hilliges, O., Butz, A.: Shoot & copy: Phonecam-based information transfer from public displays onto mobile phones. In: International Conference on Mobile Technology, Applications and Systems (2007)
8. Carletta, J., et al. (17 authors): The ami meeting corpus: A pre-announcement. In: Renals, S., Bengio, S. (eds.) MLMI 2005. LNCS, vol. 3869, pp. 28–39. Springer, Heidelberg (2006)
9. Donoser, M., Bischof, H.: Efficient maximally stable extremal region (mser) tracking. In: IEEE Conf. on Computer Vision and Pattern Recognition (2006)
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In: Comm. of the ACM (1981)
11. Fuhrmann, T., Harbaum, T.: Using bluetooth for informationally enhanced environments. In: Proceedings of the IADIS International Conference e-Society 2003 (2003)
12. Fockler, P., Zeidler, T., Bimber, O.: Phoneguide: Museum guidance supported by on-device object recognition on mobile phones. Research Report 54.74 54.72, Bauhaus-University Weimar (2005)
13. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
14. Indyk, P., Motwani, R.: Approximate nearest neighbors: Towards removing the curse of dimensionality. In: STOC 1998: Proceedings of the thirtieth annual ACM symposium on Theory of computing (1998)
15. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: IEEE Conf. on Computer Vision and Pattern Recognition (2005)
16. Lowe, D.: Distinctive image features from scale-invariant keypoints. Intern. Journ. of Computer Vision (2003)
17. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. PAMI 27(10), 1615–1630 (2005)
18. Niblack, W.: Slidefinder: A tool for browsing presentation graphics using content-based retrieval. In: CBAIVL 1999 (1999)
19. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR 2006 (2006)
20. Paletta, L., Fritz, G., Seifert, C., Luley, P., Almer, A.: A mobile vision service for multimedia tourist applications in urban environments. In: IEEE Intelligent Transportation Systems Conference, ITSC (2006)
21. Rohs, M., Gfeller, B.: Using camera-equipped mobile phones for interacting with real-world objects. In: Ferscha, A., Hoertner, H., Kotsis, G. (eds.) Advances in Pervasive Computing, Austrian Computer Society (OCG) (2004)
22. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Intern. Conf. on Computer Vision (2005)
23. Vinciarelli, A., Odobez, J.-M.: Application of information retrieval technologies to presentation slides. IEEE Transactions on Multimedia (2006)
24. Want, R.: Rfid - a key to automating everything. Scientific American (2004)