Enabling Mobile Phones To Support Large-Scale Museum Guidance

We present a museum guidance system called *PhoneGuide* that uses widespread camera equipped mobile phones for on-device object recognition in combination with pervasive tracking. It provides additional location- and object-aware multimedia content to museum visitors, and is scalable to cover a large number of museum objects.

Introduction and Motivation

Mobile phones have the potential of becoming a future platform for personal museum guidance. They enable full multimedia presentations and – assuming that the visitors are using their own devices– will significantly reduce acquisition and maintenance cost for museum operators. However, several technological challenges have to be mastered before this concept can be successful. One of them is the question of how individual museum objects can be intuitively identified before presenting corresponding information.

We describe a lightweight object recognition method that is realized with two-layer neural networks. In contrast to related systems (*see related work section*) that perform computational intensive image processing tasks on remote servers[4] or on high-end mobile devices (such as tablet PCs[14]), our intention is to carry out all computations directly on mobile phones. This ensures little or even no network traffic and consequently eliminates cost for online times.

Normally, the classification rate of a computer vision based recognition system decreases with an increasing number of objects[6]. Using pervasive tracking technology, however, allows considering only a small subset of objects at a time. This is realized by dynamically reconfiguring and retraining the neural network during runtime with objects that are in the visitor's proximity.

Applying pervasive tracking only, as it is done by similar approaches (*see related work section*) does not provide the accuracy to differentiate individual objects that are located within the signal range of the same emitter node (e.g., an RFID tag[9] or a WLAN[10] base station). Combining pervasive tracking with computer vision techniques for on-device object recognition represents a powerful tool with respect to scalability and accuracy. In addition, it prevents from attaching additional identifiers (such as barcode tags[5] or infrared emitters[7,8]) to every single object exhibited in the museum.

In a field survey our system was able to identify 155 real museum exhibits from multiple perspectives with a recognition rate of 95% and a classification speed of less than one second per object. A coarse grid of only eight low-cost Bluetooth emitters distributed over two museum floors was used to achieve these results. Once an object has been recognized, related multimedia presentations such as videos, audio[*], text, computer graphics and images are displayed on the phone.

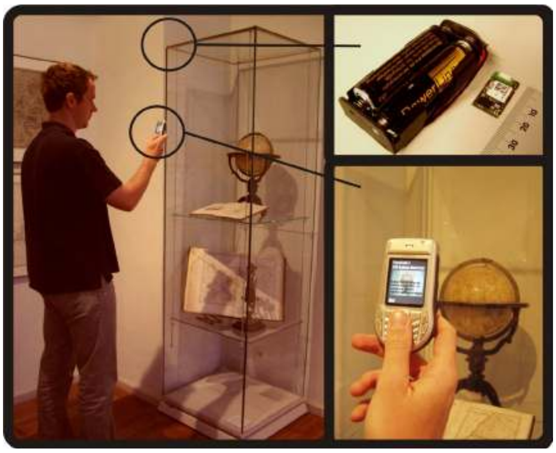

Figure 1: Mobile phone enabled guidance in a museum: On-device object recognition via computer vision combined with pervasive tracking through a coarse grid of Bluetooth emitters (emitter and battery pack shown above).

PERVASIVE TRACKING VIA BLUETOOTH

Context awareness is one of the main goals of ubiquitous computing. In ubiquitous computing context is any information that can be used to describe a situation. Context-aware applications adapt according to the location, nearby people, other accessible devices, time, temperature, etc. Different sensors can be applied for acquiring the actual context.

Several radio frequency emitters can be used for tracking. One of the most promising technologies is radio frequency identifiers (RFID). It requires special hardware for receiving the signals that are not available yet for consumer mobile phones. The same applies for WLAN, although newer phones are already equipped with WLAN chips. Currently, Bluetooth is supported by a wide range of existing mobile phones and can be utilized for pervasive tracking without creating extra cost.

With the aid of a coarse grid of Bluetooth emitters (cf. figure 1) distributed in the museum the visitors' mobile phones can approximate their rough locations:

An asynchronous service running in the background of the main *PhoneGuide* application periodically scans for Bluetooth devices and their unique IDs. If a device is found, a lookup table validates its ID and filters out invalid ones (e.g. discovered Bluetooth devices that do not belong to our tracking grid, such as other mobile phones). The remaining

[*] via integrated speaker or head-set

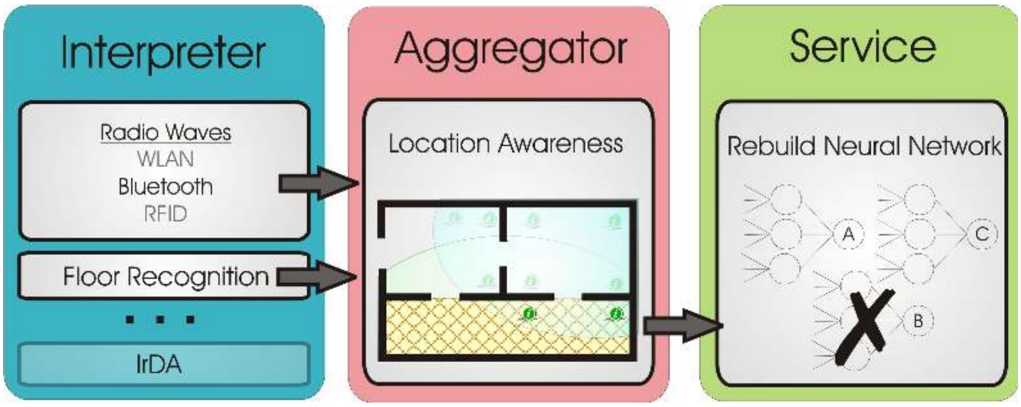

Figure 2: Pervasive tracking framework: Bluetooth tracking and floor recognition are used to dynamically reconfigure and train a neural network depending on the actual location of the visitor.

emitter IDs are added to a temporary device list. At the end of the discovery process a call-back function compares the new set of discovered emitters with the previously discovered set. Any change in the set indicates that other exhibits are now in the proximity of the visitor. This triggers an automatic reconfiguration and retraining of the neural network for adapting the on-device object recognition process to the new environment.

Location Awareness

Depending on its signal strength, every emitter can cover a limited area. Its range is also affected by reflections and abortions of the signal by artefacts, such as walls or people. Different signals of multiple emitters can overlap, and are consequently detectable simultaneously. Thus, an unstructured grid of emitters partitions the environment into different spatial cells of superimposed and single signals.

Unfortunately, current mobile phone APIs do not allow evaluating the strength of the Bluetooth signal. This would improve the tracking precision, as it is the case for similar systems that evaluate WLAN signals[10]. However, determining the actual cell in which the visitor is located provides sufficient information for our approach.

In particular, every recognizable object can be assigned to the corresponding cell in which it is located. Consequently, knowing the cell of the visitor leads to the objects that are in his/her proximity.

Additional context information can be used to refine the fragmentation of the cells, and to provide a higher accuracy. We will explain later how our object recognition technique can be applied for extracting additional state information about the surrounding environment (e.g., by recognizing the floor texture while the visitor is moving) using computer vision.

An aggregator collects the different context information of every interpreter, such as Bluetooth signals, floor texture, etc., and derives the corresponding cell in which the visitor is located (cf. figure 2).

Conjunction (i.e., AND relation) would be a possible concatenation of the different interpreters. Unfortunately the detection of radio signals, such as Bluetooth, is error-prone. Due to dynamic absorption and reflection effects (e.g. by other visitors) not every Bluetooth emitter that might be normally visible can always be detected. A conjunction has no fault tolerance. A possible consequence might be that an object cannot be recognized because it is assumed to be not in the visitor's proximity.

Only *available* context information can be used for location inference. So we use fault-tolerant *implication* instead of conjunction. Thus, if an object is located in a cell defined by two interpreters, but only one provides information, this object *is* used for recognition. Compared to conjunction, implication creates an overhead of potential matches for subsequent object recognition. However, this is essential if the context information is not reliable.

ON-DEVICE OBJECT RECOGNITION

In computer vision, objects can be recognized by classifying images of the objects. The simple comparison of raw pixel data is computationally too expensive and strongly variant to even small changes in the images, such as perspective or lighting. Instead, the images are normally transformed into sets of local[1] or global[2] vectors that describe their content. These feature vectors can then be compared efficiently for finding images of objects with similar features. Closest- neighbor match algorithms are frequently being applied for this task[14] – but are usually inefficient for object recognition:

Recognizing objects from multiple perspectives, would require to store the feature vectors of all possible perspective images of all objects in a database, and to compare them during runtime with the feature vector of the image taken for the object to be recognized. For a large number of N (n=0...N-

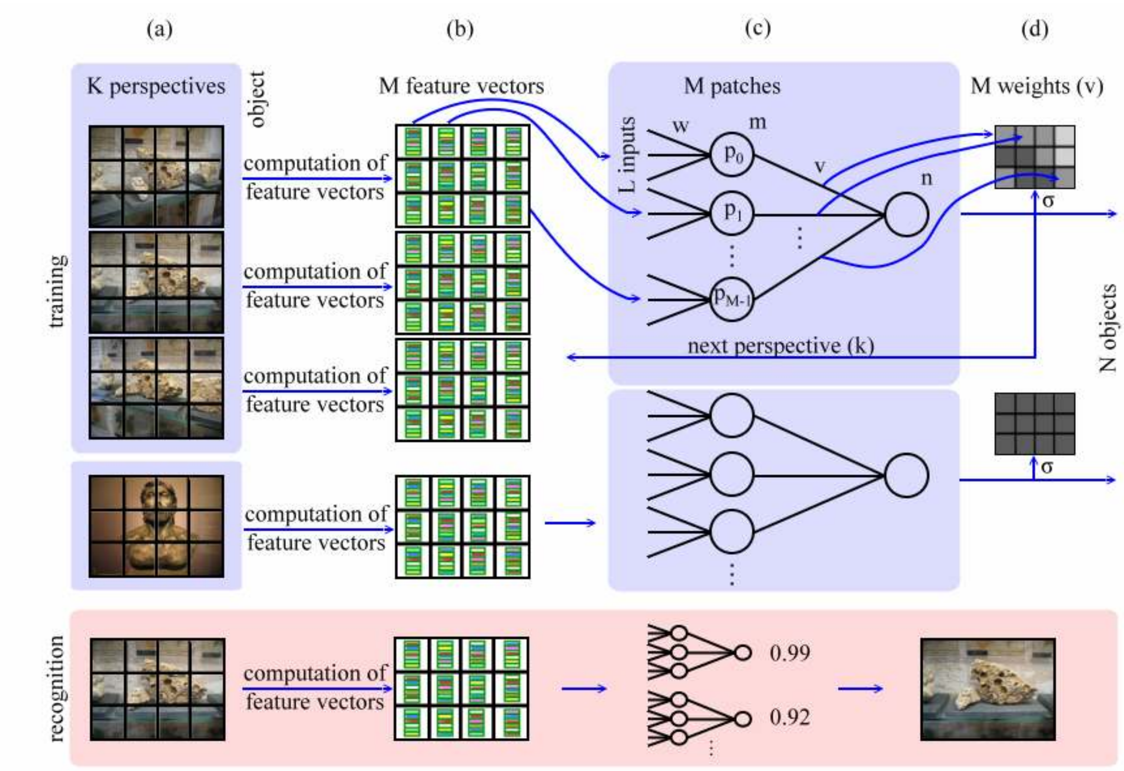
(a)
(b)
(c)
(d)
K perspectives
object
M feature vectors
M patches
M weights (v)
training
computation of feature vectors
computation of feature vectors
computation of feature vectors
computation of feature vectors
L inputs
w
m
p0
p1
pM-1
v
n
σ
next perspective (k)
N objects
σ
recognition
computation of feature vectors
0.99
0.92

Figure 3: Training / recognizing objects: Divide perspective images of object into patches (a), compute feature vectors for patches (b), train TLNN with feature vectors / find maximum excitation of all perceptrons (c), weight patches based on output of first layer / return object based on ID (d).

1) objects this is inefficient on platforms such as mobile phones.

Instead, we follow a linear separation strategy implemented with a *two-layer artificial neural network* (TLNN). The TLNN is trained and executed directly on the phone, rather than on a remote server. Training such a network allows compressing the feature vectors of all perspective images which belong to the same object into a single set of normalized weights. These weight vectors are assigned to a single object, rather than to a single image and serve as a fingerprint for recognizing the object in other images. Thus, weight vectors have the same dimension as feature vectors.

Since we are mainly interested in recognizing the actual object within the image rather than the surrounding environment, the foreground and the background have to be distinguished from each other. This is particularly important because the background can differ much more than the foreground in different perspective images of the same object.

This is realized by assuming that the object is always approximately centered in every image. Segmenting the images into multiple patches allows up- or down-weighting them depending on their information content. As Artiklar et al.[3] we weight each patch individually but apply a recognition on the sum of all patches rather than on each patch individually.

In the following, our recognition approach is described in more detail.

Global Features

An optimal selection of features is essential for achieving a high recognition rate. We have identified and investigated several global features that are suitable for recognition with linear separation strategies. These features describe different normalized color and intensity relations, such as mean, variance or histogram ratios, as well as structural properties, such as edge/non-edge-pixel or horizontal/vertical edge-pixel ratios of the image content. More details are presented in Föckler, et al[6]. In the following, we want to refer to a set of feature values as feature vector (*f*).

Weighting Image Patches

Computer vision based object recognition techniques often apply expensive image segmentation techniques for clipping away the background before classifying the foreground. In contrast to this, our method considers the whole image – regardless of the object's structure. However, important image parts are up-valued while less important ones are down-valued.

The images are segmented into *M* (*m:0...M-1*) rectangular patches of uniform size, while the object must be centered by convention (cf. figure 3). We consider a patch as important, if it contains similar features in multiple perspective images.

Since the background may vary more than the foreground from different perspectives, it is likely that such patches contain the object rather than the background. Note that we do not explicitly separate the background from the foreground. Rather than that we focus on areas that have similar features in different perspective images.

Instead of computing a feature vector for the entire image, one is computed for every image patch. The variance of corresponding patch-individual feature vectors (*f*) for multiple perspective images leads to a weight expressing the patch's importance.

Recognition and Training

For every object to be recognized, the TLNN contains *M* perceptrons in the first-layer – one for each image patch. Each first-layer perceptron has *L* (*l:0...L-1*) input channels – one for each feature value. The *L* features of all *M* patches are passed to their corresponding input channels. The weight vector (*w*) at the first layer inputs are multiplied with their corresponding feature values and the weighted sum over all input channels is the result of each first-layer perceptrons' output channel.

The results at these output channels are passed to a perceptron in the second layer that is responsible for recognizing the associated object. Consequently, the second layer of the TLNN contains *N* perceptrons – one for each object to be recognized. Thus, the entire TLNN consist of *N* object-independent subnets (cf. figure 3).

The input channels of the second layer perceptrons are weighted (with *v*) depending on the corresponding patch's importance. While the weights of the first layer are initialized by the feature values of one arbitrary perspective image, the weights of the second layer are initialized with *1/M*.

For recognition, a new image of an object has to be taken and the feature vectors for all patches are computed. The object is recognized by finding the output value of the second layer with the maximal excitation through the following activation function:

$$\max_{n=0}^{N-1}\left(\sum_{m=0}^{M-1}\left[\left(\sum_{l=0}^{L-1} w_{nml} f_{nml}\right) v_{nm}\right]\right) \qquad (1)$$

If the recognition of a particular object failed its subnet has to be trained with the set of *M* feature vectors of the new image that has caused the failure. The weights at the first layer are updated with the following learning function:

$$w'_{nml} = \frac{w_{nml} + L \cdot \varepsilon \cdot f_{nml}}{\left|w_{nml} + L \cdot \varepsilon \cdot f_{nml}\right|}, \qquad (2)$$

where L is the learning rate (empirical value) and ε is the computed local error value (difference between maximum excitation and computed output).

With the *M* output values p_{nm} of the first layer the weights (v_{nm}) of the second layer are computed as follows:

$$v_{nm} = \frac{\sum_{k=0}^{K-1}(1 - \|\partial_{nmk}\|)}{\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}(1 - \|\partial_{nmk}\|)}, \qquad (3)$$

where ∂_{nmk} is the variance of the output channels at the first layer for *K* (*k:0...K-1*) sequentially trained perspective images.

The training has to be repeated for all perspectives of all objects until the TLNN's weights converge. This can be done automatically or manually.

Dynamic Network Configuration

Having at least one feature vector of all possible perspectives for each recognizable object, an individual TLNN can be configured and trained dynamically depending on the visitor's location. This is possible since the objects that are located within the visitor's proximity are known through the information provided by the pervasive tracking mechanism.

One subnet for each eligible object is created and automatically trained with all feature vectors that are available for this object. The automatic training is repeated until all first-layer weights of the entire network have converged.

Note that the set of feature vectors is created only once (when the system is installed in the museum). They are transmitted to and stored on the visitor's mobile phone together with the presentation content. The dynamic configuration and the training of a particular TLNN is performed continuously and unnoticeable while the visitor is moving through the museum.

An alternative to dynamic network configuration would be, to pre-train a single TLNN for every cell and load the corresponding network depending on the user's location. However, because of instabilities of the Bluetooth signals due to dynamically varying absorption and reflection situations, a particular cell might not be detected. Instead, all potential cells and their corresponding networks could be pre-computed by combining all emitter IDs in every possible variation. The amount of required data would be too large and for a classification too inefficient. A dynamic network configuration adapts to the current visibility situation while keeping the memory and processing requirements at a minimum.

Continuous Recognition

Object recognition is normally triggered by the visitor when pressing a button on the phone to take a picture of the object. The recognition method

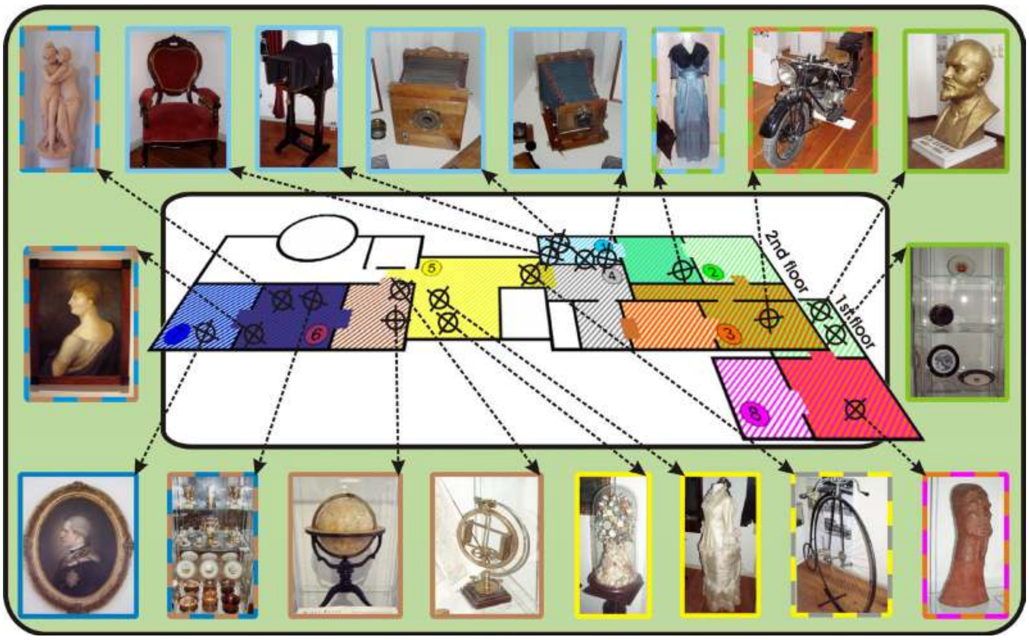

Figure 4: Eight Bluetooth emitters (encircled) placed at different locations in two floors of a museum. A small subset of the 155 objects and their positions are shown. The two-colored hatched areas approximate the signal cells spanned by one or two emitter(s) with the same color(s). The colored frame of each object image also indicates the signal cell in which they are located.

itself, however, performs the entire classification in less than one second on today's consumer phones. This enables an enduring recognition without the need for an explicit trigger event.

As explained earlier, such a *continuous recognition* mode can be used to extract additional context information that –together with other information, such as the Bluetooth signals– supports the estimation of the visitor's location. It can, for instance, recognize specific features that are present only in a particular room – such as unique textures on ceilings or walls, or a particular room illumination, etc.

We used this mode in our experiments to recognize the floor texture while the visitor is moving from exhibit to exhibit – orienting the phone in such a way that its camera is pointed downwards. Different room illuminations are reflected by the floor and consequently support the classification.

The continuous recognition mode usually has to differentiate only a few states – thus its hit rate is high. But when the visitor stops in front of an object and lifts the phone for taking a picture of it can be problematic. The reason is that during this time images of other artefacts (such as a showcase or the object itself) can be captured that lead to misinterpretations of the context and consequently to a wrong location estimation. To overcome this problem, we evaluate the pixel flow in he live video stream during the continuous recognition mode. If there is no or little pixel flow, the phone is not moved (e.g., when targeting at an object). If it is very high the phone is moved quickly (e.g., when lifting it). In both cases, the continuous recognition mode is disabled. In addition, we evaluate the recognition result (i.e., the output value of the second layer). It must be above a predefined threshold to be valid.

FIELD SURVEY

We have evaluated our system in the Museum of the City of Weimar. The museum displays a large and varying pallet of artifacts ranging from furniture, over cloths to pictures in fifteen different rooms on two floors (cf. figure 4).

Our software[†] was preinstalled on Bluetooth-enabled Nokia Series 60 phones (6630 and 6670), providing a 1.3 MPixel camera. Despite the high camera resolution we are using only a 160x120 pixels image resolution for fast feature computations.

To enable pervasive tracking within the museum eight Bluetooth emitters were placed below the ceiling of different rooms in order to ensure a consistent coverage.

Taking three perspective images for each of the 155 objects took less than one hour. The IDs of the emitters were automatically detected during this one-time image acquisition task, and stored together with the corresponding feature vectors (containing 14 feature values for each of the 12 patches of every perspective image). Note that no raw image data is stored at any time. The size of the data set required on the device for this experiment was approximately 626 kilobytes.

During the actual guidance task, individual TLNNs were dynamically configured and automatically trained – depending on the visitors' location. The

[†] Symbian OS or Java

automatic training was stopped if the output excitations of all configured perceptrons were above or equal to 98%, or if a maximum number of 20 training passes was exceeded. The automatic training required approximately 3-10 seconds in our experiments, and was triggered automatically when a visitor moved from one cell to another one. The recognition rate for all 155 objects from multiple perspectives was 95%.

Figure 5 illustrates the number of required training passes with respect to the number of objects being located within the individual cells (shown in figure 4). It is easy to see that the training passes do not necessarily scale with the number of objects being located in a cell. It mainly depends on how well the object features can be separated from each other with a TLNN. Small sets with similar objects (e.g., the plates being displayed in the same showcase, cf. figure 4) might need as many training passes as large sets with different objects. In this case, the automatic training requires more time to converge.

Problematic for the on-device object recognition are varying lighting conditions. Our system would perform worse in outdoor environments. However, the lighting in museums is fairly constant which leads to a stable recognition rate over different day times and days. Self-reflections (e.g., in showcases) or shadows cast by the visitors themselves can sometimes influence the recognition rate. In our experiments this was another reason for an imperfect recognition rate.

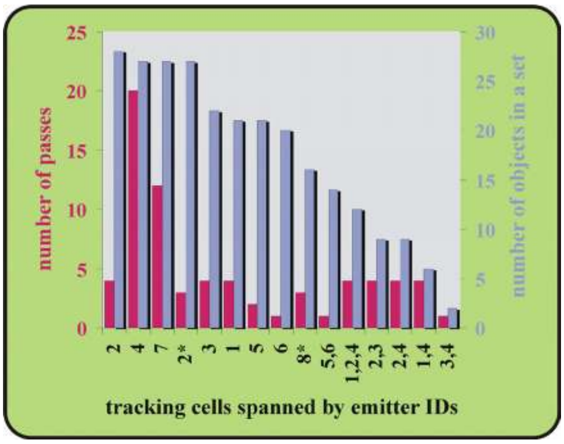

Figure 5: Passes required for automatic training of different objects sets within the corresponding cells shown in figure 4. The asterix (*) indicates cells with a floor texture that differs from the floor texture in all other cells.

Some objects cause strongly varying features for different perspectives (e.g., the motorcycle in figure 4). In such cases, multiple perceptrons can be trained for the same object (e.g., one for each discrete perspective segment), and assigned to the same object ID. This leads to slightly larger networks, but also to high and stable recognition results when moving around an object. For the motorcycle in figure 4, we used 4 perceptrons (two for the front and back views, and two for the side views).

The number of eight Bluetooth emitters for covering 15 rooms appeared to be adequate. It ensured a continuously robust tracking of the visitors. A finer granularity leads to smaller tracking cells containing less objects, and consequently to a higher recognition rate. The floor recognition provided additional location information.

DISCUSSION AND OUTLOOK

Pervasive tracking alone does not provide the required precision for differentiating every single object in a museum. The range of radio or optical signals (e.g., RFID, Bluetooth, infrared or WLAN) is either too large and covers multiple objects simultaneously, or is too small for tracking tasks. The second case, however, would require attaching single emitters with short signal ranges to every individual object and ensuring that their signals can still be received by the visitors' end device. This is a very inefficient approach for a large number of objects.

It was predicted that by the end of the year 2005, over 50% of all mobile phones will be equipped with digital cameras[13]. Thus, object recognition enabled by computer vision techniques has a large potential to overcome these problems. However, object recognition methods do not scale very well. Their recognition rate drops significantly with an increasing number of objects.

We have shown that with a combination of pervasive tracking (using only a coarse grid of emitters) and on-device object recognition a scalable system with a high recognition rate can be realized.

Self-reflections on highly specular surfaces, such as glass, are unavoidable in public museums. These effects can cause the recognition rate to drop. Using camera-phones with integrated flash allows applying so-called "flash-on/flash-off" techniques to eliminate self-reflections in the image. Similar techniques can be used for extracting shadows cast by the visitors.

The number of objects in a cell-individual neural network decreases with an increasing tracking precision. Evaluating the signal strength of the Bluetooth emitters in addition to their IDs would allow determining a visitor's position more precisely. Unfortunately, Symbian OS 7.0 does not provide an API for RSSI yet.

Passive RFID tags are clearly preferred over active tags, such as Bluetooth or infrared emitters. Their low acquisition and maintenance cost allows to distribute a large number of them in a museum to provide a dense tracking grid. However, appropriate RFID readers must become standard equipment for mobile phones first, before being successfully established.

We believe that a personal museum guidance enabled by mobile phones has several advantages over currently applied technology, such as audio guides: First, information is communicated more efficiently through multimedia presentations, including images, video, audio, text, and computer graphics, rather than presenting pure audio content. Second, taking a picture for obtaining information about a particular object is more intuitive than looking up and keying in an abstract number. Third, the museum operators benefits from lower maintenance and acquisition costs for their presentation technology, since the end-devices are provided by the visitors themselves.

In future, entering a museum might automatically trigger the transformation of a visitor's mobile phone into a piece of personal guidance equipment –but only temporary– for the time being inside the museum. This could involve a seamless download process of data and application from a base station, and the automatic disabling of critical functions (such as storing photographs or making/receiving phone calls).

ACKNOWLEDGEMENTS

We thank the Senckenberg Museum of Natural History Frankfurt, the Museum for Pre- und Early History Weimar, the Museum of the City of Weimar, and CellIQ for their support.

RELATED WORK

Object Recognition

Object recognition is a wide field of computer vision. In order to recognize objects in images, global or local features are extracted.

Today, *local features* (e.g., local corner points or image fragments) are mainly used in recognition systems due to their ability to be invariant to scaling and rotation and their support of recognizing partly occluded objects. Lowe[1], for example, presented an algorithm for detecting local scale invariant features based on local extrema found in Gauss-filtered difference images. Later, he demonstrated the possibility to extract highly distinctive features that could be matched in a large database with a high hit rate[15].

Object recognition that is based on *global features* mostly extracts color, texture or structural information from the entire image. For instance, Swain et al[2] presented a recognition system using color histograms. Artiklar et al[3] divides the images into a fixed set of areas. With its global features computed for each area the local distances to corresponding areas of object images stored in a database are determined. A vote on the recognized object is then cast based on a probability contribution.

Object recognition methods that are completely performed locally (i.e., on the mobile device itself) is nearly unexplored. A reason for this is the existing hardware limitations of these devices. Our own previous work[6] describes an on-device object recognitions system using global image features and a single layer neural network to achieve a recognition rate of 91% for no more than 50 museum objects. Related systems used mobile front-end devices for image capturing and simple pre-computation only. The computational expensive classification is then done on stationary back-end servers. This creates additional network traffic. Fritz et al.[4] proposed such a system for recognizing outdoor objects like buildings and statues using a PDA and a wireless connection (WLAN/GPRS/UMTS) to a server. The server classifies the objects and sends back the results to the PDA. Various ongoing initiatives follow the same principle, but use mobile phones instead of PDAs. Lowe's distinctive image feature method[1] is sometimes being applied for recognition on the server side.

Several groups, such as the Semacode Cooperation[5], recognize artificial markers displaying barcodes instead of arbitrary objects. This simplifies the computer vision process, but requires attaching additional labels to all exhibited objects.

Location Awareness

There are various approaches of location-aware frameworks for different scopes: indoor or outdoor environments, consumer applications with mobile direct marketing and payment services, hospital environments, and museum guidance, etc. A good overview about the roles of mobile devices in ubiquitous environments is given in Siegemund, et al.[11].

IrReal[7] is a building information and navigation system based on Palm Pilot PDAs. Several *infrared (IR)* emitters, located throughout the building, stream localized data to nearby devices. This technique, referred to as implicit tracking, does not explicitly estimate the user location, but provides location specific information. The Hippie system[8], as another example, locates the user's position via an IR system installed at entrances of different building sections (such as rooms) and on particular objects. By evaluating the infrared signal, the system can detect the object and a server provides additional information about the object or other places of interest. Infrared emitters are also used by Ciavarella et al. [12] to provide location specific floor maps to the visitors.

One main disadvantage of optical signals, such as infrared, is that the line-of-sight between emitter and receiver must not be occluded. In case of IR the signal range is also small – forcing the visitor to get close to the emitters.

LANDMARC[9] is a location sensing prototype system that uses *Radio Frequency Identification* (RFID) for locating objects inside buildings. The signal strength is not taken into account, but instead a large number of low cost RFID tags are applied to span high resolution grid. RADAR[10] is another RF based system for tracking users inside buildings using standard 802.11 network adapters. In this case, the signal strength of multiple base stations positioned in a given area is taken into account to gain a higher tracking precision. This system combines empirical measurements and signal propagation modeling in order to determine the user location, and enables location-aware services and applications.

Bay et al. have recently published[13] a prototype system that is very similar to our approach: Tablet PCs are used for interactive museum guidance. A variation of Lowe's SWIFT method[15] is used for object recognition, while Bluetooth emitters are used for automatic room detection. They reported to achieve recognition rate of 80% for 22 objects. Two rooms are differentiated by evaluating the IDs of two Bluetooth emitters – one placed in each room. The average detection time of object recognition was 10 seconds on a Tablet PC. Despite the application of high-end hardware, their performance and recognition rate is rather poor. It was shown earlier that in common museum environments, a recognition rate of more than 90% for 50 objects with recognition times of less than one second can be achieved on off-the-shelf mobile phones[6]. The combination of object recognition with pervasive tracking that is presented in this article leads to recognition rates of 95% for 155 objects, and recognition times of still less than one second on common mobile phones. Our method is scalable, and a larger number of objects can be detected without a drop in quality or performance by increasing the number of emitters.

References

1. D.G. Lowe, "Object Recognition from Local Scale-Invariant Features", *Proc. International Conference on Computer Vision*, 1999, pp. 1150-1157.
2. M. Swain and D. Ballard, "Color indexing", *Proc. International Journal of Computer Vision*, vol. 7, no. 1, 1991, pp. 11–32.
3. M. Artiklar, M. Xiaoyan, M. H. Hassoun and P. Watta, "Local voting Networks for Human Face Recognition", *Proc. International Joint Conference on Artificial Neural Networks (IJCNN '03)*, vol.3, 2003, pp. 2140- 2145.
4. G. Fritz, C. Seifert, P. Luley, L. Paletta and A. Almer, "Mobile Vision for Ambient Learning in Urban Environments", *Proc. International Conference on Mobile Learning (MLEARN 2004)*, 2004.
5. Semacode Cooperation, "Semacode", retrieved from WWW, http://www.semacode.org, 2004.
6. P. Föckler, T. Zeidler, B. Brombach, E. Bruns and O. Bimber, "PhoneGuide: Museum Guidance Supported by On-Device Object Recognition on Mobile Phones", *Research Report 54.74 54.72, Bauhaus-University Weimar, February 2005 and submitted to Int. Conference on Mobile and Ubiquitous Multimedia*, 2005.
7. A. Butz, J. Baus and A. Krüger, "Augmenting Buildings with Infrared Information", *Proceedings of the Int. Symposium on Augmented Reality (ISAR 2000)*, IEEE CS Press, pp. 93-96.
8. R. Oppermann and M. Specht, "A context-sensitive nomadic exhibition guide", *Second Symposium on Handheld and Ubiquitous Computing* – HUC2K (2000) pp. 127-149.
9. L. M. Ni, Y. Liu, Y. C. Lau and A. P. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID", *First IEEE International Conference on Pervasive Computing and Communications (PerCom'03),* 2003, p. 407.
10. P. Bahl and V.N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system", *Proceedings of IEEE Infocom 2000, Tel-Aviv, Israel.*
11. F. Siegemund, C. Floerkemeier and H. Vogt, "The value of handhelds in smart environments", *Proc. on Architecture of Computing Systems*, October 2004, pp. 291-308.
12. C. Ciavarella and F. Paternò, "The design of a handheld, location-aware guide for indoor environments", *Proc. Personal and Ubiquitous Computing*, 2004, pp. 82-91.
13. M. Macedonia, "Small is Beautiful", *IEEE Computer*, 2004, vol. 37, no. 12, pp.122-123.
14. H. Bay, B. Fasel, and L. v. Gool, "Interactive Museum Guide", *Proc. of UbiCom*, September 2005.
15. D. G. Lowe. "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, January 2004, vol. 60, no. 2, pp. 91–110.