Pers Ubiquit Comput (2009) 13:165–178
DOI 10.1007/s00779-008-0194-3

ORIGINAL ARTICLE

Adaptive training of video sets for image recognition on mobile phones

Erich Bruns · Oliver Bimber

Received: 2 July 2007 / Accepted: 5 February 2008 / Published online: 5 March 2008

Abstract We present an enhancement towards adaptive video training for PhoneGuide, a digital museum guidance system for ordinary camera-equipped mobile phones. It enables museum visitors to identify exhibits by capturing photos of them. In this article, a combined solution of object recognition and pervasive tracking is extended to a client–server-system for improving data acquisition and for supporting scale-invariant object recognition. A static as well as a dynamic training technique are presented that preprocess the collected object data differently and apply two types of neural networks (NN) for classification. Furthermore, the system enables a temporal adaptation for ensuring a continuous data acquisition to improve the recognition rate over time. A formal field experiment reveals current recognition rates and indicates the practicability of both methods under realistic conditions in a museum.

1 Introduction

Camera-equipped mobile phones represent an ideal platform for mobile personal computer vision (CV) applications. In combination with integrated RF-technologies like Bluetooth or wireless LAN, this opens the opportunity to enhance pervasive localization through CV techniques. In this context we have developed PhoneGuide, a personal museum guidance system for camera-equipped mobile phones based on pervasive tracking, local object recognition, and temporal adaptation.

In the following we want to use the term *object recognition* to refer to the recognition of exhibits. Technically, however, we apply *image recognition*. In our previous work [1, 2] prototypes of the PhoneGuide system were developed and evaluated that utilize a one-layer artificial neural network (NN) for vision-based classification on mobile devices. Global color features were extracted from three captured images of each exhibit. Bluetooth emitters that were distributed in a museum allowed a rough localization of visitors to limit the number of objects to be recognized—and consequently to increase the classification rate. Depending on the particular Bluetooth-cell in which a visitor was located, an individually adapted one-layer NN was trained in real time directly on the mobile phone.

One limitation of this approach is its scale-variance. As soon as visitors take pictures from different distances than trained, the recognition rate decreases significantly. Training the system for additional distant perspectives, or applying more sophisticated classifiers would lead to recognition improvements, but also to an unacceptable loss of performance required for an increased number of training passes on the mobile phone.

We describe two different classification approaches that ensure scale-invariant object recognition performed directly on low performance mobile phones: The first method configures and trains a one-layer NN on the mobile device in real-time. The second technique pre–trains a three-layer NN offline that is transferred to the phone for online recognition during runtime. For both methods, a client-server-architecture allows collecting data continuously, preprocessing it, and adapting it to the user behavior over time. Both systems are trained by capturing videos

E. Bruns · O. Bimber (✉)
Bauhaus-University Weimar, Bauhausstrasse 11,
99423 Weimar, Germany
e-mail: bimber@uni-weimar.de

E. Bruns
e-mail: bruns@uni-weimar.de

that show exhibits from different perspectives and distances.

The advantage of local object recognition that is performed directly on the mobile device compared with object recognition on a remote server [3, 4] is it's scalability: No communication between local devices and server is required. Instead of processing the classification requests of multiple users on one server sequentially (which possibly leads to long response times), classification is decentralized to the local devices. The disadvantage is the lower recognition rate of simplified classification techniques that achieve an acceptable performance on mobile phones. To overcome this, multiple inter-playing steps narrow the set of possibilities successively when identifying an object (cf. Fig. 2): First, location tracking of the phones limits the classification set to objects within the proximity of the visitor. The tracking information is used in addition for displaying the visitors' current location on a map. Second, a vision-based object recognition identifies the selected object from the remaining set, and arranges the recognition results visually relative to the NNs' excitations. From this, the user can select manually (cf. Fig. 1). Third, the classification results and optional meta-data (e.g., time, location, etc.) are recorded during runtime to guide a temporal system adaptation with the goal of increasing the robustness of the object recognition over time.

It is to emphasize that during run-time no data is transmitted to the server: the entire object recognition process is performed directly on the mobile phone.

For experienced users we achieve an average recognition rate of 92.6% for 139 museum objects. For the same objects, we achieve an average recognition rate of 82% under realistic conditions with 15 unexperienced museum visitors during a formal field experiment.

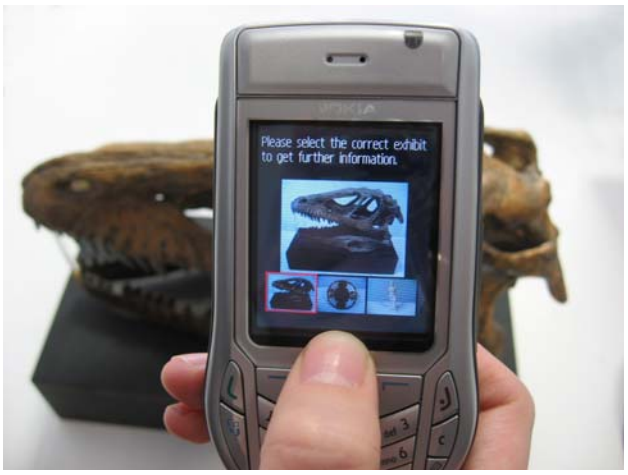
Please select the correct exhibit to get further information.

Fig. 1 Identifying a museum object with PhoneGuide: the set of potential objects is successively narrowed through device localization, local object recognition, and manual user selection

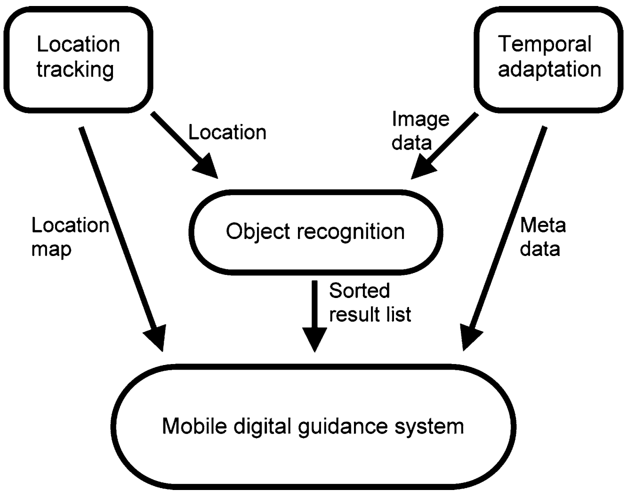

Fig. 2 System overview: location tracking provides an initial selection of regionally located objects and provides a location map to the user. Individual exhibits within the same region are identified through vision-based object recognition. Classification results and optional meta-data (such as time stamps, location, etc.) are recorded and used for a temporal adaptation of the system

The remainder of this article is organized as follows: chapter 2 discusses the related work. While chapter 3 provides details on the static and the dynamic training algorithms, chapter 4 gives an overview of the pervasive tracking mechanism, and describes the temporal adaptation techniques and the user interface. In chapter 5 the results of our evaluation and field experiment are presented. Chapter 6 concludes this work and indicates possible future improvements and extensions.

2 Related work

In this section, content-based video retrieval systems (CBVR) that use similar keyframe extraction techniques are introduced first. Next, adaptive machine learning systems are presented that are comparable to our temporal adaptation approach. Finally, digital guidance systems that address the same application space as ours are discussed.

2.1 Keyframe extraction for CBVR

Preprocessing video data for effective content representation is one of the major research fields in content-based video retrieval. In CBVR digital video material is processed to offer video query functionality based on a frame-, shot-, scene- or video-level [5]. On frame-level every image is examined to answer query requests. For the remainder, representative frames (called keyframes) are extracted for indexing.

In [6], for instance, the aim is to find videos with equal content although possibly different sizes and resolutions. In this case, every frame of each video is important for comparison. In [7], as another example, every frame is transformed to symbols based on computed features. By doing this, videos become comparable on a character-basis. Consequently, string matching algorithms can be applied.

Keyframe extraction techniques allow reducing the amount of data significantly. Thus, query requests can be processed faster. It is mainly applied at shot-level where shot-boundaries like cuts, dissolves or wipes are detected [5]. Frames located before or after these boundaries are marked as keyframes. Only these keyframes are used to process queries. For finding them, different kinds of image features (mainly global features based on color, contrast or spatial frequencies) are extracted. Pickering et al. [8], compute histogram features for each frame. The features of every frame are used to compute the Manhattan–distance between the current frame and the 16 consecutive frames. If the distance is below a pre-defined threshold, a shot boundary is detected. For cuts, the first frame of the boundary is marked. This represents a single shot. For transitions, each tenth frame is defined as a keyframe to avoid that it is part of the transition itself. After this, new features are computed for each detected keyframe. Convolution filters and different kinds of color histograms (RGB, HMMD, etc.) are utilized for this. Three different methods are evaluated for classification: First, the vector space model is applied by computing the Manhattan–distance between each keyframe within the database and the current query image. The result is the video that contains a keyframe with the smallest distance to the query image. The second method uses the AdaBoost-algorithm to classify query images. The third method is a variant of the distance-weighted k-nearest neighbors approach. Note, that this CBVR example is representative for similar approaches [9–11]. They only vary in their individual feature sets and applied classifiers.

In recent years, MPEG encoding is used for keyframe extraction. Therefore motion vectors that are computed in MPEG videos are examined to find keyframes [12, 13]. If a frame (predictive or bidirectional frame) is expected to contain backward predicted blocks but does not contain any, the content of consecutive frames must have changed radically. This indicates a shot boundary.

As in [8], we apply a distance function for extracting keyframes from a video. However, instead of examining only a short interval of consecutive images, we evaluate the whole video to ensure that no keyframe is represented twice. Keyframe extraction methods based on MPEG-encoded videos is not suitable in our case, since they do not consider the content of the video itself but only motion vectors. These, however, carry no essential information for object recognition because they are influenced by camera movements.

2.2 Adaptive learning

Many adaptive machine learning approaches require data collected from users for achieving continuous improvements. Relevance feedback methods used in information retrieval systems represent one class of techniques for adaptive learning: Users evaluate the results after a query is processed by indicating the relevance for each result item relative to the query item. After this, the system adapts to these user inputs and presents a new query result. An overview of relevance feedback approaches can be found in [14]. For example, MacArthur et al. [15] uses a decision tree for an image retrieval application that is adapted, depending on the users' feedback. It is based on weighting features (computed from the database images) differently in relation to the query. The color of a car, for example, is unimportant when searching for a particular brand. Therefore, multiple result images are presented to the user. These are separated into the categories of relevant and irrelevant images with respect to the current query. The feedback is used to build a decision tree that is then applied for dividing the remaining images into both categories. Finally, K relevant images with the smallest distance to the query image are presented to the user, and are evaluated again. This is repeated several times until the correct image is found or the query is canceled.

Comparable to [15], we perform a query by taking a photo of an object and compute a sorted list of possible results. The correct object is finally selected by users to retrieve information about it. The corresponding image is stored and applied for adapting and retraining the system offline. In contrast to [15], where queries are independent from each other, our temporal adaptation approach is sustainable: Previous requests influence future queries since the query data is used to improve the classifiers after each request. Draper et al. [16], introduces an adaptive object recognition system called ADORE that dynamically selects different vision procedures to accomplish object recognition tasks. These tasks are modeled as a Markov decision process in which rules are mapping perceptual states onto actions to approach object recognition as a supervised learning task. In contrast to our system, [16] is not able to collect data over time for continuous re-configuration depending on the user behavior. On the other hand, our system applies only one static set of features.

2.3 Mobile digital guidance systems

Mobile digital guidance systems can be separated into two main categories: location-based and image-based approaches. Location-based systems usually acquire and display position information by equipping the mobile device with a receiver for Infrared [17], wireless LAN [18] or GPS [17, 19]. The interested reader is referred to an overview by Baus et al. [20]. Luley et al. [3] combine location- and image–based techniques and present a digital city guide for mobile phones that allows to identify buildings in a city by taking photographs. Captured images as well as additional GPS-information are sent to a remote server using GPRS or UMTS. On the server's side, images are compared with a known data set using the SIFT–algorithm [21], which extracts local image features. The GPS information is used to narrow the data set in advance. Information about the buildings is retrieved from a second database and is sent back to the mobile device. Hare et al. [4] introduce a digital museum guide based on a Pocket-PC. As in [3], the captured image is sent to a remote server for recognition. The SIFT-algorithm is used as well for feature extraction but the classification is based on text retrieval techniques. Two hundred images were captured for evaluation. The database consisted of 850 paintings. The recognition rate was specified with 80% for queries with the match being under the top 20 positions. The probability drops to 20% if only the top position is considered.

Bay et al. [22] introduce a mobile museum guide based on a tablet-PC. The recognition is in this case performed on the device thus no data needs to be transferred to a remote server. An enhancement of SIFT, called SURF [22], is applied for object recognition. It was reported that 205 images of 20 exhibits were captured from different perspectives to ensure scale-invariance. For evaluation, 116 pictures were taken and a maximum recognition rate of 91.5% was achieved. In combination with their previous work [23] where Bluetooth-emitters are used to determine the visitors' position, this work comes close to our approach. However, instead of using high performance tablet-PCs, our system supports similar classification rates on ordinary, low performance mobile phones. This holds potential advantages for both—museum visitors and museum operators: since the visitors can use their own phones, the acquisition and maintenance costs required for handed out devices will be reduced or even eliminated. In addition, PhoneGuide temporally adapts to user inputs and consequently improves over time while [22] remains static.

Takacs et al. [24] apply SURF in combination with a nearest neighbor matching strategy for on-device object recognition in outdoor environments. In comparison to this approach, our system is faster (approximately by factor 5) and our classification run-time does not decrease with an increasing number of sample images per object.

In [3], as well as in [4], the classification is performed on a remote server and is therefore contrary to our approach in which the recognition is executed directly on the mobile phone. The advantages of a decentralized object recognition has been discussed in Sect. 1.

3 Offline data reduction, clustering and abstraction

As mentioned earlier, the main limitation of the object recognition algorithm in [2] is its scale-variance. It is due to the low amount of collected data for each object and the performance restrictions of today's mobile phones. Therefore, two pre-processing techniques were followed to support a fast classification of exhibits independently of the users' position.

Our *dynamic training* approach divides the image-based representation of an object into several separated views. For this purpose, a video of every object is taken (cf. Fig. 3). A server extracts keyframes from these videos based on computed global features (arranged as feature vectors) and a distance function. These keyframes are clustered depending on their similarity. For each resulting cluster one perceptron is trained. In this way, multiple quickly and ad hoc trained perceptrons are applied to classify one object. A data abstraction technique reduces the number of feature vectors without a loss of classification performance by merging multiple feature vectors. The perceptrons of multiple objects are assembled to form a NN in real time on the mobile phone, based on the spatial location of the user (defined by indoor location tracking using RF signal cells, see Sect. 4.1).

Our *static training* approach applies the extracted keyframes to generate three-layer NNs for all possible location cells resulting from user tracking. It is not necessary to cluster or reduce the data since the NNs are pre-trained on a remote server. Clustering, however, is still performed to eliminate outliers before training. After training is succeeded, the networks' weights are transferred and stored on the mobile device in order to select the correct classifier depending on the users' current location. In the worst case, for $2^N - 1$ NNs have to be created for N discrete location cells. However, in practice the number of location cells is much smaller, since the signal emitters are widespread in the museum. Consequently, no more than three emitter signals are superimposed simultaneously. This reduces the number of location cells significantly.

In the following sections, the necessary preprocessing steps (*keyframe extraction*, *keyframe clustering* and *data abstraction*) are explained in detail for both techniques. First, however, the applied global features are introduced.

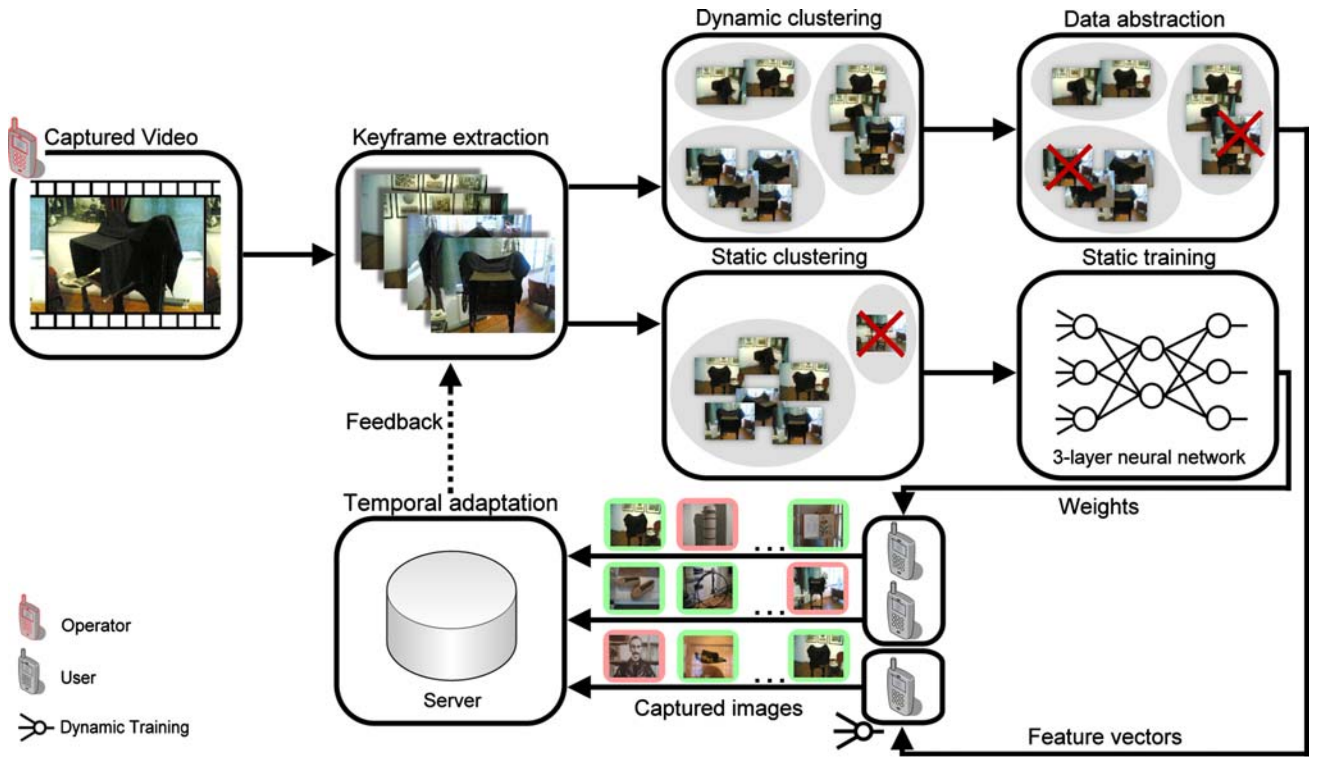

Fig. 3 Flow-chart displaying the preprocessing steps for the dynamic and the static training in combination with temporal adaptation: after capturing the exhibits, keyframe extraction, keyframe clustering and data abstraction is performed. Finally, the classifiers are generated on the server (static training) or on the mobile phone (dynamic training)

3.1 Global features

In contrast to today's object recognition approaches that widely apply local features, we use global features for describing images since they are less expensive to compute on low performance mobile devices. We apply 40 features composed of mean and variance values of each color channel as well as three 10-bin histograms. In [2], we have used only four histogram features (maximum peak in intensity of each color channel and gray channel) and 14 features in total. It is obvious that the more histogram bins are used the better is the description but the slower is the training performance for a dynamic training. To find the best trade-off between the number of bins and the training duration, we have trained and recognized 15 similar sample objects with different numbers of histogram bins. The average training time on a mobile phone for each constellation was measured, and is plotted in Fig. 4. Ten bins for each color channel (3 × 10 for RGB color + 10 for mean and variance) seems to be an appropriate compromise between recognition rate and speed. To enable an adequate evaluation (see Sect. 5), we use the same number of histogram bins for the static training as well. An investigation, and an evaluation of global feature sets can be found in [1]. This global feature set is not computed for the entire image. Instead, each frame of a captured video is divided into 12 equal image patches as explained in [2]. Consequently, one feature vector for each patch is computed.

3.2 Keyframe extraction

The keyframe extraction is applied for distributing image data among each perspective equally to ensure that every

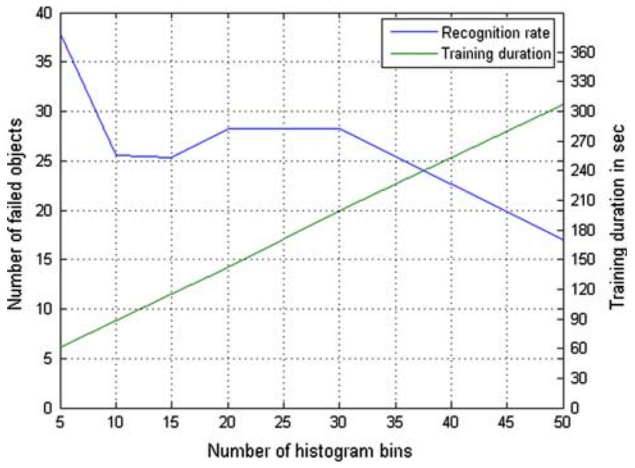

Fig. 4 Training duration versus recognition rate for different bin sizes

representative perspective is covered and weighted similarly during training. Thereby, redundant information is filtered out. For the dynamic training, the extraction of keyframes is also important for reducing the amount of image data, since more data obviously leads to longer online training durations on the mobile device. As in [8], we use a distance function (Eq. 1) to determine keyframes:

$$d = \max_{i=0}^{N-1}\left(\sum_{j=0}^{M-1} s_{i,j} \cdot f_j\right) \tag{1}$$

where N is the number of all keyframes and M is the dimension of the feature vector. Keyframes are identified by computing the product between the normalized feature vector f of a new frame with the normalized feature vector s of all already identified keyframes. If this product is below a predefined threshold t_1, the new frame is strongly different from the existing keyframes, and is consequently identified as a new keyframe. Initially, the first frame of each video is marked as a keyframe.

We apply a threshold close to 1 for the static training to sort out frames that are almost identical. For the dynamic training, the threshold is estimated empirically. However, a subsequent data abstraction technique (see Sect. 3.4) will automatically adjust the number of required keyframes in addition.

3.3 Keyframe clustering

After extraction, the keyframes are clustered into groups of similar ones. For the dynamic training, this is done to reduce the complexity of the object data. In case of the static training the clustering is applied to eliminate frames that are likely to not belong to the corresponding object.

As explained earlier, the dynamic training creates and trains multiple perceptrons for a single object to achieve a scale and perspective invariance. The extracted and clustered keyframes are used for this. We want to associate the term *virtual objects* with each of these clusters. Our technique is outlined in the following algorithm:

1. Cluster all remaining keyframes of one object into v (initially $v = 1$) virtual objects.
2. Train v perceptrons with the assigned keyframes.
3. Recognize each keyframe: if the excitation of a keyframe at its corresponding perceptron is below a predefined threshold t_2, the whole cluster (and all its keyframes) is rejected. If the excitation of all keyframes at their corresponding perceptron are above t_2, the cluster and its keyframes are accepted and stored.
4. Go to step 1 with $v = v + 1$ and with all rejected (remaining) keyframes until all keyframes are accepted.
5. Repeat these steps m times and return the constellation with the smallest number of clusters.

First, the extracted keyframes of one object are clustered into two sets using the k-means algorithm. For each set, one perceptron is trained by separating the keyframe data into a training and a validation set. If each frame of one set has a perceptron output higher than the empirically estimated threshold t_2 (equal for all objects), this set is stored. Otherwise, all frames of the set are rejected. This algorithm is performed multiple times since the k-means algorithm can deliver different cluster constellations for the same input data. Since the separation capabilities, and consequently the recognition rate, degrade with a larger number of (virtual) objects, the optimal solution is the one with the smallest number of clusters.

If only a single perceptron for one object is trained (cf. Fig. 5), the mean as well as the minimum of the perceptron's excitation is much lower than if multiple perceptrons are trained (cf. Fig. 6). Thus, this approach leads to higher recognition rates. The number of required perceptrons is automatically derived and depends on the complexity of the object.

For the static training, there is no need to divide real objects into multiple virtual objects since the classifiers are trained offline on a server. However, a clustering is still performed to eliminate frames that are captured by users but that are likely to not contain an exhibit. As mentioned earlier, the captured images are stored on the mobile device for adapting and improving the system continuously (see Sect. 4.2). But this can only be successful if the images are correlated to the correct objects. It might happen that users take photos of something completely different (e.g. of the floor or a wall) and associates this with an exhibit. These

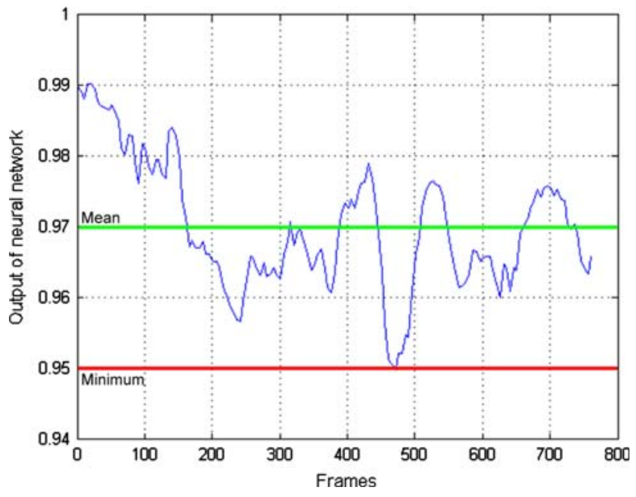

Fig. 5 Identification rate: excitation output of a one-layer neural network (perceptron model) if one object is represented by one perceptron that is trained through ten training passes

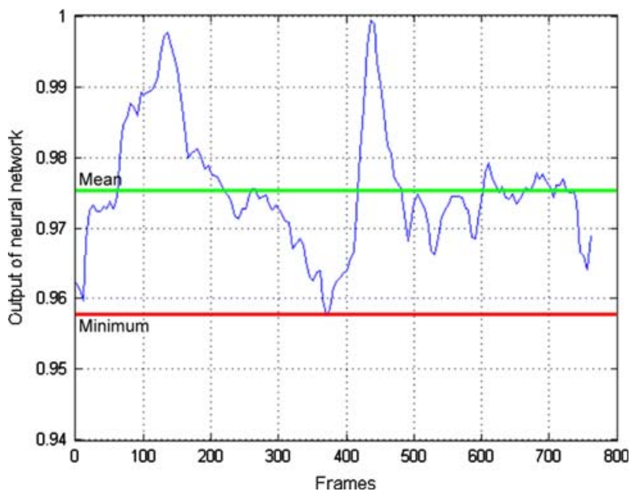

Fig. 6 Improved identification rate for the same object as in Fig. 5, represented with four perceptrons (also trained through ten passes)

images would influence the training and recognition performance negatively. Therefore the following algorithm is used for identifying these outliers through clustering:

1. Cluster all keyframes of one object into two sets.
2. If the absolute ratio between the number of elements of the smaller set to the number of elements of the larger set is below a threshold t_3, then the smaller set contains outliers and is deleted.
3. Execute step 1 and 2 until the ratio is equal or larger than t_3.

This algorithm is based on the assumption that outlier images differ significantly to already captured object images. However, it might happen that correct object images are deleted. Yet, they represent only a small minority and are therefore being ignored, as long as the algorithm is executed in appropriate time intervals to ensure that different but correct object images can accumulate. This ensures that such sets are large enough and are not deleted.

3.4 Data abstraction

To reduce the number of keyframes for each cluster, a data abstraction step is applied in addition. In case of the dynamic training a compromise between the amount of feature vectors and a sufficient object representation has to be found. Therefore, the following algorithm is used to further reduce the number of keyframes while keeping the recognition rate constant (this is repeated for all clusters):

1. Initially train a perceptron with all N_c keyframes of a cluster (virtual object). Determine the initial mean output excitation O at the perceptron over all N_c keyframes. The number of merged keyframes is $i = 1$ (cf. Fig. 7, step 1).
2. Cluster the keyframes of this virtual object in $N_c - i$ subsets.
3. Average these $N_c - i$ subsets. Each subset represents one feature vector (cf. Fig. 7, step 2).
4. Train a perceptron with the $N_c - i$ feature vectors.
5. Determine the current mean output excitation A of the trained perceptron over all N_c keyframes.
6. If $A < O$ or $i = N_c - 1$, exit and use the last or current cluster configuration respectively as result. Else go to step 2 with $i = i + 1$ (cf. Figs. 7, step 3–5).
7. Execute steps 1–6 m times and choose the result with the best abstraction rate (1 - (#keyframes after abstraction/#keyframes before abstraction) (cf. Fig. 7, step 7).

This algorithm is based on the assumption that the feature vectors of similar keyframes can be averaged without a loss of necessary information. To ensure that the quality of the representation is constant, we evaluate the mean excitation output of the perceptron as indicator. To allow a higher compression, O can be decreased. Executing the algorithm $m = 25$ times has been proven to be sufficient in our experiments.

4 Online localization, adaptation and employment

The computing power of today's mobile phones is still far too low to carry out sophisticated object recognition algorithms. Simpler techniques that are based on global features that can be computed quickly (such as ours), are—by themselves—not reliable enough to distinguish between a realistically large number of objects. Therefore, we have extended our system with additional features (cf. Fig. 2) like user tracking and temporal adaptation that enhances the recognition rate and the scalability.

This chapter summarizes our pervasive tracking technique that was introduced in [2]. We explain a temporal adaptation technique next, that adjusts to the user behavior over time. Finally, we describe the user interface of the guidance system that was implemented on the phone.

4.1 Localization

In [2], we introduced an indoor location tracking system for mobile phones that determines roughly the location of users (i.e. their phones). For this, a small number of low cost RF-emitters (RF-emitters like RFID or wireless LAN are possible if supported by the phone—yet we used Bluetooth) were distributed in a museum. Single and multiple superimposed signals partition the environment into multiple location cells depending on signal interferences and reflections. Each cell is defined by one or more

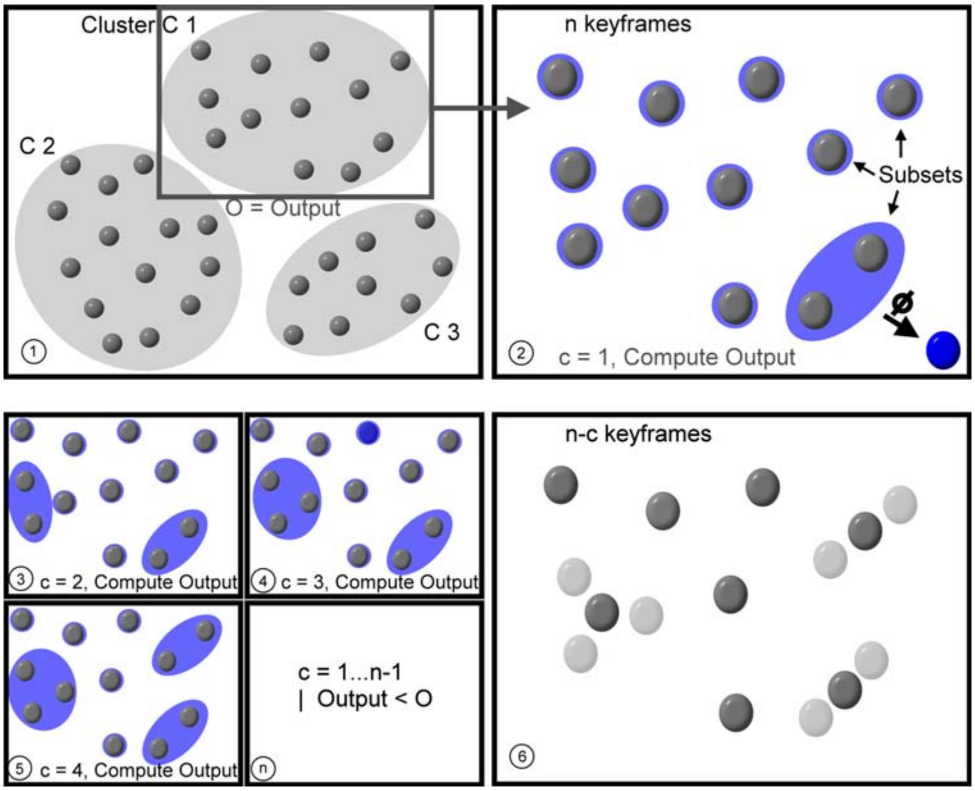

Fig. 7 Data abstraction example: step 1: compute mean output excitation *O* of a cluster. Step 2: average two feature vectors of two similar keyframes and compute output again. Steps 3–5: increase the number of maximally merged keyframes until an exit condition is reached (step 6). Step 7 illustrates the final result

emitter-IDs (e.g. a Bluetooth-ID in our case). The mobile guidance system scans continuously for new emitter-IDs and derives the current location cell from them. Based on a lookup-table that contains the correlations between location cells and objects within the cells, a new NN is trained during run-time that is optimized to recognize objects within the users' proximity only.

4.2 Temporal adaptation

Temporal adaptation is a technique that collects a variety of information during run-time. This information is interpreted offline and applied to adjust the system's performance or to provide additional feedback to users. For instance, if an exhibit has been recognized, recommendations can be offered for which objects are usually visited next. Time stamps and durations can be recorded while recognizing objects to adapt and optimize the computed features. If, for instance, the recognition fails only at a particular time of the day, the number of histogram bins can be reduced during this time to increase the robustness against lighting changes in future. This is only applied to the objects within the corresponding location cell.

In practice, we have developed an approach to continuously collect image data while visitors are using the system. Therefore, we store the images' feature vectors when selecting the correct object from the predicted sorted result list (SRL) (see Sect. 4.3.1). These vectors are transmitted to a server when visitors are leaving the museum. They are applied to retrain the three-layer NNs on the server by adding them to the existing feature database. The new NNs are transferred to the mobile phones of visitors entering the museum. This procedure can be seen as a sort of continuous user guided offline training and has two advantages: First, providing more valid input samples for training a NN leads to a more robust separation between the objects' feature vectors and consequently to higher recognition rates. Second, the NNs are optimized over time to recognize objects from certain, common perspectives and distances. Initially, objects are operator-trained by capturing videos from several positions where users are expected to be standing when taking photographs. This is only a rough estimation and is based on the operator's experiences. Furthermore, it is hardly possible to capture an object from all perspectives. With the temporal adaptation, new perspectives are captured. While more common perspectives are implicitly up-weighted by the NNs, less common perspectives and outliers are down-weighted. Consequently, the NNs adapt to the visitors' behavior and converge at weights that are highly specialized for the individual objects.

The optimization through our temporal adaptation technique is only used in combination with the static training. Since the amount of feature vectors that have to be trained is continuously growing, the optimization is not suitable for the dynamic training. Note, that the size of the

NNs that are transmitted to, and finally used by the mobile phone does not increase. Neither does the recognition time.

4.3 User interface

This section introduces the SRL as well as the location map that are presented to the visitors during run-time. Furthermore, a video recognition mode is presented that allows users to capture a short video sequence instead of a single image to recognize an exhibit with a higher reliability.

4.3.1 Sorted result list

To ensure that the stored correspondences between the physical objects and the feature vectors are correct in order to apply the temporal adaptation, a graphical interface called *sorted result list* (SRL) was implemented. This enables users to browse and select the correct object from an image list while the related feature vectors are stored on the mobile phone. The items in the SRL are ordered based on their generated NN's maximum excitation that is proportional to their recognition probability. By ordering the items, the browsing time to identify the correct object manually is kept at a minimum. Thus, the better the object recognition, the smaller is the effort of the visitors to select the correct exhibit. In the best case, the object is recognized immediately and it requires only a simple click operation for selection since the first object is pre–selected by the system (cf. Fig. 1).

4.3.2 Location map

Since the scanning of RF-emitters introduces delays (e.g. for Bluetooth at least 12 s), the current user location is displayed on the phone. Only if the correct NNs are selected for the current location cell, objects can be successfully recognized. Based on the detected location cell, the room within the museum in which the visitor is located is highlighted and additional textual information is provided (cf. Fig. 8). This gives a visual indication to the users when a location transition has been detected and an object recognition becomes possible.

4.3.3 Video recognition

Beside an object recognition from one photograph only, we have developed a video recognition that evaluates multiple images to increase the recognition rate. This enables the system to recognize an exhibit more precisely since more information about the object is collected. Thus, in the video recognition mode, the system executes multiple recognition passes for different video frames. The more frames are processed, the more reliable is the result—but the longer is the duration the users have to wait until it is displayed. The following algorithm represents an acceptable trade-off between these two factors:

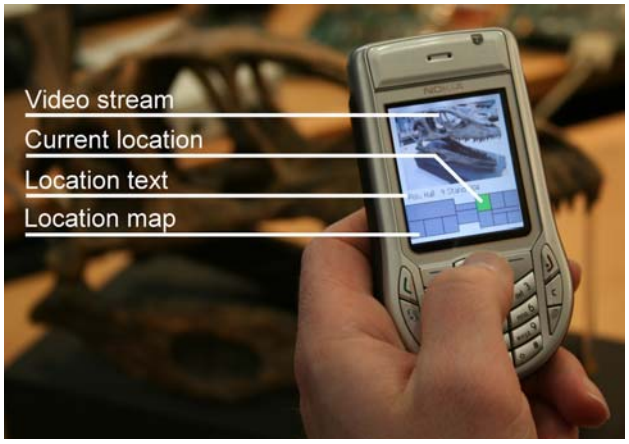

Fig. 8 Object recognition with PhoneGuide: Elements of the user interface are annotated

1. Recognize the first video frame.
2. Test if the next frame is a keyframe and perform a recognition.
3. *After 3 processed keyframes:* if all frames are associated with the same object, present the result and exit. Else, go to step 2.
4. *After 4 processed keyframes:* if 3 out of 4 frames are associated with the same object, present the result and exit. Else, go to step 2.
5. *After 5 processed keyframes:* the algorithm is canceled. The most frequent result is displayed.

This algorithm ensures that only a minimum number of frames are processed for recognizing an object, while an upper bound is not exceeded. In practice, users have to move the mobile device during the video recognition mode to capture the object from different perspectives and distances (the images are not cached). To ensure that the processed frames are different from each other, keyframes have to be extracted. Therefore, the keyframe extraction approach that is described in Sect. 3.2 is carried out directly on the mobile phone during run-time. If turned on, the video recognition mode is visually indicated on the GUI.

5 Evaluation

Our system was tested and evaluated under realistic conditions in the City Museum of Weimar. We carried out experiments with one experienced user (a user who also trained the system) and with 15 unexperienced subjects. We have trained the system to recognize 139 exhibits that

Table 1 Preprocessing details for both techniques

Property	Dynamic training	Static training
Extr. keyframes	1,225	7,464
Clusters	254 (min: 8 max: 1)	–
Abstraction rate	14.1%	–
Frames/object	7.3 (min:1 max: 8)	51.8 (15, 80)
Memory	ca. 1 MB	ca. 0.35 MB
Duration preproc.	ca. 10 min	ca. 120 min

were distributed over two floors. Among them were small (e.g. coins or medals) and larger (e.g. statues or models) exhibits that were placed in front of mirrors, in showcases and next to windows. We asked each subject to fill out a questionnaire to investigate the usability and the acceptance of the system.

For training every object, we have captured individual videos of approx. Twenty-six seconds with about 4 fps in a resolution of 160 $\times$ 120 pixels. They were transferred to a PC (Intel Centrino 1,5 GHz, 512 MB RAM.) for preprocessing (see Sect. 3). We have distributed nine Bluetooth emitters in the museum that spanned 16 different location cells. The smallest cell contained two objects, the largest 46. To compare the dynamic and the static training, we applied both methods for recognizing each object from six different perspectives and recorded the resulting recognition rates. Furthermore, we evaluated our previous work [2] again under identical conditions to investigate the improvements.

In the following, we want to compare the characteristics and recognition performance of the static and the dynamic training first. Our temporal adaptation approach is evaluated next. Finally, we highlight the results of the field experiment.

5.1 Dynamic versus static training

For preprocessing, the server application[1] on the PC processed 14155 frames in the context of our experiments (details can be found in Table 1). The duration for training the NNs on our mobile phone (Nokia 6630, ARM-926 220 MHz, ca. 3.3 MB RAM, Symbian 8.0a) as part of the dynamic training was strongly dependent on the number of objects, clusters and keyframes. Roughly 10 s–3 min are needed to generate and configure the NNs based on the current location cell during run–time, using a maximum iteration number of ten passes.

It turns out that the amount of computed clusters for the objects can vary a lot. The main reasons for this are shadow casts or self-reflections on showcases of the operator: in these cases, multiple keyframes are extracted and therefore more clusters are generated.

Table 2 illustrates the recognition rates for both approaches achieved by an experienced user who used the system before.

In [2], we reported a recognition rate of over 95% without supporting scale-invariance, and by using 14 global features. This, however, was achieved by recognizing each object once and by keeping the distance to each object the same at all times. Thus, the images taken for recognition were similar to the three captured images that were used for training. Yet, as soon as visitors took a photo from a different perspective or scale, the recognition rate decreases significantly. We experimentally found a recognition rate of 52.6% when enforcing different scales (i.e. by taking photos from 6 different perspectives at two different distances) for our previous system. Our new techniques achieve 61.2% (static) and 70.3% (dynamic) under the same conditions (for 88 recognized objects). This shows that the new classifiers and the preprocessing do improve the recognition rate but they can not compensate for an insufficient object description. When using 40 features (see Sect. 3.1), and consequently provide a better object representation, we encounter a recognition rate of 92.6% for the static training. This is comparable to the recognition rate pointed out in [2]—but this time, the recognition is independent of the users' location and distance. With the dynamic training, we achieved a recognition rate of 85.6%. This indicates that simple linear classifiers in combination with an appropriate preprocessing (keyframe extraction, clustering and data abstraction) can be almost as successful as non-linear classifiers.

The video recognition mode (see Sect. 4.3.3) achieved a recognition rate of 95.7%.

Figure 9 presents the development of the recognition time on the mobile phone based on the number of objects that are currently trained. With an increasing number of objects, the recognition time of the dynamic training increases much faster than the recognition time that is required for the static training. On average, two perceptrons are added for each new object when using the dynamic training. Consequently, the more objects are trained the more perceptrons have to be iterated. In contrast, the static training adds—for each new object—exactly one new neuron to the output layer of the three-layer-NN.

Figure 10 displays the number of failed objects for different numbers of failed perspectives. It illustrates that it is more likely that objects are not recognized from only one single perspective rather than from multiple perspectives: only one object for the static approach and five for the dynamic approach were not recognized at all.

[1] Implemented in Matlab with Java interface for Bluetooth connections.

Table 2 Recognition rates of our previous system (without preprocessing), the new approaches with preprocessing (static and dynamic), and of the video recognition based on the static training

Number of features	Without preprocessing (%)	Dynamic training (%)	Static training (%)	Video recognition (%)
14 features[a]	52.6	61.2	70.3	–
40 features	–	85.6	92.6	95.7

[a] For 88 objects

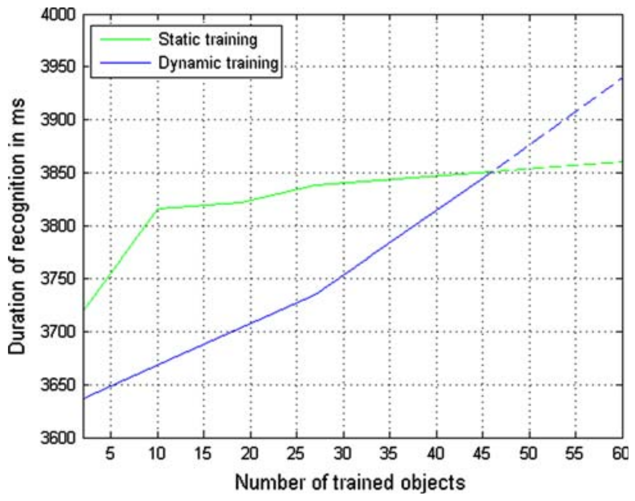

Fig. 9 Duration of recognition versus the number of trained objects for the static and the dynamic training

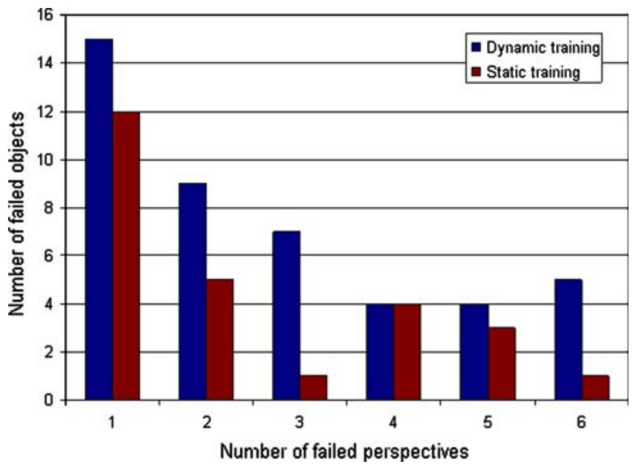

Fig. 10 Number of failed objects versus number of failed perspectives for the static and the dynamic training

5.2 Temporal adaptation

As pointed out in Sect. 4.2, the basic idea behind a temporal adaptation is to continuously collect data to retrain and improve the recognition rate of the NNs over time.

For investigating the efficiency of temporal adaptation, we performed the following experiment: ten objects were trained via static training. Afterwards, these objects were recognized from 6 different perspectives. As soon as the

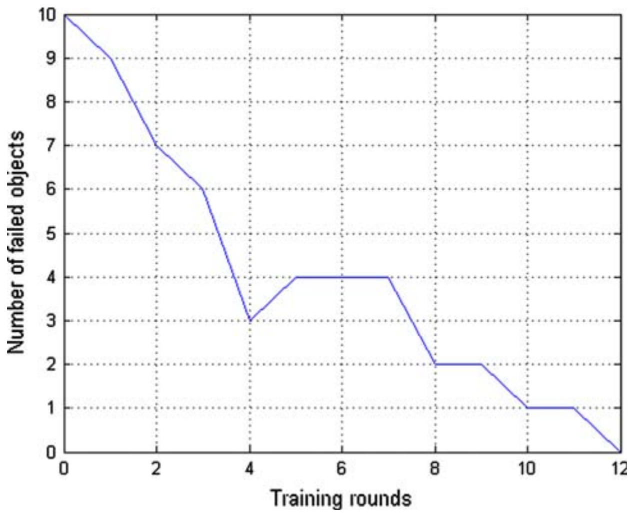

Fig. 11 Number of failed objects after each iteration of the temporal adaptation

recognition of one perspective failed, the next object was identified. Furthermore, the last image of every exhibit that has been captured was stored on the mobile phone. After all objects have been approached once, the stored data was transferred from the phone to the server and new classifiers were trained. With these new classifiers all objects were approached again. This was repeated until all objects have finally been recognized from all perspectives. The temporal improvement of the recognition rate throughout 12 training rounds is shown in Fig. 11. Occasionally, the system needed multiple training rounds until an improvement was achieved (e.g. round 5–7). In this case, the NNs required more temporal data to be influenced.

Figure 12 shows the temporal development of computed keyframes and clusters for the same experiment. As expected, the number of keyframes and clusters converge over time since the number of different perspectives and scales is limited. This validates our initial assumption that a temporal adaptation improves the recognition rate while the sizes of the required NNs converge (instead of keeping on increasing continuously).

5.3 Field experiment

While the section above summarizes experiments that were carried out by an experienced user, this section presents the

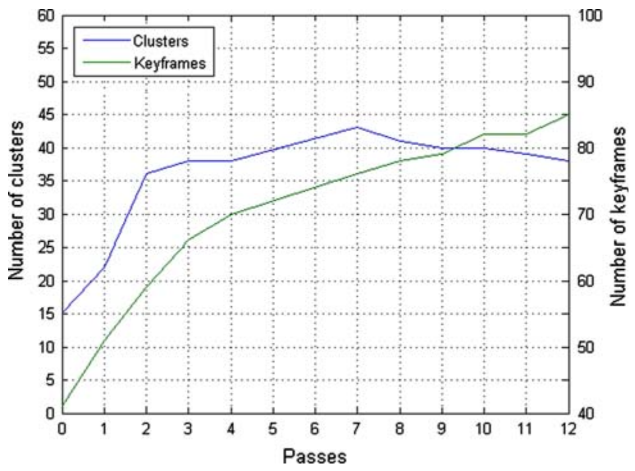

Fig. 12 Number of computed clusters and keyframes during temporal adaptation

results of a field experiment with unexperienced museum visitors. We want to discuss the user behavior during applying the system first. Next, the recognition rates that were achieved in this experiment are discussed. Finally, we evaluate the feedback provided in handed out questionnaires.

5.3.1 *User behavior*

A temporal adaptation can only converge if the number of perspectives and scales of an object is truly limited. Thus, we assume that most visitors will approach the same objects from similar perspectives and distances. To prove this, an experiment was carried out to identify the visitors' locations when they photograph exhibits for recognition. Therefore, two different objects (a globe and an oven, see Figs. 13 and 14) where chosen. They differ in their dimension as well as in the size of their surrounding area in which visitors were able to stand. Fifteen subjects[2] were asked to take a photo of both exhibits separately. They were not informed about our experiment by any means. The subjects' locations were registered and are plotted in Figs. 13 and 14. Although the users could move in a 90 degree radius around the globe (cf. Fig. 13), they all chose the same perspective. The distance, however, varied in a range of 70 cm. For the oven (cf. Fig. 14) the subjects could choose a position within an area of about 5 square meters. However, they all chose a position within a common area of approx. 1 square meter. This strengthens our assumption that different visitors will behave similar and are likely to choose only a small subset of all possible positions and scales. Thus, these common perspectives and scales will be up-weighted temporally over time, while less common outliers will be down-weighted. This behavior is essential for our temporal adaptation.

[2] Seven of these 15 subjects were female. Their average age was 27.4 years. The oldest person was 55, the youngest 22-year-old.

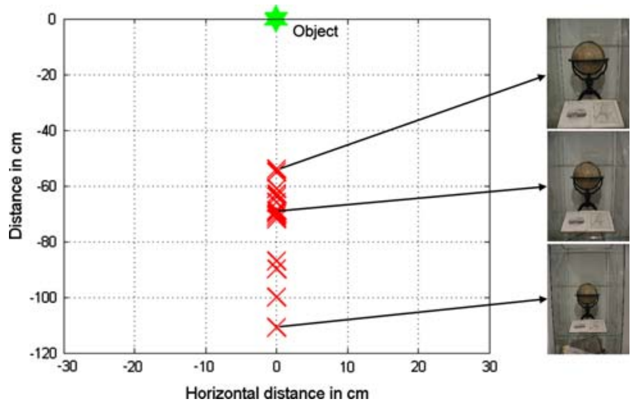

Fig. 13 Subjects' registered locations in front of the globe

5.3.2 *Recognition rate*

For evaluating our system under truly realistic conditions, we carried out an extended field experiment with the same 15 subjects, recognizing 139 different objects in the museum. We chose the same object set that was selected for the field experiment described in [2]. This allows comparing the results of both approaches. Our aim was to evaluate the handling and the acceptance of the system, as well as its recognition rate for completely unexperienced users. The static training was applied for this experiment since it achieved the highest recognition rate in our previous experiments.

Each subject was asked to recognize ten different objects with the photo recognition mode and five objects with the video recognition mode (see Sect. 4.3.3). In total, all of the 139 were approached at least once. The results of the experiment are displayed in Table 3.

Although the number of subjects is too low to be representative, it gives indications of a first trend with respect to the quality of handling, and it is sufficient to determine the recognition rate.

We achieved a recognition rate of 82% for unexperienced users. This is approx. 10% less than for the experienced user. One reason for this is that the museum visitors tried to recognize objects from perspectives that were initially not trained. However, this can be compensated by the temporal adaptation, and is confirmed by the fact that 40% of the failed objects were still ranked at positions 2 or 3 in the SRL (see Sect. 4.3.1). By gathering more data over time, the classifiers are specialized more efficiently. Another reason for the lower recognition rate of the unexperienced users was a wrong usage of the system. Sometimes subjects did not capture an object entirely. This was only observed for one (older) person that experienced difficulties while using the mobile phone in general.

The video recognition rate of 77.1% was below our expectations. We have two explanations for this: First, the

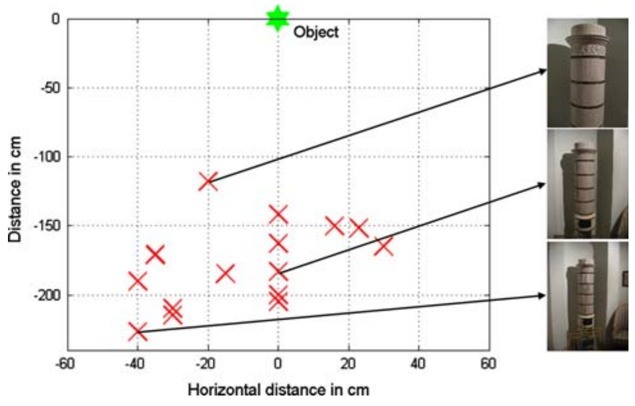

Fig. 14 Subjects' registered locations in front of the oven

Table 3 Recognition rates for experienced and unexperienced users

Recognition mode	Experienced (%)	Unexperienced (%)
Photo recognition	92.6	82.0
Video recognition	95.7	77.1[a]

[a] For $15 \times 5 = 75$ objects

subjects captured perspectives that were not trained. Again, this can be compensated temporally over time. Second, while capturing a video, the application executes multiple processing tasks. This causes short delays in the video stream. Therefore, subjects had difficulties in centering the objects all the time. This could be overcome by faster devices. In general, the video recognition took approx. 12–21 s.

5.3.3 *Acceptance*

The subjects were asked to fill out questionnaires to provide answers on two central questions: how well is the handling of the system components? How high is the acceptance of using such a system in practice? Overall, 20 different questions had to be answered by each subject. Answers could be provided through a ranking between 1 (worst) and 7 (best).

In the following, we want to summarize the results: The handling as well as the user interface was easy to understand and to learn. Almost all subjects were satisfied in this point. The recognition quality was rated on average with 5.8 (out of 7). The most critical point was judged to be the location tracking via Bluetooth. Varying signal ranges due to interferences and reflections sometimes forced the subjects to wait up to one minute until they could proceed with the correct location being determined. Loading the images for the SRL required max. 8 s. after entering a new location cell. This delay was also criticized by the subjects. It seemed that a non-perfect recognition is more tolerated by users than long waiting times. This is also confirmed in [25], in which a location–based as well as an image–based tour guidance system was evaluated. Although the error rate of the image-based system was much higher than the error-rate of the location-based system, the subjects did not favour one of the approaches over the other.

Thirteen (out of 15) subjects favored the image-based recognition over the video recognition mode, although most subjects confirmed that the handling of the video recognition is sufficient. Two subjects explicitly stated that the latency for recognizing an object (Nokia 6630: feature calculation: 2.4 s; classification: 0.2 s; capturing and loading user interface: 1.2 s) is too long. However, on newer phones (e.g. Nokia N95) the feature calculation and classification take 0.5s on average.

Furthermore, most subjects thought (average ranking: 5.73 out of 7) that this kind of museum guide is an adequate alternative for today's audio-guides.

6 Conclusion and future work

In this article we have presented techniques and algorithms that enable scale-invariant and adaptive object recognition.

A client–server architecture was developed to continuously collect and preprocess object data for improving the internal representation of exhibits. Two general approaches were introduced for training different classifiers from this data: a static training technique trains three-layer NNs on a remote server and transfers the final weight sets to the mobile phones for recognition. A dynamic training technique achieve similar recognition rates with weak classifiers (one-layer NNs) by configuring and training them during run-time on the mobile phone after clustering the collected object data on a server.

Under equal conditions, we have achieved a maximum recognition improvement of 40% compared to our previous, scale–variant algorithm [2]. Using a temporal adaptation, our system is able to collect object data continuously and to increase its classification rate over time.

A field experiment in a museum has shown that our prototype can, in principle, pass a practical acceptance test.

Although the static training achieved a higher recognition rate (92.6%) and is more applicable since the NNs have not to be configured and re-trained on the mobile phone (which would require additional time), the dynamic training (recognition rate: 85.6%) is still an appropriate solution for selected tasks: If, for instance, objects have very diverse shapes and colors from different perspectives the dynamic training can compensate for this much better than the static training since every exhibit is separated into multiple virtual objects. Consequently, each view can be trained individually. Furthermore, the dynamic technique trains NNs during run-time. This offers the opportunity to apply the system in environments where many RF-emitters change their position dynamically (e.g. in a bookstore where all books are tagged with RFID-chips). Comparable to the museum scenario, the mobile phone trains a NN online, based on the received RF-signals. The static training would not be applicable in this case, since the number of different NNs to be trained in advance would be too large since they have to be transferred to the mobile phone. Furthermore, the NNs would have to be adapted manually for every modification in the objects' arrangements. Our field experiment has demonstrated that a location tracking with Bluetooth-emitters is not satisfying. Because of dynamic interferences, the signal range can vary much and the location tracking is not reliable. This can lead to lower recognition rates. Alternative tracking techniques (such as via wireless LAN or RFID) have to be investigated in future. Yet, they have to be supported by off-the-shelf devices. In some cases, only transitions of visitors from one room to another have to be detected. Short range emitters attached at door frames would allow detecting these transitions.

Although the temporal adaptation can also compensate small changes in daytime illumination, our system is still light-variant. The better color features describe an image, the more sensitive the recognition becomes with respect to illumination changes. Therefore, new image features have to be investigated to achieve a full lighting invariant state. An additional adaptive feature selection (as part of the temporal adaptation) can determine the development and effectiveness of particular features over time. Based on this development, optimal feature sets can be chosen from a global pool of features for each individual location cell. The behavior at the temporal adaptation over a long application period has to be investigated and evaluated.

Acknowledgments We thank the Senckenberg Museum of Natural History Frankfurt, the Museum for Pre- and Early History Weimar, the Museum of the City of Weimar, and CellIQ for their support. The PhoneGuide project is supported by the Stiftung für Technologie, Innovation und Forschung Thüringen (STIFT). Further information is available at http://www.uni-weimar.de/medien/AR.

References

1. Föckler P, Zeidler T, Brombach B, Bruns E, Bimber O (2005) Phoneguide: museum guidance supported by on-device object recognition on mobile phones. International conference on mobile and ubiquitous computing
2. Bruns E, Brombach B, Zeidler T, Bimber O (2007) Enabling mobile phones to support large-scale museum guidance. IEEE multimedia 14(2):16–25
3. Luley PM, Paletta L, Almer A (2005) Visual object detection from mobile phone imagery for context awareness. Proceedings of the 7th international conference on Human computer interaction with mobile devices & services, pp 385–386
4. Hare JS, Lewis PH (2004) Content-based image retrieval using a mobile device as a novel interface. Proceedings of the first international workshop on mobile vision, pp 64–75
5. Marchand-Maillet S (2000) Content-based video retrieval: an overview. http://viper.unige.ch/marchand/CBVR/
6. Zobel J, Hoad T (2006) Detection of video sequences using compact signatures. ACM Trans Inf Syst 24(1):1–50
7. Adjeroh DA, Lee MC, King I (1999) A distance measure for video sequences. Comput Vis Image Underst 75(1–2):25–45
8. Pickering MJ, Rüger S (2003) Evaluation of key frame-based retrieval techniques for video. CComput Vis Image Underst 92(2–3):217–235
9. Feng H, Fang W, Liu S, Fang Y (2005) A new general framework for shot boundary detection and key-frame extraction. Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval, pp 121–126
10. Miene A, Hermes Th, Ioannidis G, Fathi R, Herzog O (2002) Automatic shot boundary detection and classification of indoor and outdoor scenes. Text REtrieval Conference (TREC)
11. O'Toole C, Smeaton A, Murphy N, Marlow S (1999) Evaluation of automatic shot boundary detection on a large video test suite. The challenge of image retrieval (CIR99)—2nd UK conference on image retrieval
12. Browne P, Smeaton A, Murphy N, N O'Connor, Marlow S, Berrut C (1999) Evaluating and combining digital video shot boundary detection algorithms. In: Proceedings of the fourth Irish machine vision and information processing conference
13. Nguyen DT, Gillespie W (2003) A video retrieval system based on compressed data from mpeg files. In Conference on convergent technologies for Asia–Pacific region, vol 2, pp 555– 560
14. Zhou XS, Huang TS (2003) Relevance feedback in image retrieval: a comprehensive review. Multimed Syst 8:536–544
15. MacArthur S, Brodley C, Shyu C (2000) Relevance feedback decision trees in content-based image retrieval. IEEE workshop on content-based access of image and video libraries, pp 68–72
16. Draper BA, Bins J, Baek K (1999) ADORE: adaptive object recognition, pp 522–537
17. Abowd GD, Atkeson CG, Hong J, Long S, Kooper R, Pinkerton M (1997) Cyberguide: a mobile context-aware tour guide. Wirel Netw 3(5):421–433
18. Cheverst K, Davies N, Mitchell K, Friday A, Christos Efstratiou (2000) Developing a context-aware electronic tourist guide: some issues and experiences. CHI, pp 17–24
19. Pospischil G, Umlauft M, Michlmayr E (2002) Designing lol@, a mobile tourist guide for umts. Mobile HCI, pp 140–154
20. Baus J, Cheverst K, Kray C (2005) A survey of map-based mobile guides map-based mobile services—theories, methods and implementations, chap. 13. Meng/Zipf, Springer
21. Lowe DG (1999) Object recognition from local scale-invariant features, pp 1150
22. Bay H, Fasel B, Van Gool L (2006) Interactive museum guide: fast and robust recognition of museum objects. Proceedings of the first international workshop on mobile vision
23. Bay H, Fasel B, Van Gool L (2005) Interactive museum guide. The seventh international conference on ubiquitous computing UBICOMP
24. Takacs G, Chandrasekhar V, Gelfand N, Xiong Y, Chen W-C, Bismpigiannis T, Grzeszczuk R, Pulli K, Girod B (2008) Outdoors augmented reality on mobile phone using loxel-based visual feature organization. To appear in: IEEE Transactions on pattern analysis and machine intelligence
25. Davies N, Cheverst K, Dix A, Hesse A (2005) Understanding the role of image recognition in mobile tour guides. Proceedings of the 7th international conference on Human computer interaction with mobile devices & services, pp 191–198