

Phone-to-Phone Communication for Adaptive Image Classification

SUPPLEMENTARY MATERIAL

Application of museum guidance system with one-time adaptation (at museum entrance), pervasive tracking and local image recognition. Without ad-hoc phone-to-phone communication:

<http://www.megaupload.com/?d=22SOHGZT>

Simulation of a large number of visitors with phone-to-phone communication:

<http://www.megaupload.com/?d=O4PKJA5N>

ABSTRACT

In this paper, we present a novel technique for adapting local image classifiers that are applied for object recognition on mobile phones through ad-hoc network communication between the devices. By continuously accumulating and exchanging collected user feedback among devices that are located within signal range, we show that our approach improves the overall classification rate and adapts to dynamic changes quickly. This technique is applied in the context of our *ANONYMOUS SYSTEM* – a mobile phone based museum guidance framework that combines pervasive tracking and local object recognition for identifying a large number of objects in uncontrolled museum environments.

1. INTRODUCTION AND MOTIVATION

The increasing computational possibilities of today's mobile devices –especially of camera-equipped mobile phones– enable image retrieval and object recognition applications on them. Examples are museum guidance or city guidance applications that allow for the identification of objects by simply taking photographs of them. The challenge of these methods is to become as robust as possible – even when applied in highly dynamic, large scale and uncontrollable public environments. Museums in particular are demanding of image classification algorithms because hundreds to thousands of objects have to be recognized from different perspectives and under varying lighting conditions. Most related systems (e.g., [5, 6]) use mobile devices only as a front end where captured images are sent to a server for classification. Such centralized systems do not scale well with an increasing number of users and an increasing number of classification requests.

To address this problem, we have developed an adaptive museum guidance system, called *ANONYMOUS SYSTEM* [16], which performs the classification task directly on camera-equipped mobile phones instead of sending and receiving information to and from a server. Since every visitor does this in parallel, and independent of one another, the wait-



Figure 1: Application of *ANONYMOUS SYSTEM*.

ing time for retrieving classification results is minimal. To guarantee a maximum classification rate, however, *ANONYMOUS SYSTEM* has to adapt to dynamic changes in the environment (e.g., changing daylight) and users' behavior (e.g., how visitors approach particular objects).

In principle, *ANONYMOUS SYSTEM* works as follows: Objects that are located in a same area are trained into the same 3-layer neural networks that are used directly on the phone for local classification. The correct classifier for a particular location is selected on the fly by the phone through continuous estimation of its rough location. Pervasive tracking of the devices is enabled by a sparse set of distributed Bluetooth beacons [15]. The set of pre-trained classifiers are downloaded from a server onto all phones of visitors when entering the museum. During their stay, visitors can classify objects in less than 1 second by taking pictures of them. Before presenting exhibit-unique multimedia content, the classification result is displayed as a probability ordered sequence of potential recognition candidates (cf. figure 1). The first candidate in this sequence has the highest probability of being the correct object. The second candidate has the second highest probability, and so on. The user is able to select the correct object with a minimum number of clicks. If the object is not classified correctly, for instance, the user has to select another object from the list that is not the top candidate. By doing so, user feedback is provided implicitly, which is stored in combination with the captured image on the phone. This feedback information is uploaded to the server when the visitors leave the museum. It allows adapting the classifiers before transmitting them to new visitors that are entering the museum. Our on-site studies proved that with this simple adaptation technique, stable recognition

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

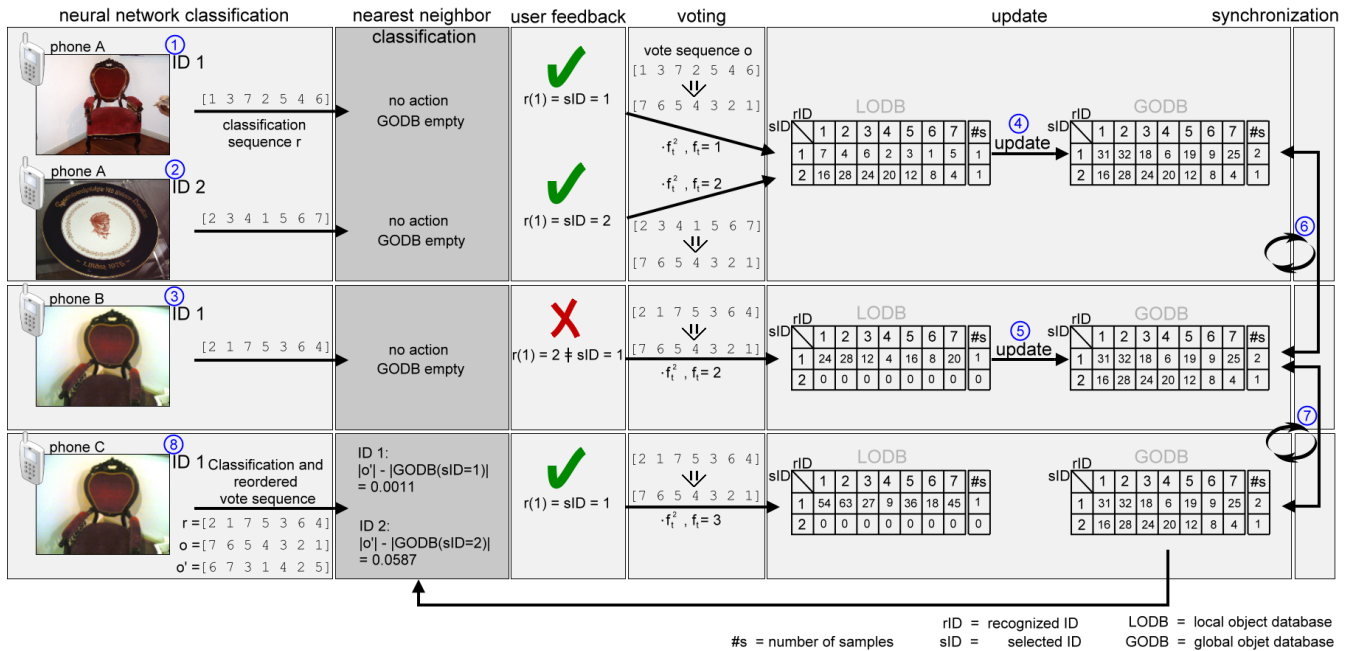


Figure 2: Adaptation example (sequence numbers encircled): Object 1 (chair) is first successfully recognized on phone A (1). The voted classification sequence is time-weighted with f_t^2 and stored in A’s LODB. Then object 2 (plate) is successfully recognized on phone A, and the LODB is expanded accordingly (2). Phone B fails in recognizing object 1 (3), and stores the weighted vote sequence in its local LODB. Then phones A and B are able to establish a connection. They both update their GODBs first (4+5) before synchronizing them (6). Next, a connection between phones B and C is established and GODBs are synchronized (7). The classification of object 1 on phone C (8) would fail when relying solely on the feedback from the neural network, but it will succeed if the nearest neighbor classification is performed together with the GODB.

rates of 82% can be reached under realistic conditions (i.e. tested in a museum over several days with real visitors) [16, 17].

One drawback to this approach is that an adaptation to changes that happen during the actual stay of a visitor is not possible. Once downloaded at the entry, the classifiers remain unchanged until leaving the museum. Therefore, the classification rate can drop significantly after sudden changes, such as temporally varying lighting conditions (e.g., caused by sunlight).

Addressing this problem is the motivation of the work presented in this paper. We explain a technique that distributes the user feedback information during runtime through ad-hoc network connections between local devices. By doing so, we enforce cooperative classification improvements during the actual stay of the visitors. The general functionality of our technique has been tested with a small number of real devices in a museum. For proving its scalability, however, we have developed a simulator that evaluates our method for many hundred devices under several conditions. The simulation parameters have all been gathered in a museum, and are therefore realistic. We will show that ad-hoc phone-to-phone synchronization not only leads to higher overall classification rates, but also to quicker adaptations to dynamic changes during runtime.

2. RELATED WORK

Related work can be separated into two categories: Wireless ad-hoc communication for mobile devices and distributed

classification systems.

A popular example for wireless ad-hoc networks is car-to-car (inter-vehicle) communication. This research field investigates the exchange of dynamic traffic and other information between moving vehicles using WiFi signals and GPS tracking [10, 4]. In addition, several applications for mobile devices exist that utilizes wireless technologies, such as Bluetooth, for ad-hoc communication. Aalto et al. [2], for instance, propose a push-based advertising system. A sensor retrieves addresses of mobile devices and forwards them to an advertisement server. This server then sends location aware advertising directly to the mobile device via WAP. A similar system called BlueTorrent [7] supports peer-to-peer file sharing based on Bluetooth enabled devices. The file data is separated into blocks which are broadcasted to nearby pedestrians and interchanged between moving users. Murakami et al. [9] implemented a simplified version of the hierarchical ad-hoc on-demand distance vector method for efficient data routing in large mobile phone networks. Based on Bluetooth, their system addresses applications like mobile games, tracking applications and mobile emergency systems. Wang et al. [13] categorize games with respect to the way how game engines are updated in mobile ad-hoc networks (e.g. asynchron, synchron, real-time, etc.). They also discuss challenges that arise when developing mobile P2P games in Java ME using the available Bluetooth API (see also [11]). In general, our approach is similar to [7]. But the ad-hoc communication we propose is combined with adaptive image classification. Instead of supporting cooperative file down-

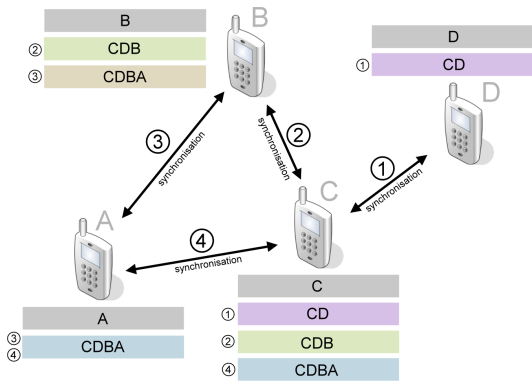


Figure 3: Propagation example of classification and feedback data in a mobile ad-hoc network (sequence numbers encircled): Initially, all phones store their local *LODBs* only. First, phone C synchronizes with phone D (1), then C with B (2), then A with B (3), and finally A with C (4). After these steps, A, B, C share the same and most up-to-date information that have been collected by all phones. Phone D would also be up-to-date if it would now synchronize with either A,B, or C.

loads, we focus on increasing the recognition rate and on reacting to dynamic changes quickly in a cooperative manner. The goal of distributed classification approaches is to outsource the complex classification task to multiple networked nodes. Each node performs a sub-classification that is merged into the final result.

Besides various approaches that are based on static networks, Luo et al. [8] introduce a distributed classification method for P2P networks. In contrast to client-server systems, their method is scalable by adding new nodes. Each node builds its local classifiers based on a modified version of the pasting bites method, while the results of all classifiers are combined by plurality voting. Wolff et al. [14] executes a sequential association rule mining algorithm on local databases of each node in a P2P network. Each node then participates in distributed majority voting to retrieve the combined result of all connected peers. Siersdorfer et al. [12] describes a method for distributed document classification in P2P networks. Each node generates classifiers for locally stored documents, which are propagated through the network to help other nodes in improving their own classifiers.

This last example is similar to our approach. The main differences are that in our case the network connections are highly dynamic and that the classification duration does not increase with the amount of exchanged classification data.

3. ENHANCED IMAGE CLASSIFICATION THROUGH AD-HOC NETWORKS

The main challenges in image classification are the determination of a well-defined feature set as well as the collection of sufficient training samples in order to achieve invariance to different appearances of the same captured content. Since the computational possibilities (memory, CPU) of mobile phones are still restricted in comparison to desktop PCs or PDAs, sophisticated feature descriptors ([3]) that are robust against varying content can only be applied at the cost

of high computational effort. In turn, this leads to extensive waiting times that distract from the museum experience, which is a situation not tolerated by the visitors in general [16]. In addition, museums are highly uncontrolled environments with respect to illumination as well as to visitor behavior. Thus, the training set that was recorded in an initial training phase can differ significantly from the data to be classified during runtime. Even highly sophisticated feature sets and classifiers can not dynamically compensate for this. Consequently the probability of false positives increases in dynamic situations, such as changing illumination or different user behavior.

The basic idea that is presented in this paper, is the collection of information about the individual classification behavior of all visitors simultaneously on each user's local device. This information is derived from user feedback that is recorded during individual recognition tasks. It is then shared with the phones of other visitors whenever possible. A continuous synchronization within such dynamic ad-hoc networks allows adapting and improving the classification rate of all users over time.

In section 3.1 we will explain the principle adaptation method, and in section 3.2 we will show how the synchronization can be realized in practice.

3.1 Adaptation

As explained in section 1, we retrieve an ordered sequence (r) of potential object candidates as a response from a 3-layer neural network after each classification. The first candidate in this sequence has the highest probability of being the correct object. In cases of false positives, however, it is likely that the correct object is still ranked under the top candidates - even though it is not the correct match. It is also likely, that the classification sequence is slightly different for each object. These two assumptions enable a consecutive classification step: Instead of relying solely on the candidate with the highest probability, we take the object-individual classification sequence into account. This allows for a dynamic compensation to environmental changes, as will be explained below.

With respect to figure 2, a vote is assigned to each ranked candidate in the classification sequence, after a recognition is carried out. The candidate with the highest probability ($r(1)$) gets the highest vote, the second element ($r(2)$) gets the second highest vote and so on. The resulting vote sequence (o) is multiplied with the square of a time-dependent factor f_t^2 to amplify more recent classification results. It is then stored in a local object database (*LODB*) table, which is individually managed on each phone. The size of the *LODB* table is $N \times N$, where N is the number of objects that are trained in the neural network¹. The columns of the *LODB* indicate the recognized object IDs (rID) of r , as classified by the neural network. The rows indicate the selected object IDs (sID) that result from the users' feedback after each classification. Thus, the time-weighted vote sequence o computed from each classification sequence r is stored in the *LODB* at row sID . In the optimal case, sID and $r(1)$ are always equal. In this situation, the diagonal of the *LODB* will store the highest votes. If, however, the neural network classification must frequently be corrected by user feedback, this will lead to higher votes on *LODB*'s off-diagonal entries.

¹The number of objects can be restricted through location-aware services, such as pervasive tracking.

If the same sID results from the user feedback of multiple recognition tasks, the new vote sequences are weighted and added to the existing entries in the corresponding $LODB$ row. A row-individual counter is then increased to indicate how many samples have been accumulated.

Our goal is to distribute and synchronize the information collected from the feedback of each user to all other users and apply this knowledge for adapting the classification of each individual visitor. If, for instance, the classification of one particular object will fail for many users due to the change of environmental lighting (e.g., sunlight passing through a window), this will be detected through a similar user feedback and votes of classification sequences for this object. The $LODB$ of each user will then reveal a similar pattern at row sID that corresponds to the object. If this information can be shared with users that have not yet approached the object, it will help to adapt and improve the classification for this object before the users approach them. In the following, we will explain how this shared information can be used for adapting the local classification process, while section 3.2 goes into more detail about how the synchronization takes place.

For now, let us assume that an ad-hoc network connection can be established between phones that are located within signal range. In addition to the $LODB$, each phone stores a global object database ($GODB$) that contains the information gathered from other phones and from local classification trials. The $GODB$ has the same structure as the $LODB$. Each phone's $GODB$ will be updated with the local information stored in the $LODB$, but it will also be synchronized with information stored in the $GODB$ s of other phones. How these updates and synchronization steps are realized will be explained in section 3.2.

The synchronized $GODB$ s that contain classification and feedback information from multiple phones and users allow adapting and improving the local classification process. Again, a classification sequence r is the result after a new recognition attempt.

Instead of relying exclusively on $r(1)$, as the result suggested by the neural network, the information stored in the $GODB$ is also considered in a second step. We perform a nearest neighbor classification between the vote sequence that is derived from r and the row vectors stored in the $GODB$. Note, that before finding the nearest neighbor, the order of the entries in o has to be rearranged and normalized. This is necessary to match them with the order stored in the $GODB$. The rearranged sequence is denoted with o' , and is used for nearest neighbor classification instead of o . Finally, the ID that corresponds to the computed nearest neighbor represents the overall classification result. Figure 2 illustrates an example, where the initial classification through the neural network would fail. Only the additional comparison with the $GODB$ in the second step leads to a correct result. As explained earlier, the unchanged voting sequence o is time-weighted with f_i^2 and added to the $LODB$. The next section describes how the $GODB$ s are updated and synchronized.

3.2 Synchronization

As mentioned in section 3.1, the $GODB$ s are synchronized among phones as soon as they are within signal range. The ad-hoc network is highly dynamic as visitors are moving continuously through the museum. Thus, we can only transfer data between two directly connected phones without rout-

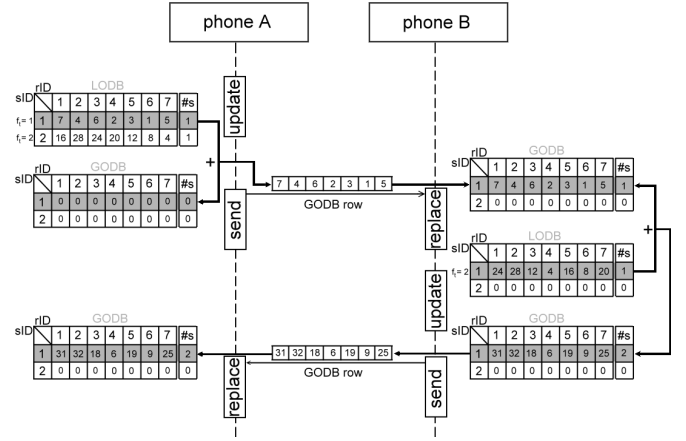


Figure 4: Steps for synchronizing one $GODB$ row: If phone A synchronizes with phone B, A's $GODB$ row has to be updated before being transmitted. After receiving it at B, this row replaces the existing row in B's $GODB$. Then, B has to perform an update and sends the result back to A, where it replaces the corresponding $GODB$ row. After doing this for each row, the $GODB$ s of A and B are synchronized and contain the most up-to-date information.

ing. Indirect connections over multiple phones would be interrupted frequently and would therefore not be stable. In section 4, we explain how the transmission of the $GODB$ s is realized in our current implementation.

If a connection is established, we carry out the following synchronization steps for each $GODB$ row on both sides (cf. figure 4): As can be seen in figure 2, the number of samples that have been accumulated in all rows of all $GODB$ s and $LODB$ s are recorded in sample counters. Comparing each corresponding row in the $GODB$ s of both connected phones, the row with the largest sample counter is selected. This row is first updated with the corresponding row of the local $LODB$. The *update* operation adds all entries of the $LODB$ row to the entries of the $GODB$ row –including the sample counter– and then resets all $LODB$ row entries to 0 (including the sample counter). As explained in section 3.1, new samples can be added to the $LODB$ row via new recognition tasks.

After updating the $GODB$ row, it will be sent to the other phone, and having it received there it will *replace* completely the existing $GODB$ row with the same sID (including the sample counter, as in the previous step). Then, a second *update* with the local $LODB$ row on this side is carried out. The result is sent back and replaces the corresponding $GODB$ row on the other side.

This synchronization sequence ensures that no classification and feedback information is considered more than once on the same phone. Otherwise, the data transmitted over multiple hops and at some point received by its originator would be incorrectly overemphasized, and lead to incorrectly weighted classification results. Therefore, loops are avoided in our ad-hoc network communication.

These synchronization steps are repeated row by row, until the full $GODB$ is synchronized, or the connection has been lost. It is started again, as soon as a new connection can be established –either with the same phone or with another

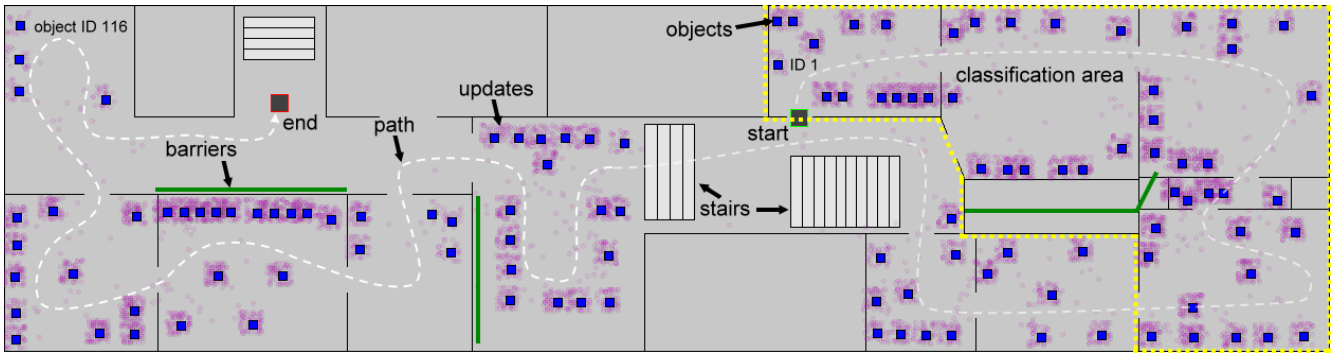


Figure 5: Simulation screenshot: The floor plan of the City Museum of *ANONYMOUS CITY* with the locations of the exhibited objects (blue boxes). For each successful ad-hoc synchronization during simulation, the location of both phones have been tagged (red dots). The majority of successful data exchanges are in the proximities of closely located objects, since visitors gather together in these areas for a longer time.

phone. Note, that rows are only synchronized if a higher sample count on either one side exists. This indicates that one side has more reliable results. If the sample counts are equal, no synchronization is triggered. This would also be the case if a new connection between two already synchronized phones will be established again. Time-weighting the vote sequences ensures that more recent classification approaches are up-weighted, while outdated information is down-weighted. Therefore, each *GODB* represents the most up-to-date classification state. The *GODB*'s sizes remain constant on all phones. Therefore, synchronization time does not increase.

Figure 3 shows an example of how locally connected information is propagated through ad-hoc network connections.

4. REALIZATION AND RESULTS

In this section we will describe the practical challenges that arise when implementing an ad-hoc phone-to-phone network (subsection 4.1), as well as an application for simulating a stream of visitors through a museum (subsection 4.2) that is used for validating our approach (subsection 4.3).

4.1 Implementation

In our current Java ME implementation, we apply Bluetooth for wireless communication since it is widely established and integrated in most mobile phones.

Three general steps are carried out to transfer data via Bluetooth between two devices: First, a scan for nearby devices has to be performed (*inquiry*). Second, for each of the detected devices, a *service search* must be executed in order to exchange connection and service parameters. Finally, the *connection* is established.

However, several practical drawbacks arise when applying Bluetooth for ad-hoc networks. For instance, during *inquiry*, the phone enters the internal *inquiry substate*. During this time, the phone can not be detected by other phones until it leaves this substate and enters the *inquiry scan substate*. To ensure that not all devices enter the same substates synchronously, and therefore never detect each other, we introduce a random waiting time period t_w between both substates. Empirically we found that $t_w=5-9$ seconds is optimal for avoiding continuous deadlocks. With this additional waiting time, we can estimate the duration D that is required

for establishing an ad-hoc connection between n_d phones:

$$D \approx t_i + n_i(t_i + t_w) + n_d(t_s + t_r), \quad (1)$$

where t_i is the inquiry time that is proportional to the number of devices within proximity and their distances. If devices can be detected, t_i is approximately 1-5 seconds per device (for Java ME) but is not lower than 10.24 seconds² – even if no devices can be detected. The parameter t_s is the times required for carrying out one service search (on average 6 seconds per device), and t_r is the time required for transmitting the *GODB*s and service parameters in both directions. Should no devices be detected during the first inquiry, we have to repeat this step on average $n_i=2-5$ times, with a delay of t_w seconds.

The transmission time t_r depends on the amount of synchronization data that has to be exchanged. In our case, the *GODB* is of the size $N \times N \times 4$ bytes (with N being the number of trained objects), and the list of sample counters is of the size $N \times 4$ bytes. With a measured average transfer rate of ~ 40 kbyte/s for Bluetooth 1.1 (specified with 732,2 kbit/s) on Nokia 6630 mobile phones, we synchronize two devices with $t_r < 1$ second, when assuming that $N=50$ or less. Phones equipped with Bluetooth 2.0 (specified with 2.1 Mbit/s) would allow much faster rates. In practice, we found that D ranges from 65-120 seconds for synchronizing three phones, for example.

Note, that this communication process runs in parallel to the actual *ANONYMOUS SYSTEM* application to ensure that ad-hoc synchronization is seamless to the users.

4.2 Simulation

We have tested and validated our approach practically with three Nokia 6630 mobile phones directly in the City Museum of *ANONYMOUS CITY*. Although these experiments showed that image classification through a synchronization between the three devices improves and adapts to dynamic situations, the number of devices is far too low to represent realistic conditions. Therefore, we have developed a simulation application that simulates a stream of visitors in a museum over time. It simulates museum visitors, but not our algorithm itself. In fact, the simulator is connected

²“Specification of the Bluetooth System”, 2003, <http://www.bluetooth.com>

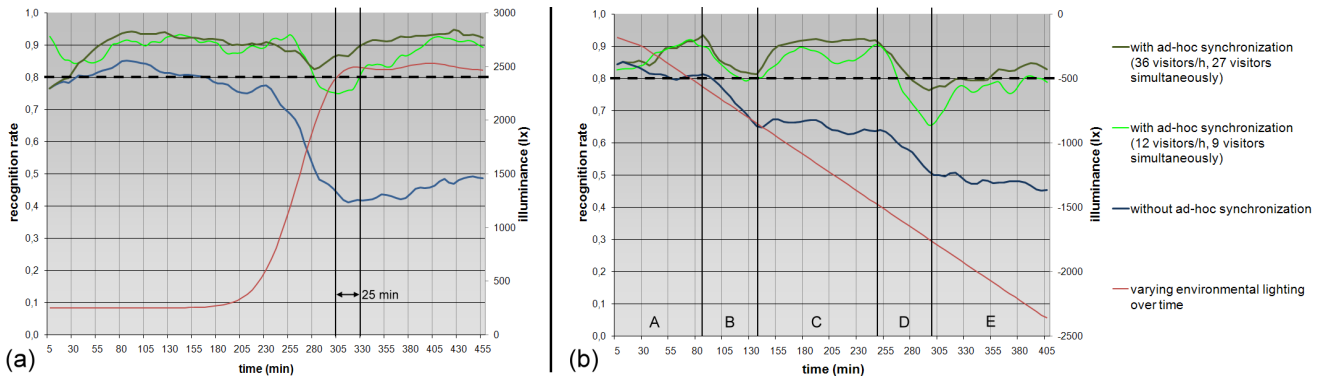


Figure 6: Average recognition rates for an unadapted (blue), and an adapted image classification (light green: 12 visitors/h and dark green: 36 visitors/h) under measured lighting changes (red). The adapted classification always performs better than the non-adapted classification, is less fragile to dynamic environmental changes, and recovers quickly to sudden changes.

via TCP/IP to the Java ME wireless toolkit emulator that executes our software in exactly the same way as it would be executed on the mobile device. This gives us the opportunity to evaluate different configurations and user scenarios for a large number of users, which are currently not possible to achieve in this extent with real users. All parameters that are used for simulation (such as a floor plan and exhibits of a museum, visitor behavior, lighting conditions, and Bluetooth signal range) have been investigated and measured in advance in the City Museum of *ANONYMOUS CITY* to guarantee the most realistic results possible.

Figure 5 illustrates a screenshot of the simulation tool. The floor plan and the locations of the exhibited objects have been measured in the museum. For each of the 50 objects³ that are located in our test area (framed with dotted yellow lines) that we consider for our simulation, we captured a pool of 200 images from multiple perspectives and distances at two different time of days. Half of these images were used for an initial training of the neural networks while the other half was used for the simulation itself. These objects were applied in previous evaluations of *ANONYMOUS SYSTEM* [15, 17], and therefore represent a reference for measuring the classification improvements. As mentioned above, our *ANONYMOUS SYSTEM* is emulated on request of the simulator. Taking photographs of an object for recognition is simulated by randomly selecting one entity of the corresponding object’s image pool that was not used for initial training. The user’s feedback is also simulated by always picking the correct object ID after classification. The range of the Bluetooth signal was defined to be 8 meters. We measured the transmission of the radio signal through walls in the museum. These measurements are also taken into account in the simulation. Thick walls that did not transmit the signal are marked with signal barriers (green lines, cf. figure 5). Consequently, blocking and transmitting room elements are considered during the simulation, but complex reflections of the radio signals are not.

The visitor behavior is modeled as follows: We do know that the visitors are guided in the museum to follow a predefined path that leads them through different exhibition contexts. Although they have the freedom to move freely, we observed

³From a total of 116 objects being displayed on the entire floor.

that most visitors will actually follow this path (from start to end). On this rough path, however, simulated visitors are free to move randomly to arbitrary exhibits (e.g., that are located in the same room) and can skip or move back to objects. In the simulation, visitors enter the museum at random times in between the opening hours. The speed with which visitors walk from exhibit to exhibit, as well as the examination time for each object is also selected randomly within the range of our observations.

The synchronization between two phones is simulated by a scheduler that continuously triggers an inquiry and service search for each simulated visitor, as explained in section 4.1. Whether or not a connection can be established and a synchronization is successful depends on the visitors’ movements and on the time that is required for the synchronization which we derived from equation 1 and the measured parameters explained in section 4.1.

4.3 Evaluation

With our simulation, we have evaluated two different scenarios to prove that an ad-hoc synchronization adapts to dynamic changes and therefore improves the image classification.

The first scenario shows the development of the recognition rate over time for rapidly changing ambient illumination. As mentioned above, the illumination changes are caused by the increasing sunlight that has been measured at these times within our test area of the museum. The result is plotted in figure 6a. We simulated 84 and 252 visitors (12 and 36 visitors per hour, entering the museum randomly) over the opening hours of one day. Note, that 12 visitors per hour equals the average number of visitors per day for a german museum, as it is reported in [1]. The average walking speed was assumed to be 4 km/h, and the average examination time was set to a range between 1 second (visitor just moves on immediately after classification) and 90 seconds (visitor listens to the multimedia information after classification).

Figure 6a presents the average recognition rates of all visitors at a particular time that can be achieved with (light/dark green) or without (blue) adapting the image classification through our ad-hoc synchronization approach. As it can be seen, the illumination changes rapidly (due to sunlight) after 210 minutes past the museum opening, and stabilizes after

305 minutes. In this case, the recognition rate of the non-adapted classification process drops from $\sim 80\%$ to $\sim 45\%$. The ad-hoc synchronization, however, leads to a higher overall recognition rate and to a quick adaptation to environmental changes, such as the lighting conditions in this example. The adapted classification drops relatively little, and can recover after a short time (in this example: after roughly 25 minutes the recognition rate for 12 visitors/h reaches the original level of the non-adapted classification, and even improves further). The classification performance after the completed adaptation is around $\sim 45\%$ higher than the performance of non-adapted classification under the same condition.

In the second scenario, we investigate how our approach performs for non-abrupt, but slight and continuous changes, such as a constant decrease in illumination. This is shown in figure 6b. Note, that we apply a synthetic illumination curve rather than true measurements for this experiment. With linearly decreasing illumination, the recognition rate of the non-adapted classification decreases in intervals. The reason for this seems to be the specialization of the neural networks weights which display different sensitivity to varying inputs. The behavior of the adapted classification is correlated to the behavior of the neural network. If the recognition rate of the neural network decreases (sections B and D in figure 6b), the recognition rate of the adapted classification decreases too. However, the adapted classification is always better than the non-adapted classification. If the recognition rate of the neural network stagnates (sections A, C, and E in figure 6b), the adapted classification improves through ad-hoc synchronization.

One other observation that can be made when investigating the distribution of successful synchronizations, as visualized in figure 5 is that most of them appear at locations with a high object density. The reason for this is that at these places, visitors will remain for a longer time within signal range, and synchronizations become more likely. But on the other hand, this also implies that objects which are located more isolated from others will cause lower recognition rates. This is the case if all visitors follow a similar path through the museum. In this situation, synchronizations will only be beneficial for quick adaptations if they are performed before the actual recognition task is carried out. This is more likely for areas with a dense object distribution, than for areas with a sparse one.

5. LIMITATIONS AND OUTLOOK

The most limiting factor of our current implementation, is the relatively long response time. As it can be seen in figures 6a and 6b, the network requires approximately 25-45 minutes for full recovery –depending on how far it is decreased. This is mainly due to performance limitations of Bluetooth communication (see section 4.1). Our simulation revealed that with faster link connections (e.g., as possible with WiFi) and an sufficiently large number of visitors the classification rate remains constantly high (i.e., does not drop down with environment changes). This, together with new approaches that additionally store and provide adaptation parameters locally (e.g., on memory-equipped signal beacons) will be investigated in future.

We believe that such an ad-hoc adaptation approach can also be beneficial for mobile image classification techniques in other application areas, such as outdoor tourist guidance.

6. REFERENCES

- [1] Institute for museum research. *Statistische Gesamterhebung an den Museen der Bundesrepublik Deutschland für das Jahr 2006*, 2006.
- [2] L. Aalto, N. Göthlin, J. Korhonen, and T. Ojala. Bluetooth and wap push based location-aware mobile advertising system. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 49–58, 2004.
- [3] W. C. Chen, Y. Xiong, J. Gao, N. Gefand, and R. Grzeszczuk. Efficient extraction of robust image features on mobile phones. In *The Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.
- [4] N. Eude, B. Ducourthial, and M. Shawsyky. Simulation of car-to-car messaging: Analyzing the impact on road traffic. In *Proceedings of the 7th IFIP International Conference on Mobile and Wireless Communications Networks*, 2005.
- [5] G. Fritz, C. Seifert, and L. Paletta. A mobile vision system for urban detection with informative local descriptors. *Proceedings of the 4th Int'l Conference on Computer Vision Systems*, page 30, 2006.
- [6] J. S. Hare and P. H. Lewis. Content-based image retrieval using a mobile device as a novel interface. *Proceedings of the first international workshop on mobile vision*, pages 64–75, Dec. 2004.
- [7] S. Jung, U. Lee, A. Chang, D.-K. Cho, and M. Gerla. Bluetorrent: Cooperative content sharing for bluetooth users. *Pervasive and Mobile Computing*, 3(6):609–634, 2007.
- [8] P. Luo, H. Xiong, K. Lü, and Z. Shi. Distributed classification in peer-to-peer networks. In *Proceedings of the 13th international conference on Knowledge discovery and data mining*, pages 968–976, 2007.
- [9] H. Murakami, A. Ito, Y. Watanabe, and T. Yabe. Mobile phone based ad hoc network using built in bluetooth for ubiquitous life. pages 137–146, 2007.
- [10] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode. Trafficview: traffic data dissemination using car-to-car communication. *SIGMOBILE Mobile Computing and Communications Review*, 8(3):6–19, 2004.
- [11] N. Ravi, P. Stern, N. Desai, and L. Iftode. Accessing ubiquitous services using smart phones. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, pages 383–393, 2005.
- [12] S. Siersdorfer and S. Sizov. Automatic document organization in a p2p environment. In *European Conf. on Information Retrieval*, pages 265–276, 2006.
- [13] A. I. Wang, M. S. Norum, and C.-H. W. Lund. Issues related to development of wireless peer-to-peer games in j2me. In *Proceedings of the Advanced Int'l Conf. on Telecommunications and Int'l Conf. on Internet and Web Applications and Services*, page 115, 2006.
- [14] R. Wolff and A. Schuster. Association rule mining in peer-to-peer systems. page 363, 2003.
- [15] XXX. Xxx. *IEEE Multimedia*, 2007.
- [16] XXX. Xxx. *To appear in Journal of Computer Graphics and Applications*, 2008.
- [17] XXX. Xxx. *Journal of Personal and Ubiquitous Computing*, 2008.