Facilitating Direct and Ubiquitous Mobile Computer Vision

Hanno Wirtz, Jan Rüth, Klaus Wehrle
Chair of Communication and Distributed Systems (COMSYS)
RWTH Aachen University
{wirtz, rueth, wehrle}@comsys.rwth-aachen.de

ABSTRACT

Computer Vision (CV) approaches, e.g., as the basis for mobile Augmented Reality (AR), depend on continuous Internet access for image uploads and large-scale Internet databases to perform image recognition against comparison images. This dependency impedes the ubiquity of applying CV in spontaneous, real-world mobile scenarios as users may not have continuous Internet access. Furthermore, the underlying databases are inherently volatile and may only afford sporadic coverage.

We hence propose DMCV (Direct Mobile Computer Vision), leveraging the proliferation of wireless communication capabilities in mobile and stationary devices to remove this dependency and to transmit CV image descriptors directly between mobile devices and recognizable objects. Building on 802.11 and Bluetooth, we explore the design space of local wireless CV information discovery and provision and evaluate to which degree each technology affords ubiquitous mobile CV. We show the feasibility and performance of our approach on commodity phones and evaluate the benefits provided by ubiquitous direct CV.

1. INTRODUCTION

Mobile Computer Vision (CV) is an integral part of proposed mobile Augmented Reality (AR) approaches [12], location-based services [14,17], and ubiquitous computing [13,21]. As illustrated in Figure 1 (left), current approaches [2, 9, 12, 14, 17] realize mobile CV via *intermediate* and *centralized* Internet-based image recognition services that build on vast image databases for comparison. As recognized in [4], these requirements of Internet connectivity for image uploads and the existence of comparison images of the object to be recognized in the respective database negate an unrestricted and spontaneous, i.e., ubiquitous, use of CV.

In this paper, we mitigate these requirements in order to facilitate ubiquitous mobile CV and propose DMCV (Direct Mobile Computer Vision), local and immediate discovery and provision of CV-recognizable objects and their information. Specifically, DMCV targets the readily available mass of smartphones as ideal consumer devices for mobile CV and leverages the comprehensive availability of 802.11 and 802.15.1 (Bluetooth) wireless communication capabilities available at objects via smartphones, embedded devices[1], 802.11 APs, laptops, etc. Objects then may be persons, buildings, monuments, or stores and hold the CV information required to detect them in a given CV technique. At its core, our design enables the spontaneous, local, and direct transmission of the CV data required for recognition from the respective object to smartphones in range. Smartphone users are then able to recognize and annotate objects in the camera view (cf. Figure 1, right).

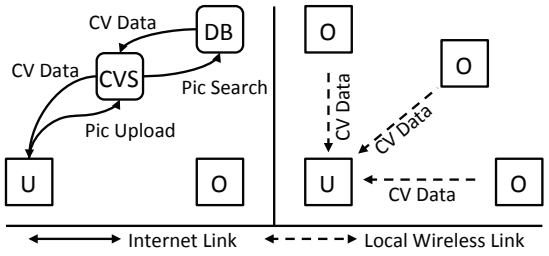


Figure 1: Currently, a mobile user (U) needs to upload an image of the object (O) to an Internet-based CV service (CVS) for a database (DB) search and comparison (left). In DMCV (right), objects directly provide their CV data locally and wirelessly.

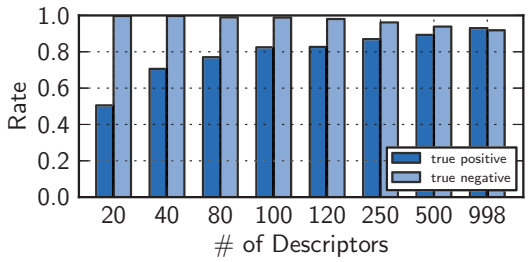
In contrast to Internet-based, centralized CV approaches, DMCV then provides the following benefits in realizing mobile ubiquitous CV and AR. i) Building on 802.11 and 802.15.1 wireless communication enables *Internet-independent* discovery and transmission of CV information as well as AR annotation data, e.g., underground, abroad, or in rural areas. ii) Local wireless communication inherently incorporates the *context and locality* of the mobile user whereas Internet-based services abstract from this context. iii) CV information only provides the data required for object recognition, improving on the time and computation overhead of uploading and comparing an image at an Internet service. iv) Object owners maintain the CV and annotation information they make public, affording *flexibility and control*.

We motivate DMCV with previously obtained results [22] that show high object recognition rates already with low amounts of CV information, allowing for transmission of CV information within the dynamics of mobile scenarios. Fig-

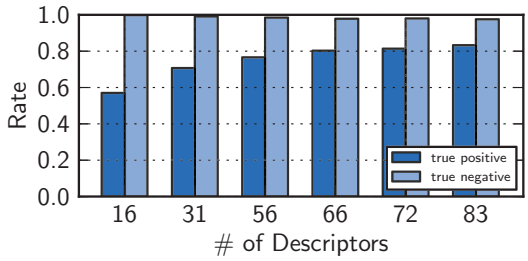
[1]E.g., Arduino Yún, Raspberry Pi.

MUM '14, November 25 – 28 2014, Melbourne, VIC, Australia

http://dx.doi.org/10.1145/2677972.2677974.



(a) Building recognition.



(b) Face recognition.

Figure 2: True positive and true negative recognition rate [22] over number of ORB [19] descriptors.

ures 2(a) and 2(b) quantify this success rate in recognizing a building and a human face in two popular image datasets from the CV community [10, 18] over the number of 32 Byte ORB descriptors. Transmitting only the 80 most distinctive descriptors, i.e., 2560 Bytes or two 802.11 frames, allows positive recognition rates of about 80 % (true positive). Equally important, the low number of ORB descriptors still comprehensively avoids false recognition of the respective object when regarding other objects (true negative). Building on these results, this paper proposes a comprehensive approach to ubiquitous and local CV on smartphones with negligible time and communication overhead.

We implement DMCV for both Android and iOS smartphones using the publicly available OpenCV library [6] and enable objects to transmit a set of ORB [19], FREAK [1], or SURF [5] descriptors that is subsequently recognized in the smartphone's camera view. Our evaluation of the CV performance on Nexus 5 and iPhone 4 and 5S phones then shows both the feasibility of immediate CV data provision and the impact of the phones' computation capabilities. Smartphones further transmit their current location to allow objects to deliver the CV information that correctly represents the pose of the user towards the object. We show the comparably small overhead of comprehensively maintaining CV data for the possible user poses. Furthermore, we realize DMCV within both 802.11 and 802.15.1 (Bluetooth), as the prevalent wireless communication mechanisms supported by smartphones. We evaluate both with regard to our goal of facilitating *ubiquitous* mobile CV, i.e., unhindered by the communication and time overhead imposed by the communication aspect of DMCV. Trading time overhead and communication scope for reliability, DMCV further supports CV data transmission via Beacon Stuffing [7], i.e., 802.11 communication without the overhead of a network association.

In short, DMCV provides the following contributions:

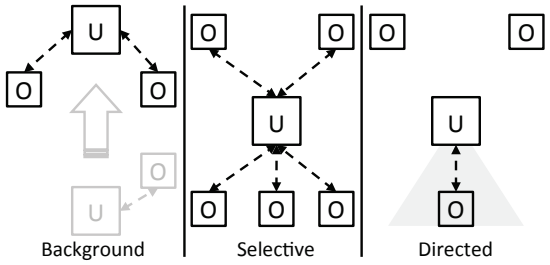


Figure 3: Usage variants in DMCV: A user (U) continuously senses object (O) CV information in the background, via triggered discovery and selection, and in directed detection of a specific object.

i) A mechanism for CV information exchange that allows devices to ubiquitously discover all recognizable objects in wireless communication range,
ii) an integration of location context to account for changing user perspectives in real-life mobility, and
iii) a feasible mechanism for user-driven content creation and updates by enabling object owners to control the provided information.

We illustrate the envisioned usage variants and requirements of mobile CV in Section 2 and address these in our design of DMCV in Section 3. Section 4 describes our prototype implementation for the popular iOS and Android platforms and evaluates the feasibility and performance with regard to the communication and object recognition aspects of DMCV. Section 5 compares our design with existing approaches and Section 6 concludes the paper.

2. BACKGROUND

In this section, we first illustrate the envisioned usage variants and applications of DMCV. We then briefly revisit the requirements for ubiquitous mobile CV and our argumentation for direct provision of CV data by objects.

2.1 Usage Variants

We envision DMCV to support the application of mobile CV in the heterogeneity of scenarios that mobile users encounter. For this reason, we differentiate between three distinct variants of *using* mobile CV that, in our opinion, comprehensively cover all possible application scenarios. Figure 3 illustrates these variants, in the following we briefly explain each variant along example application scenarios.

Background: The mobile device autonomously and continuously discovers and collects CV data from objects in the vicinity, following the mobility of the user. At any point in time, the user may then scan her surroundings with the phone camera, recognizing objects using the collected CV data and displaying the associated information. Additionally, the user may configure object information received in the background to trigger an alert, e.g., a vibration alert when encountering a specific object. Example scenarios for background application of DMCV are i) user mobility within a city scenario, where monuments, stores, buildings, and other mobile users provide their CV information proactively, and ii) an exhibition, where the user roams about and collects CV and annotation information to display once she scans the surrounding vendor spaces for demonstrations or products to decide where to go next.

Selective: Within a given location or context, a user purposefully scans for the objects available in her vicinity, e.g., via an 802.11 or Bluetooth scan or by sending a customized wireless frame. Objects respond with a semantic identifier, allowing users to select an object and request its CV data, e.g., when searching for and trying to identify a specific object and display its application content. By scanning the surroundings in the camera view, the CV application can detect the specific object, highlight it, and annotate it with the provided information. Application scenarios for selective mobile CV are i) exhibition visitors that strive to detect and query for a specific vendor or for a semantic identifier of interest, e.g., "mobile 3D gaming", and ii) city tours in which visitors want to localize a specific monument or building and gather information about it.

Directed: In contrast to the selective variant, where a user selects an object by its semantics, the current user pose and perspective determines the object to be queried and recognized. This is the traditional CV and AR application scenario [12], i.e., looking at an object through the phone camera, a user wants to gather information about this specific object. Example application scenarios for directed CV are i) mobile users that want to derive the semantics of an object in the first place, and ii) a user that immediately queries the exhibition space or product he is looking at for application information.

We further highlight the application of and differences between the respective variants in Section 3.1.1. To this end, we embed each variant into our overall design.

2.2 Requirements for Ubiquitous CV

Ubiquitous support of CV in mobile scenarios and applications by direct provision of CV data requires adjusting both the CV and communication components of current approaches. In these approaches, comprehensive provision of CV data suffers from the aforementioned challenges of fluctuating or non-existing Internet access as well as the need for centralized databases [4].

We thus argue that mobile CV benefits from an immediate, distributed, and lightweight approach that provides only the essential CV data exactly in the context in which this data is consumed. Specifically, enabling object owners (or proprietors) to simultaneously function as content creators, maintainers, and providers would remove the need for all-encompassing, inherently volatile central databases and content management within them.

A suitable communication component then needs to support a direct, localized exchange between mobile or stationary objects and mobile devices. We argue that this exchange can not occur over the Internet, as objects typically do not have an Internet address and even with Internet addresses available, publishing and looking up object addresses again requires a global database. A suitable communication mechanism should thus be ubiquitously available in the respective location context, incur minimal coordination, time, and communication overhead and, with regard to background usage, work without user interaction.

3. Direct Mobile Computer Vision

DMCV accounts for the aforementioned requirements in adapting CV approaches to ubiquitous mobile scenarios and implementing a suitable communication mechanism, respectively. To mitigate the dependency on Internet access and large databases, DMCV enables objects to provide their relevant CV data to mobile devices in wireless transmission range, inherently leveraging their locality and current mobility context. We first describe our design of object-centered and -provided CV data and then outline its embedding in a suitable communication component.

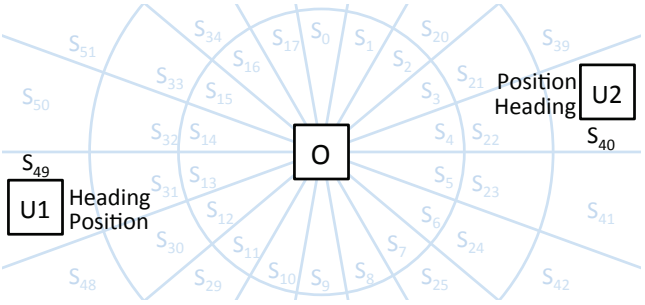


Figure 4: Objects (O) in DMCV know their location and heading. Given the position and heading of mobile users (U_i), they deliver CV data sets (S_i) matching each user's perspective and distance.

3.1 Computer Vision Component

Objects in DMCV hold the CV data that enables their detection in a given technique, e.g., via descriptor matching using ORB [19]. However, an important feature of database-driven image recognition is the ability to match images that show objects from different perspectives to derive the *pose* of the mobile user, i.e., her location and orientation towards the object. Especially, this enables recognition of objects even if the provided perspective has no exact match in the database. Removing database comparisons in favor of direct CV provision then requires explicit matching of user perspectives to appropriate CV data that enables recognition of the object from the current perspective.

DMCV thus makes use of the location context of both the mobile user and the object as illustrated in Figure 4. Each object thus knows its location and heading and holds sets of CV data (S_i), e.g., ORB [19], SURF [5], or FREAK [1] descriptors in our prototype, that cumulatively cover all perspectives a user can have towards the object. One set S_i thereby represents one or more perspective(s). The location of an object or a mobile device is thereby given by its GPS position when outdoors or by relative indoor positioning techniques, e.g., *WiFiSLAM* [11] or *SpinLoc* [20]. DMCV is thus able to account for the specific characteristics of CV approaches, such as the lack of scale invariance in ORB [19], and represent the user distance towards the object (e.g., S_{32} vs. S_{14}). Objects initially assign a heading to CV data sets to facilitate comparison with user locations. Stationary objects thereby assign the true global heading, while mobile objects, e.g., persons in mobile social networking, assign a "local" heading to CV sets, e.g., straight north is assigned to the set representing the front of the body, and adjust this heading when their global heading changes, i.e., when they rotate. An object that receives a CV data request that carries the user's location is then able to deliver the appropriate set by matching the location against the stored sets. In the example illustrated in Figure 4, user U1 requires CV data set S_{49} to detect the object O from her position, while the perspective of user U2 is represented in set S_{40}. We now illustrate how

this design realizes the aforementioned usage scenarios and discuss content creation in DMCV.

3.1.1 Computer Vision Usage Scenarios in DMCV

In order to enable **background** collection of CV data, a mobile user periodically broadcasts her location. Using this location information, objects in transmission range determine whether the user is able to view them and, if so, transmit their appropriate CV data set. Objects "follow" the user's mobility by responding with appropriate sets to updated location information, replacing the previous CV data for this object, if present. CV data sets may also feature border location indicators, outside of which the object is no longer visible, to enable timely deletion of obsolete CV data. Once the user scans the surroundings with the phone camera, DMCV detects objects using the available CV data sets.

For **selective** detection, the user actively broadcasts a designated request message that again contains her location. Objects, that overhear the request, respond with their location, a human-readable description, and an identifier for subsequent requests. The user may then select an object and directly request its CV data (plus its location and heading information). For example, searching for a specific vendor space when roaming the exhibition, the user may select this object, if available, and scan her surroundings to highlight it once it appears in the camera view. While a background search might also accomplish this, selective detection saves communication, computation, and energy resources by only detecting a single object based on semantics.

Last, a **directed** detection, i.e., recognizing an unknown object in front of the user, requires the user to provide her heading in addition to her location. Thusly informed of the user's perspective, objects that are out of this scope do not respond, saving the communication overhead of selective and background detection.

3.1.2 Content Creation in DMCV

To equip objects with CV data sets, object owners need to extract this CV data for a sufficient number of perspectives, i.e., possible views of the object. We argue that this task i) boils down to taking photos of the object, as extracting CV data from images can be encapsulated in an application that builds on the OpenCV library [6], and ii) directly incorporates changes in the appearance of the object, e.g., seasonal decorations, in contrast to obsolete comparison content in global databases. However, a 360° coverage of an object at a user position granularity of 1°, would require 360 distinct data sets per distance to cover every perspective. We exemplarily evaluate the required number of sets, i.e., the overhead of content creation, in Section 4.2, finding that the actual number is significantly lower.

3.2 Communication Component

Transmitting CV data between objects and mobile users over the Internet suffers from the difficulties of representing objects in the Internet and does not mitigate network dependency. We hence implement local wireless transmission of CV information between object and mobile device using Bluetooth, 802.11, and Beacon Stuffing [7]. In this, we strive to comprehensively explore the design space of local wireless communication available to current mobile devices, in order to contribute a notion of ubiquity afforded by each mechanism. Specifically, operating an 802.11 network at each object and transmitting CV data in this network incurs the time and management overhead of associating to the respective networks prior to obtaining CV data. Similar, Bluetooth devices need to discover other devices in range and connect to them. While both mechanisms present a feasible approach that is supported by mobile devices, this overhead might impede both spontaneous, ubiquitous CV data exchange, especially under client mobility, as well as the desired real-time character of mobile CV. In contrast, *network-less*, interactive Beacon Stuffing [7] enables objects to overload 802.11 management frames with payload to transport information without the requirement of a prior network association. However, current devices do not support the extraction of payload information transported in thusly modified 802.11 frames, reducing the compatibility and real-world applicability of DMCV.

In all mechanisms, mobile devices in DMCV transport location and heading information in wireless frames leveraging the inherently limited interaction scope of wireless communication to naturally control the number of approached objects. In Beacon Stuffing, Probe Request (PREQ) frames contain requests for CV data and are addressed to a pre-defined SSID, e.g., `base64(sha1(DMCV))` Objects then encapsulate their responses in 802.11 Probe Response (PRES)[2] frames carrying CV data or, in selective detection, the object description and an object-specific SSID for further requests. To receive responses, mobile devices perform 802.11 scans.

DMCV hence implements mobile and direct transmission of CV information in the envisioned usage scenarios in self-contained, low-overhead communication approach in the **background**, upon a **selective** trigger, or **directed** at an object. In comparison to Internet-based approaches, we leverage the near-instant nature of wireless communication to cater to the time-critical aspects of mobile CV. Furthermore, the natural restriction of CV data exchange to the local context, i.e., the communication range of wireless communication, directly pairs providers and consumers. In Beacon Stuffing, eliminating the need for network associations allows mobile devices to simultaneously collect the CV data of multiple objects in the background.

4. EVALUATION

We implemented the mobile device functionality of DMCV on a rooted Apple iPhone 4 (to realize Beacon Stuffing) and commodity iPhone 5S, running iOS 7.0.3, as well as Nexus 5 Android phones. The CV component of DMCV is realized by adapting the OpenCV library [6] (version 2.4.7), while we make use of `ioctl` to communicate with the 802.11 subsystem in order to enable Beacon Stuffing. An Asus Eee PC 1000HE running Ubuntu Desktop 12.04 implements object functionality for (multiple) objects in our evaluation. In Beacon Stuffing, the device listens to PREQs that transport location information and subsequently transmits the according CV data in PRES frames using a custom Python implementation that approximates 802.11 AP functionality in Beacon Stuffing [7]. In traditional 802.11 and Bluetooth communication, we discover and transmit CV data directly between the mobile devices. Figure 5 shows an example view of the mobile application in selective detection

[2]We refrain from using Beacon frames as each Beacon is interpreted as an 802.11 network, cluttering the observed 802.11 network landscape [7].

(a) Scan in selective detection (iPhone).

(b) Feature point detection, best seen in color (Android).

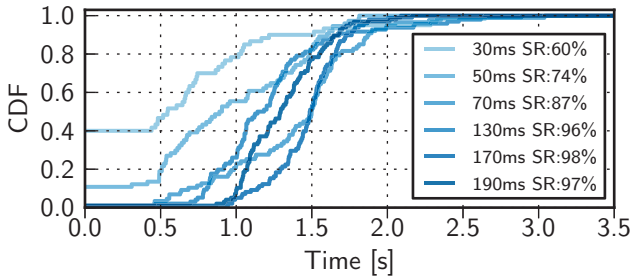
(c) Object detection and annotation (iPhone).

Figure 5: Selective detection menu and feature point detection as well as object detection and annotation in the camera view.

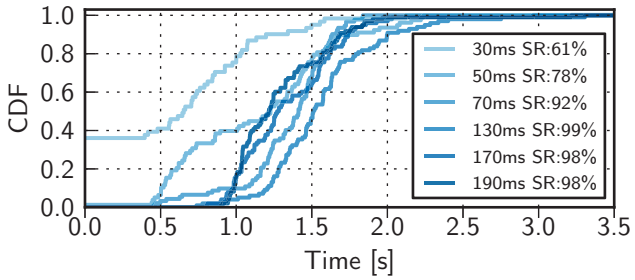
based on object description (5(a)). Furthermore, the figure shows screen shots of feature point detection at objects (5(b)) and actual object recognition (5(c)).

We implement DMCV for ORB descriptors [19], because of their small size per descriptor and computationally-efficient key point and descriptor calculation on a single image, as well as FREAK [1], and SURF [5] for comparison. The actual CV technique is exchangeable according to the respective implementation and usage scenario. The choice of CV techniques thereby induces a tradeoff between features, e.g., rotation or scale invariance, or the technique, the space requirements of descriptors, and the preparation of CV data at the object. For example, our choice of ORB descriptors requires a distance-wise partitioning of CV data sets (cf. Figure 1) because ORB is not scale invariant.

In this section, we first evaluate the performance of both the communication and CV component of DMCV. In this, we measure the time overhead and, in the case of Beacon Stuffing, the success rate of communicating CV data sets. To assess the real-world performance of the resulting system, we then measure the frame rate of DMCV when detecting multiple objects in the camera view. Then, we evaluate the



(a) Isolated scenario.



(b) Populated 802.11 scenario.

Figure 6: CDF of CV data transmission time and success rate (SR) in Beacon Stuffing over the 802.11 scan time in isolated and populated 802.11 scenarios.

number of angles an object is required to hold and offer for a comprehensive detection by the user. We thereby quantify the effort of creating the CV data for an object as well as the precision needed to derive the correct CV data set for user locations.

4.1 Performance

In this section, we first evaluate the performance of transmitting CV data in our extended Beacon Stuffing [7] mechanism as well as in 802.11 and Bluetooth. Then, we measure the usability of mobile devices in detecting (multiple) objects in the camera view using the received CV data sets.

4.1.1 Communication Performance

In the communication component, we emphasize ubiquitous and timely exchange of CV data between objects and mobile devices. To realize this, we adapt and extend Beacon Stuffing [7] to a bidirectional exchange of data within 802.11 PREQ and PRES frames. In detail, mobile devices query for CV data via PREQ frames and objects respond with their CV data in corresponding PRES frames. Owing to the intricacies and closed source of the iOS 802.11 subsystem, mobile devices fail to scan the designated 802.11 channel continuously until they receive the full CV data set. Therefore, mobile devices receive the CV data set by repeatedly scanning the 802.11 channel for a given time. Note that the mobile device starts scanning as soon as the request is sent, as it has no means of calculating the objects time requirement for CV data preparation and delivery. Furthermore, we do not assume synchronized clocks to model a real-world setting between foreign devices. As such, scan times and sending cycles are not synchronized, inducing the risk of missed frames.

We thus evaluate the overall time overhead and success rate (SR) of transmitting 117 ORB descriptors, i.e., 3 802.11 PRES frames, 100 times each for increasing scan durations.

The number of descriptors is thereby motivated by our previous results, as shown in Figure 2. Figure 6 show the measurement results for an isolated 802.11 scenario in a cellar room (6(a)) and an office scenario with substantial 802.11 background activity (6(b)). Specifically, the CDF distributions show the time overhead of successful transmissions out of all 100 attempts; the fraction of successful transmissions is indicated by the respective success rate (SR). We treat the isolated scenario results as a baseline, as we expect few side effects to influence our measurement, and view the populated scenario results as approximating real-world performance. Please note that we only had a single iPhone 4 available and will evaluate simultaneous requests and CV provision to multiple devices in future work.

Both scenarios allow similar success rates, supporting our design of transmitting CV data in robust 802.11 management frames. Perhaps surprisingly, success rates in the populated office scenario constantly are above the success rates in the isolated scenario, although the difference is 5 % at the most. The results further indicate that low scan times, e.g., 30 ms and 50 ms, hurt the success rate as responses may not arrive in this short time frame. In contrast, higher scan times, e.g., 130 ms or 190 ms, allow high success rates at the cost of marginally higher overall timings (2 s). The high overall time, relative to single scan times, is due to the integration of our Beacon Stuffing implementation in the iOS 802.11 functionality, inducing wait times and OS overhead.

The implementation of Beacon Stuffing in PREQ and PRES frames is a momentary design decision, as sending Beacon frames would "spam" the wireless medium with advertisements for non-existing wireless networks. A solution to this would be a modification of the 802.11 subsystem to not interpret Beacon frames that start with a designated prefix as network advertisements. This way, bidirectional communication in Beacon frames would not interfere with 802.11 networks and could be received by passive scanning, significantly reducing the time overhead of DMCV by avoiding unsynchronized scan and send intervals. However, we are currently unaware of such a method and thus implemented DMCV in an "802.11-friendly" manner. The general design of DMCV using Beacon Stuffing is agnostic to the 802.11 frames being used and may be adapted.

Overall, the majority of CV data transmission takes less than 2 s, still allowing for timely delivery in mobile scenarios. Furthermore, populated 802.11 scenarios do not harm the transmission, motivating real-world applicability of DMCV. Note that, while the results only describe the reception of one set of CV data for clarity, reception of multiple responses in scan time slots is possible, reducing the cumulative time.

To put these results in perspective and evaluate DMCV within more real-world applicable communication mechanisms, we evaluate the time overhead of obtaining CV data within 802.11 and Bluetooth networks. To this end, mobile devices discover and associate to a well-known network in the respective mechanism. The higher afforded data rates, in comparison to Beacon Stuffing, thereby allow the transmission of larger CV data sets, such as 5120 Byte when transmitting 80 descriptors in FREAK [1] and 20480 Byte for 80 SURF [5] descriptors. Figure 7 shows the associated time overhead as measured on a Nexus 5 device. Note that the time does not include association times, which in Bluetooth amount to 0.76 s on average and to about 1 s in 802.11.

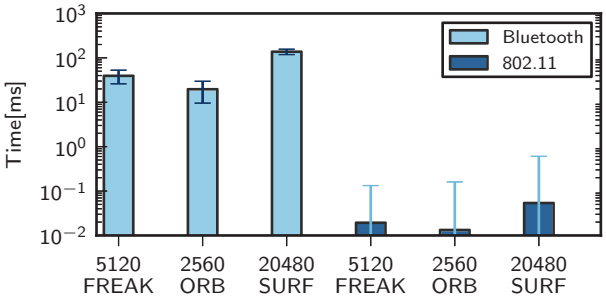


Figure 7: Transmission time overhead for ORB [19], FREAK [1], and SURF [5] CV data and their respective size (in Byte) in 802.11 and Bluetooth communication. Note the logarithmic scale.

802.11 and Bluetooth thereby amortize the increased association overhead and effort of selecting a network by acknowledged transmissions and higher data rates, requiring significantly less time to transmit CV data sets than in Beacon Stuffing, even for larger descriptor sizes. In contrast, communication within 802.11 and Bluetooth networks is limited to the devices associated to this network, in contrast to the unrestricted communication scope of Beacon Stuffing. Beacon Stuffing thereby allows to receive CV data from all surrounding devices, increasing the ubiquity of discovery and DMCV at the cost of unreliable transmissions.

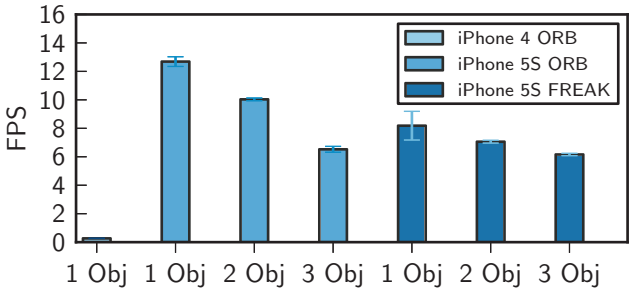
4.1.2 CV Performance

In the mobile CV component of DMCV, we emphasize the ability to detect multiple objects in the camera view, based on the sets of CV data previously received in the background. Namely, the background mode represents the computationally most expensive usage scenario because the detection process can not be restricted to a single object.

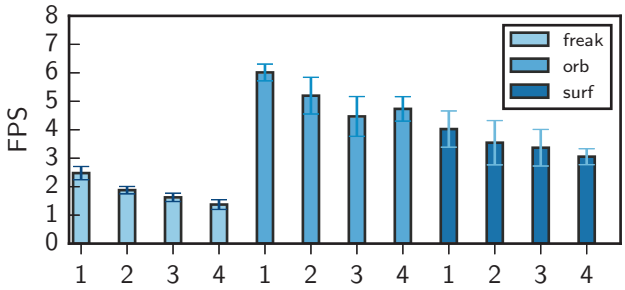
As an evaluation metric for the computational complexity of recognizing objects in DMCV, we measure the number of CV-processed and annotated frames per second (FPS), that can be displayed by the mobile device, over the number of detected objects. In detail, the number of FPS captured by the phone camera is constant, measuring the number of frames displayed after passing the CV pipeline then gives a measure of both the computational effort, the usability of the system, and the eventual computational limitations.

Figures 5(b) and 5(c) shows our measurement setup with three distinct objects, with 80 descriptors of CV data for all three objects. To derive a meaningful number of displayable FPS, we measured the frame rate over a duration of 60 sec.

Figure 8 shows the average and standard deviation of our measurement results for both the iPhone 4 and Nexus 5. We first follow our basic design of detecting objects using only 80 ORB [19] descriptors per object in a 640x480 pixel camera frame. Notably, the measurements on the iPhone 4 development device hinted at an excessive computational complexity for this device as already the detection of a single object allowed only 0.25 FPS. Detection of two objects did not allow meaningful measurement values. Given the relatively high age of the iPhone 4, we then measured the detection performance using an iPhone 5S to gain an estimate for current and future devices. For this device, continuously detecting one, two, and three objects in the camera view afforded frame rates of about 13 FPS, 10 FPS, and 7 FPS,



(a) iPhone 4 and 5S performance.



(b) Nexus 5 performance.

Figure 8: Maximum frame rates with CV detection on iPhones 4 and 5S as well as Nexus 5 Android device over the number of detected objects using ORB [19], FREAK [1], and SURF [5] descriptors.

respectively, providing an interruption-free usability. We consciously trade computation speed and thus displayed frame rates for a camera frame resolution of 640x480. This is because lower resolutions of displayed frames would speed up the processing time but would harm the detection quality and user experience. From these results, we deduce the feasibility or our approach within the communication overhead of transmitting the respective CV data set, as actually detecting objects does not induce further time overhead. Current smartphone devices appear well-equipped to compute the detection of even multiple objects simultaneously, affording discovery of CV data in the background and subsequent comprehensive detection.

To assess the dependency of these results on the given CV technique, namely ORB descriptors in our case, we additionally evaluated the CV detection performance in the same scenario when using FREAK descriptors [1]. FREAK is a slightly more recent binary descriptor that was especially designed with object recognition on mobile devices with lower computational and memory capabilities. In comparison to 32 Byte ORB descriptors, FREAK descriptors require 64 Byte per descriptor, i.e., using FREAK induces an increased communication overhead. In turn, the authors' comparison of the computational complexity [1] promises a decreased detection time overhead, offering a tradeoff between communication and computation times when choosing between ORB and FREAK. However, in our evaluation, detection of one, two, and three objects resulted in frame rates of about 8 FPS, 7 FPS, and 6 FPS, respectively. While this is a significant performance decrease in comparison to our ORB evaluation, using FREAK descriptors still affords a usable system performance. Furthermore, the performance difference may be due to implementation specifics in the open-source OpenCV library. In general, the results highlight the flexibility of DMCV with regard to the used CV technique, supporting the incorporation of future advances in CV techniques.

In comparison to the iPhone devices, the Nexus 5 Android shows lower FPS results, even for well-supported SURF descriptors. Again, ORB allows a higher FPS by virtue of its low computational complexity while SURF enjoys higher frame rates than FREAK.

4.2 Content Creation Overhead

In current mobile CV approaches that rely on central databases, content creation is a challenge for multiple reasons [4]. For example, appropriate, i.e., user-friendly, mechanisms for users to specify, annotate, and register their content at a service are missing. Furthermore, GPS information may be inaccurate since it is measured at the location of the user, instead of the object to be recognized. In DMCV, we strive to alleviate these shortcomings by enabling users to locally and flexibly create the CV information they want to provide. In our current prototype implementation, our design only requires the user to take a number of photos of the object, specify the location and perspective of these photos as well as the location of the object, and provide the annotation data.

Per our design, DMCV enables objects to hold their respective CV and annotation information and to provide it directly to mobile users. In this, the key requirement is to create and provide appropriate CV data sets for the possible user positions towards the object, i.e., the information that enables the respective CV approach to detect the object in the camera view (cf. Figure 4). The effort of content creation and provision thereby depends on the angular granularity of the CV data sets that is required to provide continuous object recognition from all possible user positions towards the object. Namely, 360° detection coverage of an object would, for a granularity of 1°, require 360 distinct CV data sets, inducing excessive effort of creating this information. Furthermore, matching data sets to user positions with 1° granularity might exceed the localization accuracy of smartphone GPS sensors and indoor localization techniques, which typically show an error margin of ±5-10 m.

In this section, we thus evaluate the required number of distinct CV data sets that enable continuous recognition. We perform this evaluation for ORB descriptors to model a worst case scenario, as ORB descriptors constitute the least capable, i.e., expressive, descriptor in our prototype implementation. Also, to model the most challenging case with regard to the diversity of object features that need to be represented in CV information, we attach two large LED monitor cases to each other in a 90° angle, as at the corner of a building. The print on each case is non-repetitive and distinct from the other case (apart from the manufacturer logo), modeling an object with distinctly different faces. We argue that this setup represents the most challenging case because CV information can not be reused within and across faces and because the 90° angle induces the sharpest possible break between faces. Also, we only use two faces, allowing a 180° view, since the remaining two faces, i.e., 180°, only repeat the settings encountered in two faces, i.e., the corner and plain faces.

For the actual measurement, we fixate the object and extract 80 ORB [19] descriptors of CV information when

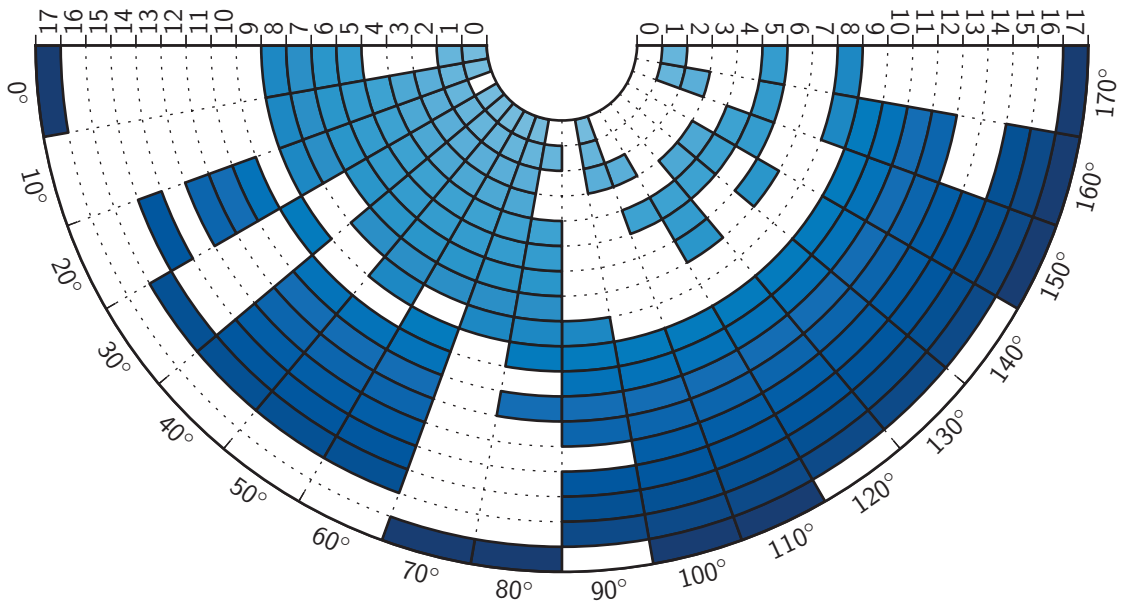


Figure 9: Detection success of CV data sets $S_i, i \in \{0, \ldots, 17\}$ with 10° granularity at all 10°-separated measurement points in an 180° view of the object. Detection (filled box) requires the object to be recognized in 95 % of processed frames. The set {S_6, S_9, S_{17}} allows detection of the object from all measurement points.

looking straight at the object in 10° steps, i.e., 18 distinct CV data sets. We then evaluate the recognition rate over a duration of 20 sec for all 18 sets at all 18 positions by measuring the number of frames in which the object is detected over the number of total frames. To gather a large number of result frames, we again use the iPhone 5S with a frame rate of 13 FPS. We treat a detection at a position as successful when object detection occurs in 95 % of the reported frames.

Figure 9 sketches the measurement setup, with the object in the upper center, and shows the detection success (filled box) of each CV data set $S_i, \{0 \leq i \leq 17\}$, at each measurement point, i.e., user position. Please note that the plot sorts sets in a specific color for each set and in increasing distance from the center for visibility; we performed all measurements from the same distance towards the object. The figure shows a clear separation of the two faces, i.e., between $[0°, 90°)$ and $(90°, 180°]$, by the CV data sets that allow object recognition from the respective measurement points. Outliers, such as S_{17} providing successful detection at measurement point 0°, are due to the aforementioned manufacturer logo being present in both faces. In each face, however, multiple sets allow detection at a high number of or all measurement points, e.g., S_6 and S_{12}, due to the rotation robustness of the ORB descriptor. From this, we deduce the need for a rather low number of sets to cover all user positions around the object, e.g., the set {S_6, S_9, S_{17}} would suffice for this 180° example. Furthermore, a low number of sets reduces the precision required from the position information as reported by mobile phones.

In real-world scenarios, objects such as buildings show highly repetitive features, such as the design of windows or facades. This motivates the assumption that, depending on the object, a rather low number of user positions need to be represented in CV data sets. A CV technique that lacks scale invariance, such as ORB, requires this number for each provided distance (cf. Figure 4). Similar, face recognition, e.g., to enable mobile social networking between smartphone users, will only require three CV data sets, one for each side of the face plus one for the front. Given that the creation of CV data basically requires the user to take a photo of the object from the respective view point, we thus argue that DMCV facilitates content creation and provision with reasonable *autonomous* overhead.

5. RELATED WORK

Numerous commercial and academic approaches [9, 12, 14, 17] enable mobile CV by processing user inputs, e.g., motion estimates or recorded images, within a dedicated server appliance that holds a representation of the respective scene or object. Within varying degrees, the costly comparison and detection steps are thereby performed on the server to accelerate the process and save resources of the mobile phone. As outlined in [4], this approach to mobile CV entails multiple shortcomings, such as the dependency on network connectivity and pre-established databases. In DMCV, we specifically depart from this design in favor of a distributed and lightweight approach.

Similar to our incorporation of location and heading information, Arth et al. [2, 3] propose leveraging the position of the user for in-scene localization, with emphasis on accuracy and speed. While targeting vision-based localization, in contrast to CV in DMCV, the proposed approaches still rely on feature databases holding a (3D point) reconstruction of a given area, a dependency we eliminate in DMCV.

Lim et al. [16] strive for real-time localization on mobile devices by removing client-server interaction. Instead, mobile devices hold an 3D reconstruction of the scene that was computed offline and match the 2D feature from the camera feed to 3D points in the model. To compute the model, they propose to use a micro-arial vehicle, highlighting the required effort, a characteristic that we aim to alleviate in DMCV to facilitate content creation.

Recently, Wang et al. proposed *InSight* [21], a method to recognize persons while removing the need for face recognition. Visual fingerprints, constructed from the current clothing and motion pattern of a user, allow mobile devices to recognize a person and display associated application data. *InSight* thereby presents an addition to CV data representations that could be transmitted in DMCV. Indeed, DMCV provides a communication system that enables the local distribution of visual fingerprints as envisioned (but not discussed) in *InSight*.

6. CONCLUSION

CV on mobile devices promises a ubiquitous enrichment of everyday scenarios by object information and annotations as well as applications that build on CV approaches, such as AR or mobile gaming. However, the applicability of mobile CV and the creation of CV content is hampered by the requirement of (global) large-scale databases that hold representations of the respective objects for comparison [4]. DMCV, in contrast, facilitates CV through transmission of the necessary CV information directly between mobile devices and recognizable objects. We seamlessly integrate DMCV into the 802.11 and Bluetooth as well as Beacon Stuffing of PREQ/PRES mechanism to establish a ubiquitous communication channel that allows mobile devices to discover recognizable objects and their CV information.

Our implementation for the iPhone 4 and Nexus 5 shows the real-world applicability of DMCV. Our evaluation shows both an affordable time overhead in the communication component as well as a high dependence on the computational capabilities of the mobile device in the CV component. Notably, the capabilities of our development device did not afford a usable mobile CV application when recognizing a single object whereas state-of-the-art devices, such as the iPhone 5S, allow fluent detection and tracking of multiple objects simultaneously. Last, creation of CV information that enables object recognition does not induce excessive overhead, as measured by the number of distinct CV data sets, but only requires CV data for a small number of selected user positions.

6.1 Discussion

DMCV meets the challenges and reduces the costs of providing a centralized, database-driven CV approach and the associated processing infrastructure with a localized technique that "crowdsources" the capabilities of existing mobile and stationary wireless devices. Local creation and provision of CV information then requires an integrated, user-friendly solution that i) builds sufficient CV information from a small number of images and the object's location and ii) offers an interface for registering application content with CV information. We argue that localized provision scopes and the ability to retain full control of the provided content serve as user incentives for using DMCV, in contrast to handing user position data, CV information, image material, and application content over to centralized CV providers.

6.2 Future Work

Future work will extend the descriptor matching design of DMCV by further CV techniques such as space-efficient feature descriptors [8] to assess the resulting design space and the associated CV performance and communication requirements. Current efforts to optimize CV techniques for the GPUs of mobile phones [15] promise a performance increase with regard to the computational effort of object detection in the future. Leveraging such efforts might further reduce the processing time on mobile devices, allowing for a simultaneous tracking of a large number of objects.

7. REFERENCES

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast Retina Keypoint. In *CVPR*, 2012.

[2] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg. Real-time self-localization from panoramic images on mobile devices. In *ISMAR*, 2011.

[3] C. Arth, A. Mulloni, and D. Schmalstieg. Exploiting sensors on mobile phones to improve wide-area localization. In *ICPR*, 2012.

[4] C. Arth and D. Schmalstieg. Challenges of Large-Scale Augmented Reality on Smartphones. In *ISMAR*, 2011.

[5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[6] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'reilly, 2008.

[7] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman. Beacon-stuffing: Wi-fi without associations. In *HotMobile*, 2007.

[8] V. Chandrasekhar, Y. Reznik, G. Takacs, D. M. Chen, S. S. Tsai, R. Grzeszczuk, and B. Girod. Compressing feature sets with digital search trees. In *ICCV Workshops*, 2011.

[9] D. M. Chen, S. S. Tsai, R. Vedantham, R. Grzeszczuk, and B. Girod. Streaming mobile augmented reality on mobile phones. In *ISMAR*, 2009.

[10] Computer Vision Group, ETH Zürich. Zurich Buildings Database. `http://www.vision.ee.ethz.ch/datasets/index.en.html`.

[11] Forbes. Apple + WiFiSLAM = Game on for Indoor Location. `http://www.forbes.com/sites/forrester/2013/03/29/apple-wifislam-game-on-for-indoor-location/`.

[12] S. Gammeter, A. Gassmann, L. Bossard, T. Quack, and L. Van Gool. Server-side object recognition and client-side object tracking for mobile augmented reality. In *CVPRW*, 2010.

[13] GOOGLE. Google Glass. `http://www.google.com/glass/start/`.

[14] H. Hile, R. Grzeszczuk, A. Liu, R. Vedantham, J. Košecka, and G. Borriello. Landmark-based pedestrian navigation with enhanced spatial reasoning. In *Pervasive Computing*, pages 59–76. 2009.

[15] B. Larson. An open source iOS framework for GPU-based image and video processing. `https://github.com/BradLarson/GPUImage`.

[16] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. In *CVPR*, 2012.

[17] NOKIA. LiveSight: immersive experiences you can act on. `http://conversations.nokia.com/2012/11/13/livesight-immersive-experiences-you-can-act-on/`.

[18] A. Opelt and A. Pinz. Image database for object categorization experiments. `http://www.emt.tugraz.at/~pinz/data/GRAZ_01/`.

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011.

[20] S. Sen, R. R. Choudhury, and S. Nelakuditi. Spinloc: Spin once to know your location. In *HotMobile*, 2012.

[21] H. Wang, X. Bao, R. R. Choudhury, and S. Nelakuditi. InSight: recognizing humans without face recognition. In *HotMobile*, 2013.

[22] H. Wirtz, J. Rüth, T. Zimmermann, M. Ceriotti, and K. Wehrle. A Wireless Service Overlay for Ubiquitous Mobile Multimedia Sensing and Interaction. In *MMSys*, 2014.