Overview of Mobile Augmented Reality

Jae Kyun Kim
Software Engineering Student
Jkkimsin_1@hotmail.com

ABSTRACT

Augmented reality is a relatively new field but also one with tremendous potential. Research towards this technology is becoming increasingly prevalent as people are beginning to acknowledge its usefulness. The idea behind augmented reality systems was to help people in some way by generating data from the environment to interact with the users. However, it faces numerous issues which do not make the systems feasible for use in practical situations. This paper focuses on the mobile aspect of augmented reality. As such, a lot of the problem comes from the hardware level. Devices such as mobile phones simply do not have the computational power necessary to provide the same level of processing as desktops. They have come up with some solutions ([6].[2]) such as removing features to create a more lightweight system and establishing a connection with a desktop for computation. What we really care about though is the impact these systems would have on existing problems. While taking into account the limitations we face with such systems, we want to provide a solution using augmented reality to enhance the users experience in some way. There have been an increasing number of interests being sparked from the art community ([4],[5]) as such systems are developed for museums and also as educational tools that help people learn. Most of the problems faced when designing systems for real world situations that go unnoticed are usability problems. There is a lacking number of research that is done for the usability interface of augmented reality. For this reason there are no usability guidelines when developing augmented reality systems. We need to make this issue well known so that people will become aware of it and address them.

1 INTRODUCTION

People are becoming more aware of concepts such as augmented reality even if they do not know the term. This is because they are shown in movies as cool and innovative technology that helps people process information in the real world. However the question that needs to be raised is if it is feasible for augmented reality systems to have a large impact in society. That is, can these systems be integrated into people's lives in such a way that they will provide useful solutions to existing problems in the real world by overcoming its own limitations?

Real-time detection and tracking is necessary and one way is using 6DOF natural feature tracking. This tracking is implemented using three feature descriptors, SIFT, Ferns and a template-matching based tracker called PatchTracker. The descriptors are modified to take into account the severe limitations of mobile phones such as slow memory, tiny caches, low throughput and limited storage. However, a realistic augmented reality system also requires exact tracking of target objects. This makes it a necessity for frameworks that allow the system to be scalable to the number of objects being augmented and provide a real-time accurate visual tracking method. This is done by setting up a server to hold the database while a mobile phone is used to retrieve and send data. It would also be ideal for mobile phones to allow collaborative augmented reality systems. They provide excellent multimedia services and have wireless network capabilities that provide a natural platform for CAR systems. However, because there are a large variety of mobile phones it would be necessary to have a performance characterization based on the different models. We will take a look at the Android and iOS operating systems and see how they compare based on problems associated with their CAR applications.

Now that we have introduced some of the technology associated with mobile augmented reality, we want to see if it is practical for such augmented reality applications to be able to enhance some aspect of society. The interests of augmented reality systems are increasing around the art society. This is because it is naturally easy to track and identify art since they have unique features which are easy to distinguish visually. They are also kept indoors where the environment is more controlled and friendly for augmented reality systems. The Louvre – DNP Museum Lab is a three year project and one of their presentations includes an augmented reality system for visitors. We will take a look at how effective this system is at navigating the visitors around the museum and providing information for the artwork. It is also important to note the benefits that can be provided to people through educational tools that are AR based. We will take a look at the usefulness of an augmented reality system that detects where the user is looking and provide information to where their gaze is at in the artwork.

Finally, we want to see the current limitations of augmented reality systems in outdoor use. For this we will use HMD to complete a series of gesture-based tasks while navigating to a target location. The HMDs themselves will be connected to a smart phone which will provide the computing power required to process information. Through this, we will have a look at some usability issues that are neglected in current augmented reality research and the results for the study.

2 PROBLEMS AND METHODOLOGIES

To have a clear picture of the state of mobile augmented reality, we will look at the technology of mobile phones, their usefulness when integrated in society and some usability issues that need to be addressed for such systems. In each section, there will be results on research that was done around the particular area to provide a better picture of the limitations and possibilities of mobile augmented reality.

2.1 MOBILE AUGMENTED REALITY TECHNOLOGY

Before research in mobile augmented reality became popular, it was always assumed that there was enough computational power for fully utilizing heavy vision algorithms. However, because of the lower capabilities of mobile phones compared to desktops (despite the improvements with smart phones), the vision algorithms are not as readily available for mobile augmented reality. It is also particularly difficult to provide a system that allows a scalable number of augmented objects as this require a large amount of memory. The design of the system should also take into consideration the different strengths and weaknesses of various models of mobile phones.

To provide real time detection tracking[6], SIFT and Ferns were both modified to be suitable for low-end embedded platforms. Some of the techniques from the original descriptors were replaced by different approaches. The resulting tracking techniques are called PhonySIFT and PhonyFerns. The final application is the result of merging the techniques of PhonySift, PhonyFerns and PatchTracker. SIFT is known to be strong but computationally expensive. Ferns on the other hand had quick classifications but required large amounts of memory. By merging descriptors with different strengths and weaknesses, the idea was to create a lightweight system that can run efficiently despite the limitations of mobile phones. The combination of the three descriptors provides a system with four major steps that make up the pipeline. It is feature detection, feature description and matching, outlier removal and pose estimation. The PatchTracker takes care of tracking once an object is identified until the target is lost and must be redetected. Both PhonySIFT and PhonyFerns are in charge of feature detection and pose estimation.. They use a FAST corner detector as well as Gauss-Newton iteration to detect feature points in the camera image and refine the pose estimated from a homograph.

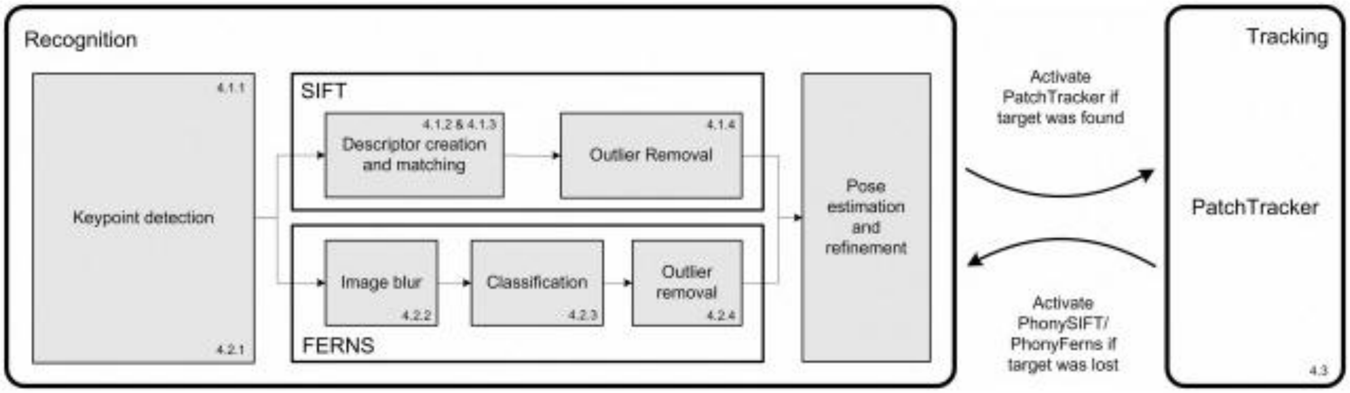


Fig. 1. State chart of combining the PhonySIFT/PhonyFerns trackers and the PatchTracker. The numbers indicate the sections in which the respective techniques are described.

PhonySIFT uses FAST corner detector for feature detection as it is one of the fastest detectors that provide high repeatability. The original SIFT uses DoGs for a scale-space search of features but this was too resource intensive and not suitable for real-time execution on mobile phones. The replacement does not estimate a feature’s scale so scale estimation is needed. This is done by storing feature descriptors from all meaningful scales. The same feature is described multiple times over different scales and this means that we can avoid a CPU-intensive scale space search by trading memory for speed. This approach does not use much memory and so is an ideal replacement for feature detection in mobile phones. Its descriptor creation and matching are also modified such that the techniques they use are feasible for mobile phones. The creation involves decreasing the 4x4 subregions with eight gradient bins each (128 elements) to 3x3 subregions with four bins each (36 elements). This modification only gives a 10 percent decrease in performance. It also requires outlier removal before doing pose estimation.

PhonyFerns also has its feature detection replaced by the FAST detector. The original authors have a simple code template for the runtime classification which makes it easy to setup. However, the original work used parameters for Fern sizes that required the database to have memory up to 32MB. This is not feasible for mobile phones so the database was modified to hold up to 2MB. PhonyFerns needs to remove outliers that are returned by the classification. It uses two steps to remove the outliers first of which is to use orientation estimated for each interest point and compute difference to the stored orientation of the matched model point. The differences are binned in a histogram and the peaks in the histogram are detected. All the matches in bins with less than a fraction of the peaks are removed since the differences should agree across inlier matches. The second step involves using a PROSAC scheme to estimate a homograph between the model points of the planar target and the input image.

PatchTracker tries to be more efficient than SIFT and Ferns when it comes to tracking. It only uses active search and based on a motion model, it can estimate exactly what to look for, where to find it, and what locally affine transformation to expect. This means that for each frame that contains the target, it tries to predict where the target will be in the next frame by using the previous frames. Since it only uses the reference image as the only data source, it doesn't need any preparations such as key point descriptions. Rather, the key points are detected during initialization using a corner detector.

PatchTracker cannot initialize or reinitialize by itself as it requires a previously known coarse pose. This means that by combining it with the PhonyFerns and PhonySIFT trackers, it is possible to create a system that makes up for the weaknesses of the individual descriptors by using the strengths of others. PhonySIFT and PhonyFerns trackers are used just for initialization and reinitialization. Once a target is detected and estimation with a valid pose is made, the PatchTracker uses the existing pose estimation to calculate poses from frame to frame for continuous tracking.

However, there needs to be a way to scale[2] the number of objects being augmented for the system to be practical. A scalable recognition module is used for the server while the mobile phone is in charge of tracking. The delay from the phone having to communicate through a WIFI network is minimal and only takes around 0.2 seconds for a cold start of the AR service. This framework is able to provide a stable performance for up to 1 million objects.

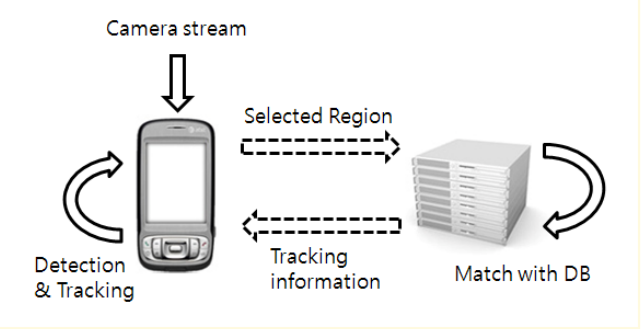


Fig. 2. Framework Overview

The image retrieval used for the server side is bag of visual words scheme also provides scalability. The vocabulary tree variant of the visual words scheme was used and this requires a quantization step which extracts representative data points called visual words from a large quantity of description vectors and is known as a very fast and accurate retrieval method.

For the mobile side, a modified version of SIFT was used. Ferns were not ideal for this situation as the descriptor was too large to be sent through the network. SIFT provided a much faster response rate for the framework overall. Just like in the previous research, the feature detection was replaced with a FAST detector on an image pyramid. Also the descriptor is modified so that instead of the usual 128 dimension, a configuration of 36 dimensions of 4 orientation bins and a 3x3 sub region descriptor was used. This approach is exactly the same as with natural feature tracking. For tracking, the Coarse-to fine matching is used and within this method, a frame-to frame tracking is used. This method is similar to that of what PatchTracker did. This shows that research that is done is carried on by others and is slowly evolving the technology.

Rather than focus more on the algorithms and their efficiencies, we will also take a look at some of the problems mobile phones can have depending on their operating system and specs[1]. This is an important factor to consider as the continuous development of mobile phones means that the variety of their hardware will only multiply. It is important to try to take advantage of this by taking it into consideration when developing an augmented reality system.

We first took a look at how to implement a system that has accurate natural feature tracking for the mobile phone. We expanded on this by also looking at adding a feature to allow a scalable number of augmented objects. This next paper concentrates on collaborative augmented reality systems. This lets multiple users share a real world environment and the data that is also modified by the augmented reality application. The main focus though is to see the different performances of the operating systems Android and iOS. Identical CAR applications will be implemented for the different systems so that it will be easier to compare the results.

Both applications have four stages. The first stage obtains the captured image from the camera. The second stage uses those images to detect the markers. The third stage uses the markers to draw a 3D object on the image. The fourth and final phase sends this information to other application nodes through some kind of broadcast. This is the collaboration part of the application.

The characterization results show that out of all the stages in the application, detecting the markers was the most time consuming. The rendering 3D objects stage is also decoupled on some devices. This process allows avoiding low refresh rate and facilitating the collaborative work. Such results and findings help to design efficient systems and applications. This practice should also be applied to other augmented reality systems as this would also potentially increase the efficiency of all mobile augmented reality systems.

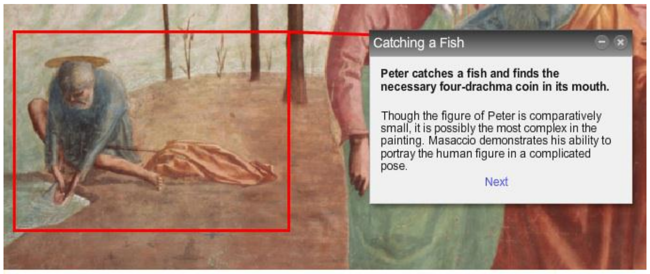
2.2 REAL WORLD USES OF AR TECHNOLOGY

Now that we have seen the scope of mobile augmented reality technology, we want to look at the feasibility of using such systems to provide an impact on society. Helping out the art community by using augmented reality systems and seeing how useful and practical they are will give us a good idea of this.

As mentioned previously, the Louvre – DNP Museum Lab[5] had a presentation for augmented reality systems. While most of the processing done for tracking and detection is done via a desktop processor connected to a portable device, this is still part of a mobile/portable augmented reality system.

The project focused on creating two main systems. The artwork appreciation system function would be designed in such a way as to explain the points of artwork appreciation directly in front of the showcase in a manner that was easy to understand and informative. The system would also need to operate under the various lighting conditions inside the museum. This condition is easier to follow than compared to having to worry about outdoor uses as the lighting inside the museum would be controlled.

The guidance system function would be compared in terms of efficiency and usability to a 2D-maps or pure audio descriptions for guiding visitors to the next point of interest. Previous results showed that only 9.7% of visitors acted correctly in compliance with the route guide when using static PDA screens and audio commentaries. The weight of the system is light (under 1 kg) and the battery provides enough power for one visit (1.5 hours maximum). The specifications of the hardware are reasonable for a normal visit to the museum. The equipment is not very heavy and very portable and the battery life is reasonable so people do not have to charge it multiple times during a visit.



Peter catches a fish and finds the necessary four-drachma coin in its mouth.

Though the figure of Peter is comparatively small, it is possibly the most complex in the painting. Masaccio demonstrates his ability to portray the human figure in a complicated pose.

Fig. 3. Web-browser based educational tool

People have used various methods to display additional information about an artwork. Current web-browser based educational tools can be used to create text pop-ups and rectangular outlines to highlight important information. However, this method usually degrades the overall experience by being distracting from the art itself. It is also particularly difficult to easily convey a synoptic narrative type of art. These types of paintings have multiple pictures crowded in one painting. The full story of the painting is conveyed by the sequence of the pictures, but it would be very difficult to decipher for those unfamiliar with the story.

For example, the painting in figure 1[4] depicts a scene where Jesus directs Peter to find a coin in the mouth of a fish in order to pay the temple tax. The best way to understand this would be to start at the center of the piece where the tax collector is demanding money. Peter is told by Jesus to take money from the mouth of a fish. By adjusting ones gaze to the left of the painting, viewers notice Peter executing Jesus' instruction. Finally, the viewers have to look at the far right of the painting to see Peter paying the tax collector. This seemingly complicated procedure of viewing art can be broken down into something that is much easier to understand.

The idea behind this augmented reality system is to track where the user is looking at in the painting. It will than blur out the remaining pictures in the painting and provide a description of what the user is viewing. This means that even without having much knowledge of how to properly view art, just by naturally looking around the painting, viewers will find information that is shown intuitively. This means that the information that is displayed will always be in a position that does not obstruct image features that are/will become important to the user. Also if the system could control the sequence in which the viewers can perceive art, by blurring unimportant areas, viewers will get a better understanding of the artwork.

2.3 USABILITY ISSUES IN AR

While there is a lot of research going behind augmented reality, users evaluations are still not recognized as being a common practice in this area. For this reason, there is a lack of understanding of human-computer interfaces and associated user requirements within real-world use.

There has not been much information about the use of gesture with wearable mobile AR interaction in an outdoor environment[3]. Even though there are numerous studies of input devices for AR systems including a wearable on-the-wrist smart phone, many of these studies were done in a controlled environment. This means there is no way to be sure if the same gestures would be effective in an outdoor environment.

To give more insight with these problems, research has been done in an outdoor scenario with HMD gear. The computation will be done by a smart phone connected to the HMD. The user interface will consist of a top-down egocentric 2D rangefinder on the top left of the screen. The target location will be represented as a white dot placed in a relative position and distance to the user. An objective target icon will also be displayed in the user's horizontal field of vision. The size of the icon will depend on the distance between the user and location.

The hand gestures implemented required the use of a black glove with coloured markers so as to increase accuracy. Users can use gestures to show details, activate an icon and hide details. The red marker on the hand acts as a cursor positioned relative to the HMD interface.

The participants were given a total of three tasks. They were chosen to determine the feasibility of using a HMD for a set of navigational procedures.. The first task required the participant to navigate their way to a location on the radar. As they approached the location, they would receive visual updates on the HMD. These include updates on the radar and where they have travelled so far. After they reach the first location, they are asked to move to another location. While navigating to the second location, they are given an update to change their destination to another set of location.

3 RESULTS AND EVALUATION

3.1 Performance for AR Systems

A frame server[6] was implemented to create comparable results for tracking quality. The server loads uncompressed raw images from the file system rather than from live camera view. The frame server and the three tracking approaches were ported to the mobile phone to compare the performance difference from the mobile phone and desktop.

Scale/Rotate "Boat"	Original SIFT	PhonySIFT without outlier removal	PhonySIFT with outlier removal	Original Ferns	PhonyFerns without outlier removal	PhonyFerns with outlier removal
Image 1	1970/1980 (99%)	316/1432 (22%)	262/319 (82%)	73/73 (100%)	128/192 (67%)	111/114 (97%)
Image 2	1596/1603 (100%)	400/1546 (26%)	192/200 (96%)	62/64 (97%)	106/183 (58%)	66/67 (99%)
Image 3	534/544 (98%)	298/1472 (20%)	273/295 (93%)	4/198 (2%)	25/169 (15%)	15/23 (65%)
Image 4	362/365 (99%)	244/1277 (19%)	220/222 (99%)	1/198 (1%)	27/173 (16%)	19/23 (83%)
Image 5	45/63 (71%)	89/1083 (8%)	31/36 (86%)	1/193 (1%)	10/166 (6%)	7/32 (22%)

Tilt "Graffiti"	Original SIFT	PhonySIFT without outlier removal	PhonySIFT with outlier removal	Original Ferns	PhonyFerns without outlier removal	PhonyFerns with outlier removal
Image 1	1017/1030 (99%)	410/1493 (27%)	331/332 (100%)	54/55 (98%)	153/195 (78%)	90/93 (97%)
Image 2	184/228 (81%)	256/1524 (17%)	199/201 (99%)	21/22 (95%)	77/179 (43%)	73/79 (92%)
Image 3	14/21 (67%)	58/936 (6%)	4/10 (40%)	12/13 (92%)	35/162 (22%)	22/27 (81%)
Image 4	0/14 (0%)	27/1109 (2%)	0/0 (0%)	3/199 (2%)	15/159 (9%)	14/21 (67%)
Image 5	0/9 (0%)	4/1232 (0%)	0/21 (0%)	1/194 (1%)	0/144 (0%)	0/11 (0%)

Brightness "Cars"	Original SIFT	PhonySIFT without outlier removal	PhonySIFT with outlier removal	Original Ferns	PhonyFerns without outlier removal	PhonyFerns with outlier removal
Image 1	1197/1212 (99%)	589/1515 (39%)	354/375 (94%)	124/126 (98%)	163/197 (83%)	112/117 (96%)
Image 2	895/910 (98%)	478/1410 (34%)	347/350 (99%)	116/117 (99%)	161/192 (84%)	120/121 (99%)
Image 3	651/661 (98%)	463/1449 (32%)	75/83 (90%)	102/104 (98%)	147/194 (76%)	101/106 (95%)
Image 4	503/514 (98%)	305/1043 (29%)	243/254 (96%)	89/91 (98%)	143/193 (74%)	104/108 (96%)
Image 5	336/349 (96%)	184/663 (28%)	151/158 (96%)	85/85 (100%)	136/190 (72%)	90/96 (94%)

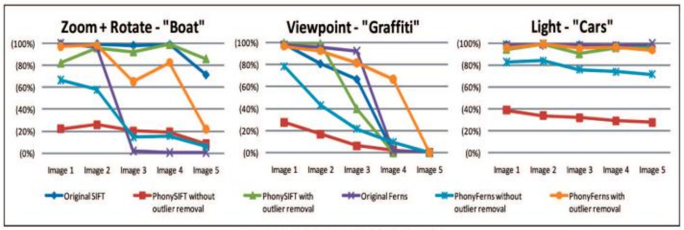


Fig. 5. Matching results for three image sets

To figure out the matching rates differences, a comparison was made with the modified descriptors and their original counter parts. The test was done on three data sets (Zoom+rotation, Viewpoint and Light). For the first data set, the original SIFT has very good results for the first 4 images while the 5th is decent. PhonyFerns shows a lot of improvements compared to its original counterpart when the outliers are removed. PhonySIFT itself isn't doing too badly and as much more consistent results than PhonyFerns with its outliers removed.

The tests show that the outlier rates of the modified methods are much higher than those of the original approaches. But even if there are high numbers of outliers, they can be successfully filtered by using the outlier removal techniques. For this reason, we could say that the modified approaches work at similar performance levels to that of the originals.

The experiment[2] for scalable number of augmented objects was carried out by the Android Nexus phone. It had a 1GHz snapdragon processor and its camera had an image size of 320x240 pixels, with frames around 20Hz. The database contained pictures of 10k music CD covers and 10k corresponding music video scenes.

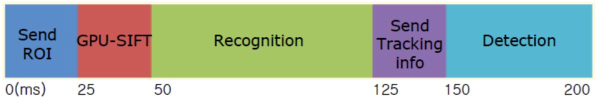


Fig. 6. Time band of initial start time

Figure 6 shows the initial start time interval. The recognition on a 10k database was up to 100ms. The time it took to receive tracking information was 25ms and the detection time was 50ms. The pure network overhead is 50ms which is almost the same as the time taken to process detection for one frame. However, the influence from an external network condition is not expected to affect the system much as sending/receiving data size up to 80kb only happens during user initialization.

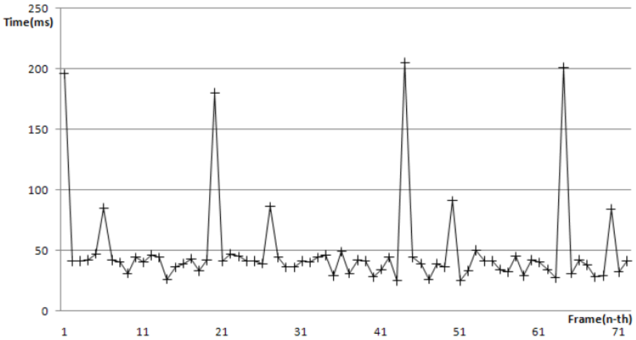
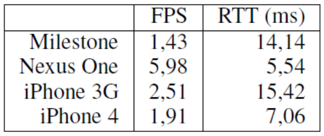


Fig. 7. Overall performance time

In figure 7, the horizontal axis is the number of frames and the vertical axis is the time spent on that frame in ms. The time for processing the frame is increased by nearly 200ms for every reinitilization of the AR service. Since the reinitilization does not happen frequently, the overall real-time performance of the system is not hurt too much. Without reinitilization, the average processing frame rate for 71 frames with 23.9 Hz which is faster than the mobile camera capture speed of 20 Hz.

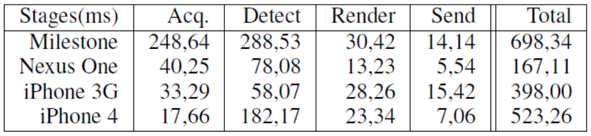
For the performance evaluation of the CAR system[1], two tables show the throughput and execution time for each smart phone. The throughput is measured in terms of frames per second (FPS) and the system latency uses the round-trip-time (RTT) for each message sent to the server. The Milestone and Nexus phones use the android operating systems. In table 1, Milestone has the poorest throughput, and the Nexus is the one with the best throughput. iPhone 3G has a better throughput than iPhone 4, however this is because of the fact that the iPhone 3G has to analyze less amounts of data than the iPhone 4. This is because the latest iPhone versions did not have an option to use lower resolutions of images.

Table 1: Throughput (in terms of FPS) and RTT for each smartphone



	FPS	RTT (ms)
Milestone	1,43	14,14
Nexus One	5,98	5,54
iPhone 3G	2,51	15,42
iPhone 4	1,91	7,06

Table 2: Execution times (ms) for each considered mobile phone



Stages(ms)	Acq.	Detect	Render	Send	Total
Milestone	248,64	288,53	30,42	14,14	698,34
Nexus One	40,25	78,08	13,23	5,54	167,11
iPhone 3G	33,29	58,07	28,26	15,42	398,00
iPhone 4	17,66	182,17	23,34	7,06	523,26

Fig. 8

Milestone has the worst performance out of all the other phones. It takes much longer in obtaining images and detecting the markers. The image acquisition of the iPhone 4 is much faster than that of iPhone 3G but it has much slower marker detection. As mentioned previously, the images that are processed by the iPhone 4 are much higher quality than those of iPhone 3G.

The results show that the most time-consuming stage in a CAR application is the marker detection stage, followed by the image acquisition stage, the rendering stage, and the transmission stage. It also shows that the best throughput, measured in FPS is obtained for Android devices. Part of the reason is that in the Android implementations, there is a separate thread that performs the rendering stage and this is not the case for iOS-based devices.

3.2 Visitor Surveys on System

One thing to consider about the visitors that participated in testing[5] the artwork appreciation system function and guidance system function is that they do not necessarily know what augmented reality is. This means that if they needed any sort of technological knowledge of the device to operate it, it would make the whole system impractical and useless.

The participants were interviewed and asked what they thought about the devices they used. The augmentations of the illustrations helped them understand every important detail about a particular artwork. They liked that there was good synchronization between audio and computer graphics and found it helpful. It also motivated them to examine the artwork in greater detail and truly appreciate the finer details. The only issue they found was that it was difficult to move their gaze from the AR system's monitor to the real artwork while holding the device with both hands. A hands free approach would definitely benefit in this situation.

The participants were impressed with the route guidance system. They said that it was an experience that could not be recreated elsewhere. The users were surprised to see 3D computer graphics emerging on the screen. They were mostly unaware of the term "AR" but understood immediately how to use the device. However there were problems as users were unsure of what AR points were. They were supposed to arrive on each checkpoint and gather more information for the next station. They were unsure how and where AR was really provided and some thought that the device itself was not working sometimes.

A lot of existing mobile AR devices do not take into account gauging where the users attention is focused and leveraging that information for the placement and delivery of AR elements. There is huge potential in exploring these areas as we could for example, gracefully degrade regions of the image that are not important at specific times. It is also possible to interact with the users in such a way as to direct their gaze at a particular important image feature by using techniques based on where they are looking.

3.3 Usability issue tests

A big problem in the experiment[3] to discover some usability issues was directly related to the external environment and the HMD gear. The HMD had little effect on improving visibility even with though it could manually adjust the brightness. Since it was difficult to see the display clearly, this affected the task performance. It was also easy for them to miss incoming visual alerts in the AR view. The biggest problem people faced was that, while wearing their devices, they had worse situational awareness. The device hindered their peripheral vision and this prevented them from noticing oncoming traffic or red lights when approaching pedestrian crossings. There is definitely some safety implications for anyone wearing a HMD device in busy urban areas.

Participants often had to shield the camera from the sunlight as it prevented them from seeing the user interface. This meant that it was very difficult to use hand gestures to navigate around the interface as they had one hand blocking the camera. Despite these problems, the icons and the actual navigation themselves were very easy to understand and operate.

4 CONCLUSION

The data that was produced from the technological aspect of mobile augmented reality systems is promising and insightful. The modifications done on existing descriptors give their mobile counterparts similar performance with minimum sacrifices. This means that it is certainly feasible to have a responsive 6DOF natural feature tracking with the current hardware level of mobile phones. Using a server to take into account the low levels of memory in mobile phones is a good solution for a scalable database of augmented objects. The delay when communicating between the client and server is minimal and does not affect the system. These technologies will be particularly useful when designing an augmented reality system that provides navigation and information in artwork.

Museums expect a device that is small and portable which the mobile phone fits this role perfectly. It should also scale when new artwork is added onto the gallery. Most importantly, the system needs to quickly and accurately identify each artwork. While using eye tracking technology is certainly useful to determine the important regions of a painting, it is not within the scope of this paper. There are also other methods of determining importance such as the order in which the image features should be shown so that viewers are able to get the full story behind the painting. As seen from this paper, the participants received augmented reality technology rather well in a museum context. While not without problems, they were excited at the prospects of this new technology.

It is much harder to provide a system that is feasible in an outdoor area. A lot of the limitations in such a system are beyond the technology of mobile augmented reality. Problems such as screen glare and the way you interact with the system are usability issues that need to be addressed more. By exploring the different ways to interact with AR systems, there may be more convenient ways of interacting with the navigation systems or telling a system which area in the painting they want information about. It is also just as important to have a full understanding of the limitations of specific models of mobile phones and base the design off of this information. This will help in creating much more efficient and powerful systems.

REFERENCES

1. Fern, V., #225, et al. (2011). Performance Characterization on Mobile Phones for Collaborative Augmented Reality (CAR) Applications. Proceedings of the 2011 IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications, IEEE Computer Society: 52-53.

   http://dl.acm.org.ezproxy.auckland.ac.nz/citation.cfm?id=2065092.2065161&coll=DL&dl=ACM&CFID=94889778&CFTOKEN=16549250

2. Ha, J., K. Cho, et al. (2010). Scalable recognition and tracking for mobile augmented reality. Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry. Seoul, South Korea, ACM: 155-160.

   http://dl.acm.org.ezproxy.auckland.ac.nz/citation.cfm?id=1900179.1900213&coll=DL&dl=ACM&CFID=94889778&CFTOKEN=16549250

3. Kerr, S. J., M. D. Rice, et al. (2011). Wearable mobile augmented reality: evaluating outdoor user experience. Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry. Hong Kong, China, ACM: 209-216.

   http://dl.acm.org.ezproxy.auckland.ac.nz/citation.cfm?id=2087756.2087786&coll=DL&dl=ACM&CFID=94889778&CFTOKEN=16549250

4. McNamara, A. M. (2011). Enhancing art history education through mobile augmented reality. Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry. Hong Kong, China, ACM: 507-512.

5. http://dl.acm.org.ezproxy.auckland.ac.nz/citation.cfm?id=2087756.2087853&coll=DL&dl=ACM&CFID=94889778&CFTOKEN=16549250

6. Miyashita, T., P. Meier, et al. (2008). An Augmented Reality museum guide. Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE Computer Society: 103-106.

   http://dl.acm.org.ezproxy.auckland.ac.nz/citation.cfm?id=1605298.1605355&coll=DL&dl=ACM&CFID=94889778&CFTOKEN=16549250

7. Wagner, D., G. Reitmayr, et al. (2010). "Real-Time Detection and Tracking for Augmented Reality on Mobile Phones." Visualization and Computer Graphics, IEEE Transactions on 16(3): 355-368.

   http://ieeexplore.ieee.org.ezproxy.auckland.ac.nz/xpl/articleDetails.jsp?tp=&arnumber=5226627&contentType=Journals+%26+Magazines&searchField%3DSearch_All%26queryText%3Dreal-time+detection+and+tracking+for+augmented+reality+on+mobile+phones