Investigating Image Processing Algorithms for Navigating Cultural Heritage Spaces using Mobile Devices

Ayodeji Olojede[1] , Hussein Suleman[2]

Department of Computer Science,
University of Cape Town
Rondebosch, Cape Town, 7700, South Africa
Oljayo001@myuct.ac.za , hussein@cs.uct.ac.za[2]
,

Abstract. The use of mobile devices is increasing in the cultural heritage and museum context. The most common approach is to provide a customized mobile device to the museum visitor to navigate museum spaces. In this paper, a mobile cultural heritage guide is presented, which enables image based navigation of rock art sites using computer vision and image processing algorithms for rock art image feature detection and extraction. Traditionally such systems have used algorithms such as Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF) and Oriented Fast and Rotational Brief (ORB). The three algorithms have been integrated in a prototype and their performance has been evaluated. It was observed that digital recognition of rock art images is possible under certain image preprocessing conditions. Also, evaluation result shows that, generally, SIFT has good accuracy and, when used in conjunction with K Nearest Neighbour matching has acceptable matching speed.

Keywords: Cultural Heritage, Rock Art Images, Content Based Image Retrieval System

1 Introduction

Cultural Heritage is a composition of physical artifacts, cultures and attributes from the past that a nation considers significant and decides to pass on to current and future generations. Examples include cultural artifacts such as rock art. Rock art is a term used to describe human-made engravings and paintings on stones [1]. Commonly associated with a nation's rich archaeological past, rock art represents an important asset for tourism and is also found at the centre of culture and education. A survey by Euro barometer shows that cultural heritage objects are often isolated and difficult to appreciate, thereby resulting to low user engagement [2]. This has generated research interest all over the world in seeking ways to make cultural heritage more desirable and more accessible. A novel approach that is still currently being researched is the use of smart mobile phones of the user, by providing a suitable mobile application that can be downloaded onto the user device. Leveraging on this approach and with emphasis on rock art, this research is focused on investigating the feasibility of using images to navigate cultural heritage spaces like the rock art sites using computer vision image processing and matching algorithms on mobile devices. To assist with this investigation, a Content Based Image Retrieval System was developed. Most heritage (rock art) sites are often left in their historical context found many miles away from civilization; the ambience of the site context is lost if the cultural heritage artifact is removed from original location [3]. The proposed system will enable the camera of a mobile device to act as a cultural heritage guide such that the user points the camera of his mobile device at the rock art of interest and takes a picture. Computer vision image processing technology recognizes the input picture and provides a ranked list of results to the user. Details such as title and description of each returned result can be easily communicated back to the user. Such an application could assist users to appreciate rock art and also make it more accessible. In the first part of this paper, we introduce cultural heritage and the issue of user engagement with cultural heritage. We then propose a mobile application that may help to tackle the problem. The other sections of this paper discuss the related work, the research approach, a discussion of the selected algorithms for feature extraction, descriptors and matching. Further the paper presents an experiment to assess and compare the performance of the selected algorithms under realistic conditions.

1.1 Problem Definition

User engagement with culture has been a problem that has persisted for many years. This is further confirmed through a recent survey that was done by Euro barometer in 2013 on cultural access and participation [2]. Although several attempts have been made in trying to address this problem, especially with the current advances in mobile technology that have given rise to a number of mobile applications for personalized cultural heritage content delivery, but very little work has been done in addressing this problem from a rock art context. Literature has shown that digital recognition of rock art images is difficult due to their cluttered and rough nature, which makes it difficult for most computer vision algorithms to process them.

1.2 Related Works

Many years ago, personalized content delivery in the heritage and museum context was only achieved through audio-guides, where an audio system is used to guide users in experiencing an exhibit in a museum or a heritage site [4]. Current advances in mobile digital technology has given rise to development of many mobile applications to support personalized delivery of multimedia content to the user.

In CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) rock [1], feature detection and extraction steps are outsourced to the user as the authors argued that current image feature detection and extraction algorithms performed poorly on rock art images. In captcha rock, the user takes a picture of a petroglyph and traces out the object of interest and submits to the application. The application extract features and does similarity checks with a database of features for finding corresponding matches. Mobile Vision [5] was designed for tourism in an urban setting. A user takes a picture of a place in his line of sight and the application intelligently returns all tourism objects around the picture taken. The application makes use of GPS to detect the user's location and Internet to return the tourism objects around the user's location. Map Snapper [6] was designed to allow users to query a remote information system based on photos of a paper map taken with the camera of a mobile device. The information system could then return useful information to the user via the device. For example, the returned information could include such things as events, facilities, opening times, and accommodation in the geographical region depicted by the query

1.3 Description of System

Heritage vision is a mobile application developed with simplicity and personalized rock art multimedia content delivery in mind. The application does not depend on Internet connectivity to function. All image processing is done on the mobile device. Taking cognizance of the high computational requirements of image processing algorithms, this application was developed with consideration of speed and memory constraints of mobile devices. As illustrated in Figure 1 below, the mobile application enables the user to take a picture of the rock art of interest with a mobile device.

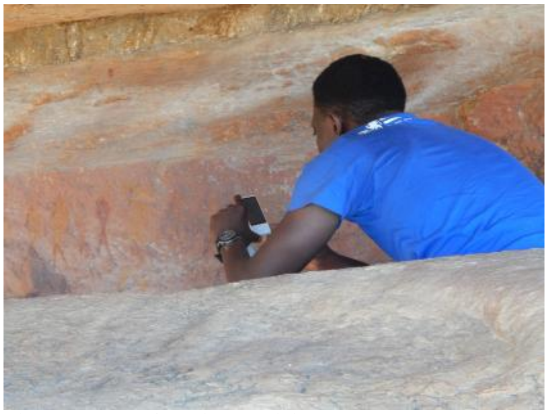

Fig 1. Usage of Heritage Vision

The picture is automatically processed by the image processing engine that runs on the mobile device. The image processing library intelligently pre-processes the image and detects distinct features and extracts descriptors These descriptors are then matched with the descriptors in the database of training images preloaded in the application. When the application finds matches based on a Euclidean distance calculation, a ranked list of results is displayed back to the user. Information such as title and description are communicated to the user.

2 Research Approach

2.1 Collection of Data

A combination of rock art images taken at a selected rock art site (cederberg region in Western Cape, South Africa) was used to create the set of training images. Query images for testing are pictures from the same rock art site taken by students with their mobile device. This was to ensure that experimentation is done based on a real life scenario. Please see figure 2 for sample data

Fig 2: Sample rock art images taken from the two sites at the Cederberg Region, Western Cape, South Africa

2.2 Mobile Application Prototype

A user centered design approach was adopted in the application prototype design. An operational prototype was implemented at this stage. Operational prototyping is a combination of a throwaway prototype and evolutionary prototype [7]. A throwaway prototype enables user engagement in the design process by producing diagrams of prototypes on paper for user evaluation. An evolutionary prototype is the actual system. In an evolutionary prototype, a clear set of requirements is developed while evaluation results from the throwaway prototype are used to complete the aspects of evolutionary prototype with an unclear set of requirements.

2.3 Image Preprocessing

The nature of rock art images makes it difficult for image processing algorithms to process them. Even the ones that eventually succeeded took a lot of time, hence the preprocessing step is necessary. Image processing involves importing, analyzing and manipulation of an image. This process helped in simplifying the image thereby making it easy to process the image further. The output of this stage is usually an enhanced or compressed image.

2.4 Image Feature Extraction and Description

In order to match images, features or region have to be detected and extracted for each image. Such a feature or region can be defined as an interesting part of the image. A single image can contain hundreds to thousands of features. The following algorithms were chosen due to their general performance.

SIFT (Scale invariant feature transform) is used for extracting distinctive invariant features from images that can be invariant to scale, rotation and to illumination [8]. SIFT uses a difference-of-Gaussian (DOG) function to identify potential interest points, which are invariant to scale and orientation.

SURF (speeded up robust features) was proposed at the ECCV 2006 conference in Graz, Austria [9]. Its purpose was to ensure high speed in three of the feature detection steps: detection, description and matching. SURF utilizes integral images for image convolutions and a fast hessian blob detector [10].

ORB (Oriented Fast and Rotational Brief) [11] was developed to be less computationally expensive than SIFT. It’s a combination of FAST (features from accelerated segment test) detectors and BRIEF (Binary Robust Independent Elementary Features) feature descriptors.

2.5 Feature Matching

The best candidate match for each feature was found by identifying its K nearest neighbours in the feature database of the training images. The nearest neighbours are defined as the key points with minimum Euclidean distance from the given descriptor vector [9]. This is a straight forward approach that linearly searches through the key points in no particular order.

3.0 Experiment

3.1 Experimental Apparatus

Image Dataset: As mentioned in the previous section, training images were gathered from the field. Images were taken with a 16mega pixel Samsung digital camera. Images were taken at different angles and orientation. All pictures were taken in daylight. As of the time of this experiment, we were able to gather over 500 images. Only 460 images were used as the set of training images. The remaining were rock art pictures taken with a mobile phone camera. This will be used as Query images for testing.

Hardware: This experiment is focused on developing the application on the Android platform. A mobile device with processing speed of at least 1GHZ and RAM of at least 1 Gigabyte is considered the minimum hardware requirement. The minimum Android version is 3.0. For this experiment, we have made use of HTC Desire 816 with a Snap Dragon processor and a memory capacity of 1.5GB.

Software: OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real time computer vision [12]. The library is free for use under the open source BSD license. The library is cross-platform. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android.

3.2 Experimental Process

In the first stage of the experimental process, we needed to archive the image dataset mentioned above. In this stage, features of the images were extracted and their descriptors stored in the database. Prior to this, it was important to pre-process the images as the un-refined form was very difficult for the algorithms to process. The algorithms that eventually made it through took over a minute to extract features. In the pre-processing stage,

1, Application normalizes the size of each image such that the final size is 288 x 216 pixels.

2, The image was split into R, G, and B channels. The Green channel was preferred as rock art paintings were more visible under this channel. For a visual reference, please see the image below. In the image below, the Green channel output is the circled section.

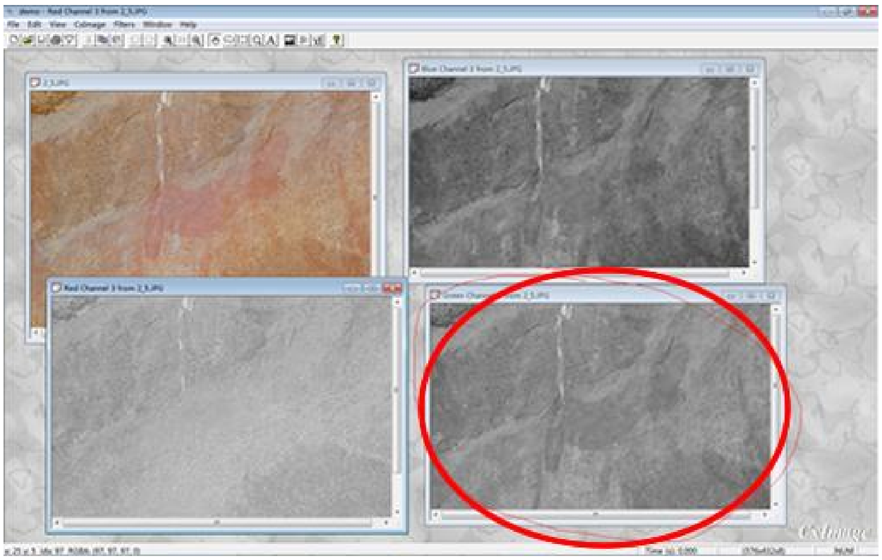

Fig 3. A view of the Original Image, R Channel, G Channel and B channel. The Green channel is circled.

3, The application performs a contrast enhancement (linear normalization) to [0, 255] as rock art images have low contrast.

The image in Figure 4 below shows final output of the image pre-processing

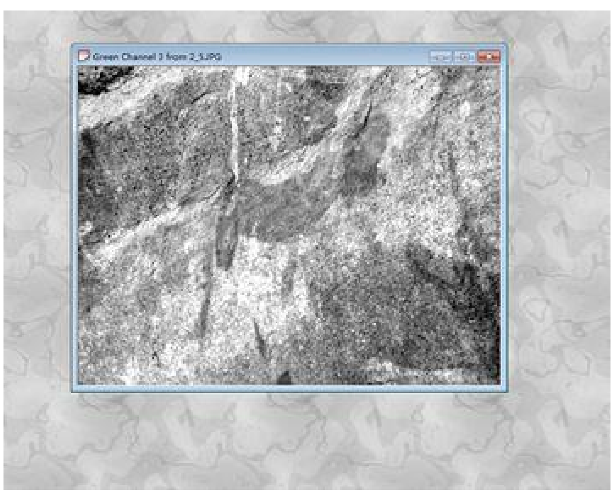

Fig 4: Result of linear normalization (contrast enhancement) of the green channel output

In the second stage, a separate apparatus for each algorithm was created. This process will enable us to visually quantify the difference between the numbers of features detected by each algorithm and also the quality of these features. It will also enable us to easily evaluate the performance of each algorithm. All apparatuses are similar in setup and image dataset. The feature detection method of each algorithm (SIFT, SURF & ORB) was executed first on the image dataset. Extracted features descriptors were stored in the database.

We then selected 5 random images from the image dataset set aside as query images. Remember these were images taken with a camera phone by students. There were no particular guidelines as to how they took the picture. It was taken at their own discretion. We ran the experiment with this image dataset on the set of query images in the database. As mentioned earlier, we used the K Nearest Neighbour matching method for establishing and sorting match results.

3.3 Experiment Results

After the experiment process, we discovered SURF detected more features on rock art than any of the other two algorithms. We set a default value for the amount of key point detection for both SIFT and ORB. SURF does not offer such an option. We did this because most of the key points detected have very low radius and most are just going to slow down the descriptor storage and matching process. The table below shows the results of the feature extraction and descriptor process. From a speed point of view, ORB is fast and we believe this is because it detected the least number of key point.

Table 1. Shows feature detection and descriptor comparison on a random query dataset. **Total Desc** represents the total number of descriptors and the **Time (ms)** represents total time taken in ms to extract descriptors

	SIFT		SURF		ORB	
Query	Total Desc	Time (ms)	Total Desc	Time(ms)	Total Desc	Time(ms)
Image 1	500	1552	421	691	387	103
Image 2	500	3709	1119	948	500	194
Image 3	500	2700	786	709	488	129
Image 4	500	3621	981	935	496	149
Image 5	500	5537	1535	1210	500	157

We also found out that the majority of the key-points were detected outside of the actual paintings. The key point concentration was more in the background region. Very few were detected around the paintings. We believe it will have a negative impact on the feature matching.

We also discovered that, when the painting region is cropped out from the background and submitted to the algorithms for processing, features were now detected around the painting region. We do not know the reason for this as of this time.

K Nearest Neighbour was adopted to compare key points. It's an exhaustive search and this was used because of its simplicity and, at the time of this experiment, there were 460 training images in the database. The method re-arranges the image results according to the confidence level. This method searches linearly and may be ideal for training images of single rock art sites but will not be ideal for a larger database comprising of training images of different rock art sites.

3.4 Evaluations.

We have evaluated the performance of the algorithms based on Precision (how many returned documents are relevant) and Recall (what fraction of the relevant document was found). We also calculated the average precision.

From the table below, it is clear that SIFT outperforms SURF and ORB. It has demonstrated competence in all the queries when used with K Nearest Neighbour Matching.

Table 2. Shows the precision and recall of the query set for each algorithm. It also shows the Average precision.

	SIFT		SURF		ORB	
Query	Precision	Recall	Precision	Recall	Precision	Recall
Image 1	0.35	0.8	0.15	0.37	0.1	0.25
Image 2	0.35	0.7	0.2	0.4	0.05	0.1
Image 3	0.3	0.35	0.2	0.22	0.5	0.58
Image 4	0.55	1	0.4	0.7	0.2	0.36
Image 5	0.55	0.3	0.25	0.14	0.4	0.23
Average Precision	**0.42**		**0.24**		**0.25**	

We also evaluated the application performance in terms of speed. From the result table below, the K Nearest Neighbour match spent more time matching features from SURF and this is evident in the feature descriptor result where SURF detected more key points than SIFT and ORB. SIFT descriptors produced more results that are relevant to the user query even when there are scale and rotational changes. ORB was the fastest but the majority of the returned results were irrelevant to the query.

Table 3. Total time taken for application to perform search

	SIFT	SURF	ORB
Query	Time to Match (ms)	Time to Match (ms)	Time to Match (ms)
Image 1	15472	20542	680
Image 2	15802	15674	676
Image 3	15184	20072	583
Image 4	16968	15875	661
Image 5	15136	22016	637

4.0 Conclusion

In this paper, we have been able to show the feasibility of digital recognition of rock art images under certain image pre-processing conditions. We also have been able to demonstrate the feasibility of object recognition on a mobile device with a particular configuration. However, further experiments will be required, most especially for the matching algorithm as K Nearest Neighbour match will not be ideal for a large image database.

References

1. Q. Zhu, X. Wang, E. Keogh, S.-H. Lee, An efficient and effective similarity measure to enable data mining of petroglyphs, Data Mining Knowl Disc (2011) 23:91-127.
2. Cultural Access and Participation. European Commission: Special Eurobarometer 399, April-May 2013.TNS OPINION & SOCIAL, Brussels [Producer].
3. Silver, Constance S. The Rock Art of Seminole Canyon State Historic Park: Deterioration and Prospects for Conservation, 1985. Report #4000-430 may be obtained from Texas Parks and Wildlife, 4200 Smith School Road, Austin, TX 78744
4. Tellis, C. & Proctor, N. (2002) Workshop: Handhelds in Museums, Museums and the Web, Boston, 2002
5. G Fritz, C Seifert, P Luley, L Paletta, A Almer (2004). "Mobile vision for ambient learning in urban environments", The third annual international mobile learning conference MLEARN 2004. Lake Bracciano, Rome, 5-6 July 2004, in-print
6. J Hare, P Lewis, L Gordon, G Hart (2008). "MapSnapper: engineering an efficient algorithm for matching images of maps from mobile phones", Proc. SPIE 6820, Multimedia Content Access: Algorithms and Systems II, 68200L (January 28, 2008); doi:10.1117/12.767207.
7. Alan M. Davis: Operational Prototyping: A new Development Approach. IEEE Software, September 1992. Page 71.
8. Lowe D. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Visions, Vol.60, No.2, and page numbers (91-110), ISBN 0920-5691
9. SURF: Speeded up robust features, http://www.vision.ee.ethz.ch/~surf/papers.html, Viewed 14 November 2014.
10. Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). Computer vision and image understanding, 110 (3):346{359, 2008.
11. Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski "ORB: an efficient alternative to SIFT or SURF", Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011
12. OpenCV Open Source Computer Vision, http:// http://opencv.org/, Viewed 11 January 2015