

# Context Photography on Camera Phones

Mattias Rost, Lalya Gaye, Maria Håkansson, Sara Ljungblad, Lars Erik Holmquist

Future Applications Lab, Viktoria Institute

Hörselgången 4, 417 56 Göteborg, Sweden

www.viktoria.se/fal

{rost, lalya, mariah, saral, leh}@viktoria.se

## ABSTRACT

Context photography uses sensors and image processing to create a picture that is visually affected by invisible factors in the environment, such as sound and movement. The system was previously implemented on a Tablet PC, but our newest prototype runs on standard camera phones. The program uses the phone's built-in microphone and real-time image analysis to create context photographs. Having the system on a mass-market platform will allow for large-scale user studies, since anyone with a compatible phone can now download and install the application.

## Keywords

context photography, sensors, image processing, prototyping for large-scale use

## INTRODUCTION

A context camera allows the user not only to take pictures of the visual scenery, but also to capture the context: sound and movement. We have implemented a context photography application on an ordinary camera phone, making the phone a pervasive prototype for exploring context photography. As opposed to other projects registering context by augmenting photographs with e.g. audio [1][2], the context camera opens up a new dimension of what can be captured on a photograph by using sensor-based data to directly influence the appearance of the resulting picture. For instance, in a noisy environment, images will get a certain visual effect whereas in a quiet environment the pictures will look different.

When taking a picture, sensor-based context information is reflected visually through graphical effects rendered in real time. A number of different graphical effects have been

implemented, using different mappings from context data to input parameters of the effects. The user can choose which effect he or she wants to use, and thus decide how the picture should be affected by the context, but the context determines the ultimate results.

After a first prototype implemented on a Tablet PC [3], our context camera prototype is now successfully ported to standard camera phones (see figure 1). This enables the use of such devices in large scale during user studies, since anyone with a camera phone can download the application. No extra equipment is needed. Besides, because users are already familiar with handling controls of their own devices, they can begin exploring the application immediately after installation.

## IMPLEMENTATION

The current context camera prototype implementation is programmed in C++ using the graphics library GapiDraw [5]. GapiDraw is a multi platform graphics library available for various handheld devices. That makes our implementation highly portable. The current implementation runs on the Nokia 6600 and Nokia 6630 phone, but can easily be ported to any system built on the Symbian Series 60 operating system. Furthermore, with a little more effort the application can be ported to any device supported by GapiDraw, given that it has enough computational power.

The application has a viewfinder, and a GUI. The GUI lets the user choose effect, calibrate the sound and motion sensitivity, browse a gallery of taken pictures, and also delete pictures. The calibration for each effect is done individually, so that the user can let one effect be affected strongly by the sound level, whereas another effect might not be effected at all.

By constantly capturing the video stream and sound input from the built-in camera and microphone, the input is processed to create context data. In addition to sensing sound volume, a Fourier Transform on the sound data computes power of low, medium, and high frequencies. The video stream is analyzed using algorithms from the image processing program Optica [4]. These algorithms compute various aspects of motion in the picture, such as a vector field depicting the motion at different points in the picture. These algorithms require heavy computations, and they had to be implemented in an efficient way and optimized to run smoothly on the camera phone.



Figure 1. Mobile phone with context photography.

The four graphical effects implemented are (see Figure 2, from left to right):

1. Traces of colored shadows follow the movement and the color of the shadows changes with the frequency components in the surrounding sounds.
2. The part of the picture with most movement is zoomed in, and rendered on top of the actual picture with the amount of transparency determined by surrounding sound volume.
3. Small white dots follow the movement as a decaying trace. The picture is pixelized with the size of the pixels determined by sound volume.
4. Movement makes the picture look like liquid. As in 3, the picture is pixelized with the size of the pixels determined by sound volume.

The camera API gives support for image retrieval in two ways: *video capture* and *image capture*. Image capture supports high resolution pictures whereas video capture only retrieves images in low resolution. Using the viewfinder, displayed images are retrieved using video capture. These frames are constantly sent to the image processing algorithms to compute motion data. As a picture is taken, image capture is used to get a picture in high resolution. To render the effects in high resolution, this motion data computed from the video stream has to be transformed to fit the high resolution picture. By doing calculations on low resolutions frames, and then transforming the calculated data, the application runs smoothly and the resulting picture is in high resolution.

The first prototype rendered the effects in real time and displayed them to the user. However, user evaluation showed that displaying the effects in real time before taking the picture affected the perception of the application [3]. Therefore in the next evaluation step the effects will not be visible until the picture is taken. However, they can still be rendered in real time to quickly show how the picture is affected by context for demonstration purposes.

#### USER EXPERIENCE

The current version of the context camera was first used in a college course about site-specific art and digital photography in San Francisco, USA. Experience from

workshops using an earlier prototype [3] showed that the way people interact with a context camera differs. A group of so called lomographers ([www.lomography.com](http://www.lomography.com)) searched for context that would change the pictures in an interesting way, whereas a group of high school students attempted to create contexts by e.g. screaming.

#### CONCLUSIONS AND FUTURE WORK

We have presented context photography implemented on camera phones. The next step is to use this prototype in a worldwide study to evaluate the context photography concept. Since the prototype is implemented on a commercial platform, people at different places and from different cultures can participate in the study. Prototyping with commercial products relieves the need to equip people with special hardware – anyone with the right device can just download the application and install it.

#### ACKNOWLEDGMENTS

We wish to thank Panajotis Mihalatos, for implementing the first prototype. This project is funded by the Swedish Foundation for Strategic Research through the *Mobile Services* project.

#### REFERENCES

1. Bitton, J. and Agamanolis, S. RAW: Conveying Minimally-Mediated Impressions of Everyday Life with an Audio-Photographic Tool. *Proc. of CHI'04*, ACM Press (2004).
2. Frohlich, D. & Tallyn, E., 1999. Audiophotography: Practice and Prospects. *Extended Abstracts of CHI'99*, ACM Press, 296-297.
3. Ljungblad, S., Håkansson, M, Gaye, L. & Holmquist L.E. 2004. Context Photography: Modifying the Digital Camera into a New Creative Tool. *Extended Abstracts of CHI'04*. ACM Press 1191-1194
4. Mihalatos, Panajotis. Optica image processing program. Available at <http://www.id.gu.se/arttech/project/dot/optica.htm>
5. Sanneblad, J. & Holmquist, L.E. The GapiDraw platform: High-performance Cross-platform Graphics on Mobile Devices. *Mobile and Ubiquitous Multimedia (MUM2004)*, College Park, Maryland, U.S.A. October 27-29 2004.

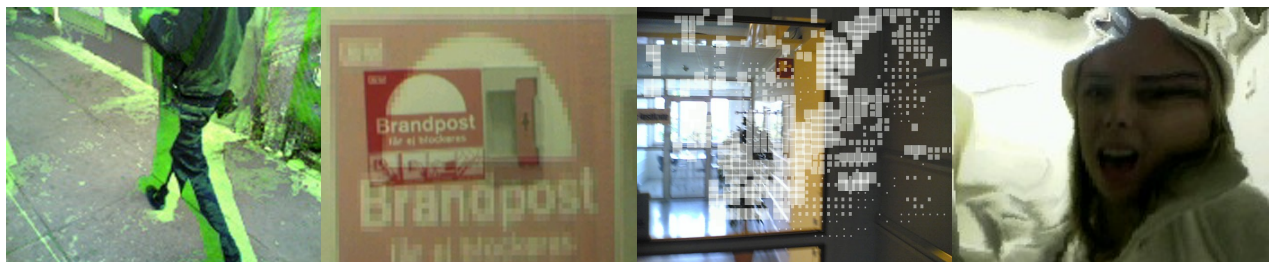


Figure 2. Pictures taken with context camera phone.