The Journal of Systems and Software 135 (2018) 205–215

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Increasing the capturing angle in print-cam robust watermarking

Anu Pramila*, Anja Keskinarkaus, Tapio Seppänen

Physiological Signal Analysis Team, Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland

ARTICLE INFO

Article history:
Received 3 February 2017
Revised 1 September 2017
Accepted 23 October 2017
Available online 5 November 2017

Keywords:
Focal stack optimization
Mobile phone application
Watermark application
Camera phone
Computational photography

ABSTRACT

It is nowadays more probable that a print media is captured and shared with a mobile phone than with a scanner. The reasons for photographing the print range from intention of copying the image to simply sharing an interesting add with friends. Watermarking offers a solution for carrying side information in the images, and if the watermarking method being used is robust to the print-cam process, the information can be read with a mobile phone camera. In this paper, we present a print-cam robust watermarking method that is also implemented on a mobile phone and evaluated with user tests. Especially, the lens focusing problem when the picture is captured in a wide angle with respect to the printout is addressed. The results show that the method is highly robust to capturing the watermark without errors in angles up to 60° with processing times that are acceptable for real-life applications.

1. Introduction

The aim of this paper is to present an algorithm for reading a watermark from a printed image with a mobile phone camera with wide angles of capture. To achieve this, three research fields are joined: print-cam robust image watermarking, mobile phone applications and computational photography.

The aim of the print-cam robust watermarking is to link the physical world with the digital without compromising the aesthetics of the artwork. The reasons for hiding information vary from increasing security to offering beneficial information to the recipient. However, in real world applications, several things make the print-cam robustness challenging. The watermark should be simultaneously robust to AD/DA transformations, rotations in 3D, scaling and translation. In addition, human interaction, JPEG distortions, lighting variations and camera related distortions, such as barrel distortions and focusing, need to be considered as well (Pramila et al., 2007).

In the scenario presented in Fig. 1, the watermarked image is located in a magazine laying on a table. The image could also be placed on a wall and people with different heights could take pictures of it from different angles or around a cluster of other people. Specifically, the lens might not focus correctly to the image taken at a large angle causing the watermark extraction to fail. It should be noted that the watermarked image is never published electronically and therefore intentional attacks against the watermark are not considered here.

Most of the proposed print-cam robust watermarking methods have been tested in relatively constricted settings and only a few of them have been proved to work on a mobile phone. Most of the methods assume that the user is able to point the camera straight in relation to the watermark and so the distortions are minimized.

The first print-cam robust watermarking method was proposed by Katayama et al. (2004). The method was based on a sinusoidal watermark pattern and a frame for synchronization. They noted that utilizing a frame is not a problem as it shows the user that a watermark is present.

Kim et al. (2006) relied on an autocorrelation function for watermark extraction. They embedded a pseudo-random vector by tiling it repeatedly and later detected the peaks that were formed by the autocorrelation function. The method was tested on a digital camera on a tripod and human interaction was required for the final extraction of the watermark to minimize geometrical distortions. Likewise, Pramila et al. (2012) used autocorrelation although they employed directed patterns. The message was encoded in the angle of these patterns and the method was robust to $\pm 20°$ of the tilt of the optical axis.

Yamada and Kamitani (2013) placed the information only in some locations in the image in order to preserve quality of the image. The watermark was embedded with spread spectrum techniques and read from the video feed by trying to find the watermark from each frame in turn. The method required the user to keep the camera near perpendicular to the watermark.

All these methods operate in the spatial domain and only a few have proposed methods in other domains. Delgado-Guillen et al. (2013) operated their method in log-polar transfor-

* Corresponding author.
E-mail address: anu.pramila@oulu.fi (A. Pramila).

https://doi.org/10.1016/j.jss.2017.10.029

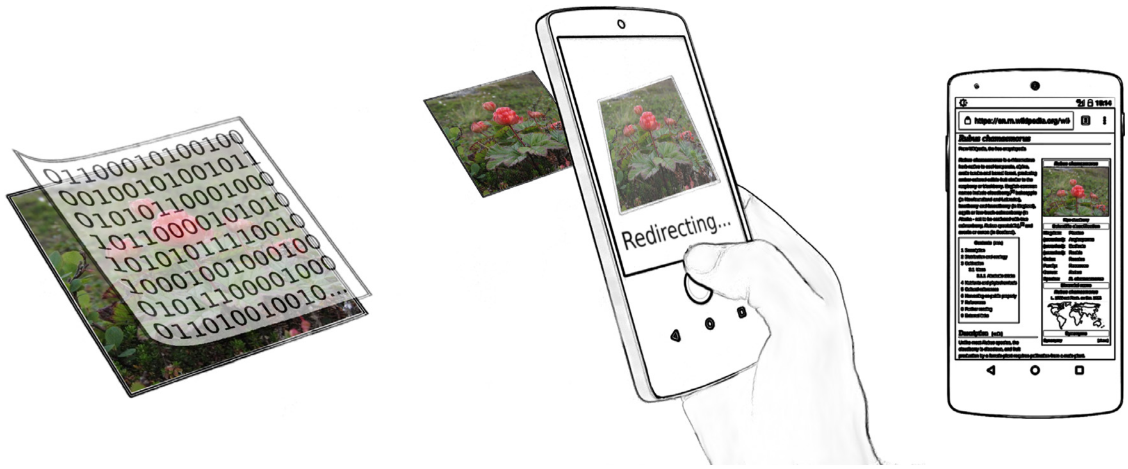

Fig. 1. The scenario. The watermark is embedded in an image. The watermark contains a link to a webpage that the user can read with his/her camera phone.

mations and Pramila et al. (2008) divided the method across several domains.

In addition, commercial applications have been launched. Most notably, Digimarc (2015) introduced an application called Discover. The aim of the application was to connect users from magazine pages to the Internet.

All the aforementioned print-cam robust watermarking algorithms rely on the autofocus feature of the camera. However, when the image is captured in a large angle, autofocus does not work. None of the methods consider the large angles at which depth of field of the camera comes into play and parts of the image are unfocused. Increasing the depth of field would solve the focusing issues and traditionally this is achieved by decreasing the size of aperture. However, this requires more lighting or longer exposure time, which might introduce distortions such as the motion blur. In addition, mobile devices often have a fixed aperture and changing the depth of field is not possible.

Computational photography refers to techniques for capturing and processing images digitally, therefore overcoming the limitations of traditional optics. The effect of increasing the depth of field can be achieved with computational photography by capturing a focal stack, a series of images focused at different depths, and building a new image with an extended depth of field. Several methods have been proposed, unfortunately, only a few of them take into account the movement between images that occurs when images are captured free handedly. None have been suggested to be used in combination with watermarking.

One of the most promising methods was proposed by Vaquero et al. (2011) who utilized FCam programmable-camera software stack (Adams et al., 2010). For better efficiency and results, they noted that not all of the images in the focal stack are required for building an all-in-focus image. Their method was based on sweeping the focus of the camera lens and selecting the optimal set of images to be taken according to small sharpness maps.

However, all the cameras cannot focus the lens on specific focal points, nor sweep the lens. Sakurikar and Narayanan (2014) approached this problem by using camera autofocus so that the viewfinder was divided into 16 blocks and autofocus algorithm was pointed towards each block in order. Overly similar images were removed before registration. Solh (2014) selected three predetermined focal distances and captured pictures at these lens locations. The frames were then aligned and fused. Zhang et al. (2013) captured the whole focal stack but removed the worst images from the stack. They took advantage of the IMU (Inertial Measurement Unit) to receive information about the movements of the phone, so that the sharp images could be discerned from the blurry images more easily.

Here, the print-cam robust watermarking is combined with the all-in-focus imaging into a system that is robust to severe 3D distortions. To solve the focusing issues, an algorithm is proposed that employs computational photography to improve the sharpness of the images for the watermark extraction.

The algorithm, denoted here as WCAM (Watermark CAMera), begins by collecting a focal stack that is scaled down to a fraction of a size for fast calculations. The number of images in the stack is then optimized by selecting a small amount of the images in the stack that contain most of the information about the scene. The selected images are registered and blended into an all-in-focus image. From this scaled-down all-in-focus image, an approximation of the correct watermark location and alignment in the original images can be determined.

The method takes advantage of our previous results on combining computational photography and robust watermarking (Pramila et al., 2017). An evolved algorithm is presented which is 35 times faster by taking better into account the inherent features of the selected watermarking method and demands of camera phones. The method is robust up to capturing angles of 60° with varying distances. In addition to testing capturing angles, the robustness is evaluated with user tests.

In Section 2, the embedding process of the watermark is explained. In Section 3, the implementation of the watermark extraction is presented and the required methodologies are explained. The method is implemented on a camera phone and tested with users. The results have been collected in Section 4, and finally Conclusion is given in Section 5.

2. Watermark embedding and printing

The scenario imposes requirements for a moderate amount of geometrical distortions that occur as the approximations of transformation matrices are utilized instead of actual matrices as well as medium capacity. The blind print-cam robust method by Pramila et al. (2012) fulfils most of the requirements. With few changes, mainly in extraction part, the method adapts well to the scenario.

The overview of the watermark embedding method is illustrated in Fig. 2. The watermark is embedded in the Y-channel of YCbCr transformation of the image by dividing the image into k blocks and embedding the watermark in the blocks. Unlike in Pramila et al. (2012) in which $k = 1 \dots 9$, in here $k = 1 \dots 16$ in order to increase the capacity. The size of the block depends, there-