A Single Pixel Camera based on a DLP Video Projector

Stefan Grönroos[1,2], Wictor Lund[2], and Jerker Björkqvist[2]

[1]Turku Centre for Computer Science (TUCS)
[2]Åbo Akademi University, Department of Computer Science Joukahaisenkatu 3-5A, 20520 Turku, Finland
{stefan.gronroos,wictor.lund,jerker.bjorkqvist}@abo.fi

KEY WORDS compressive sampling, camera, dlp

ABSTRACT

In this paper, we describe how a single pixel camera was constructed using widely available components, with the most important component being a repurposed video projector. A single pixel camera uses one single sensor, i.e. one pixel, to read light instensity. This single sensor samples different combinations of parts of the image multiple times, and uses that data to construct the final image. The image quality is primarily determined by the number of times this sampling process is performed, but also by the algorithm used to reconstruct the final image. Both the hardware and software components used for the build are described in this paper. We also discuss the limitations of our prototype, and how it could be improved given better hardware components, and/or improved software.

1 INTRODUCTION

A camera using a single pixel to aquire the image may seem like a strange idea at a time when most cameras use sensors consisting of millions of pixels. There are some possible advantages of such a camera though, especially when capturing light outside the visible spectrum — such as infrared imaging for example — where large sensor arrays may be quite expensive to make. In this case a single sensor may prove a lot more cost effective.

The single pixel camera concept has been explored by a research group at Rice University [7]. Our camera is very much inspired by the setup described in [7], and as such the theory of operation is well established.

In this article we will describe our prototype single-pixel camera built using a rather old high-definition (HD) video projector, some microcontrollers, a photodiode, and some miscellaneous circuits, as well as the software used to perform the rather heavy signal processing involved. The resulting initial prototype captures grayscale images with resolutions around 1000-4000 pixels, which is inferior to state of the art cameras, however we will also discuss planned improvements to the camera, as much higher resolution, and color images should be feasible using the same technique.

Digital cameras use image sensors consisting of millions of light sensors, or pixels. Once the shutter release is pressed, the light is measured at every pixel separately for a certain duration to build the digital image. In a single pixel camera we instead focus the incoming light on a single sensor. Only replacing the sensor array with a single sensor won't work, of course, unless we want a single pixel image as the result when we push the button. The trick here is being able to control which parts of the image (or rather, the incoming light) are let through to the sensor. A simple way to do this would be to simply block everything except one "pixel" of the image at a time, and measure each pixel one at a time, for example by scanning through each row in turn (raster scanning). There are however more clever ways to sample the pixels.

1.1 Compressive sampling

Compressive sampling is a new sampling theory, that suggest that we can translate analog information into digital form with much fewer samples compared to the standard Nyquist sampling theory. In [2] the author presents the fundamental mathematics behind the theory.

In compressive sampling, we would like to reconstruct a vector $x \in \mathbb{R}^N$ from linear measurements y

$$y_k = \langle x, \phi_k \rangle, \quad k = 1, \ldots, K, \quad \text{or} \quad y = \Phi x \tag{1}$$

We aquire information about the unknown signal by sensing x against K vectors $\phi_k \in \mathbb{R}^N$. We are interested in the case $K \ll N$, i.e. where we have much fewer measurements than unknown signal values. For instance, a time series of a sinus wave, can be expressed as one single fourier transform coefficient, if we let the vector ϕ_1 correspond to a sinus wave.

Given the observations y and the vectors Φ, the original vector x can in many cases be exactly recovered by solving the convex program

$$\min_{\tilde{x} \in \mathbb{R}^N} \|\tilde{x}\|_{\ell_1} \quad \text{subject to} \quad \Phi \tilde{x} = y \tag{2}$$

The methodology of compressed sampling can be applied to image acquisition. This is is in fact done in Magnetic Resonance Imaging, and is also common in many fields of science, such as astrophysics. In the case of singe pixel camera, we generate pseudo random vectors ϕ_k and let them represent pixel maps. The image to be aquired is mirrored on the $k = 1 \ldots K$ pixelmaps, and the resulting total intensity y_k is recorded. We assume that number of samples K is much less than the number of pixels N in the image to be sampled, i.e. $K \ll N$.

Say we are capturing an image 32 pixels wide by 32 pixels high for a total of 1024 pixels ($N = 1024$). Using the raster scan, we need to sample 1024 times, or we will get pixels with unknown values in the resulting image. Using compressive sampling, we might instead get a low quality (soft, noisy) version of the full image already at a few hundred samples or less ($K = 100$), depending on how good our reconstruction algorithm is. As we approach $K = 1024$ measurements, the resulting image gets gradually better. Similarly to traditional image compression, where we trade off quality for a smaller file size, we can trade off quality for a lower number of samples. In contrast to traditional image compression where we sample the whole image, after which we throw away information; with compressive sampling we never even sample the whole image in the first place.

In JPEG compression, we can often obtain indistinguishable quality using a compression to 2 bits per pixel (2 bpp). This corresponds to a compression ratio of 24:2 (original bitmap 24 bpp). In some cases, we can use 0.25 bpp compression (reduction of almost 100:1). Similar compression ratios could be seen as an objective in the single pixel camera scenario.

1.1.1 Recovery algorithm

To recover the image, according to Equation 2, some optimization algorithm have to be used. Candès and Romberg [1] have developed a collection of algorithms for solving compressive sensing problems called ℓ_1-magic. In compressive image sensing one is in general interested in total variation (TV) minimization which is used in image processing to remove noise from images. TV minimization uses the $|\cdot|_1$-norm and gives raise to non-linear optimization problems [5]. These problems are generally recastable into second-order cone programs (SOCP) which are easier to solve. Li has developed an augmented Lagrange algorithm [3] for solving SOCP problems using less computational time for TV minimization problems compared to the ℓ_1-magic algorithms.

2 METHODOLOGY

The main idea of the project was to build a single pixel camera out of an old video projector. As mentioned, in order to perform compressive sampling, we need to be able to only let through certain parts of the image to the sensor at a time. This, in a way, is similar to how a video projector operates, only in reverse. One common type of video projector is the DLP (Digital Light Processing) projector. Figure 1 gives a simple illustration of how such a projector commonly works. Basically, we have a lamp (often a metal-halide type lamp, but LEDs have also become more common), which shines through a spinning color wheel. The colored light then hits a digital micromirror device (DMD) by Texas Instruments, which is an array of very small mirrors (much smaller than the width of a human hair). The mirrors can be individually pivoted into one of two positions. By pivoting the mirrors, the incoming light can be reflected either into a light sink which absorbs the light, or out through the front lens onto the projected image.

The color wheel and DMD are synchronized to produce the RGB-colors of the picture. If we “reverse“ the flow of light by replacing the projector lamp with a light sensor, we will basically have a single-pixel camera, where the incoming light can be blocked or let through by the DMD. A projector appeared to be a good candidate for building a camera, since all of the optics are already in place. The optics are however tuned to the task of projecting an image onto a wall, and not for photography, which is a disadvantage with the approach.

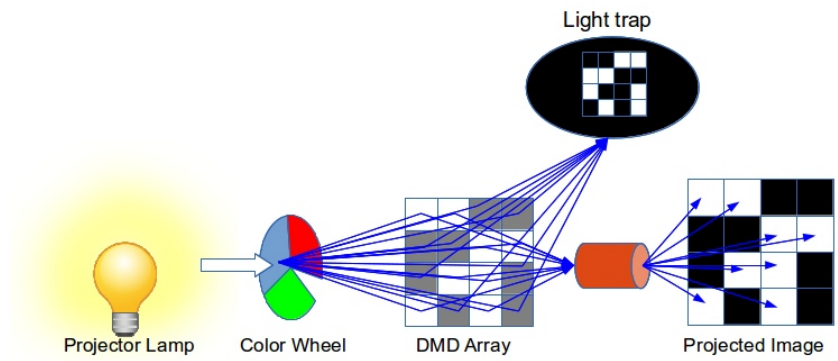

Figure 1: A simplified illustration of the operating principles of a typical DLP projector. The projector lamp illuminates the DMD array through a color filter. Depending on the orientation of the individual mirrors, the colored light beam is either directed away into a "trap", or let through to the projected image.

3 IMPLEMENTATION

We decided to build the camera based on an InFocus LP 350 projector [4] from the beginning of the 21th century (roughly 2002 based on date codes on the projector). The projector has a native resolution of 1024x768 pixels. The plan was to feed the projector with black and white patterns corresponding to which parts of the image should be blocked and let through to the sensor. The patterns would be fed through the VGA input at a refresh rate of 60 frames per second, and thus 60 patterns per second (more on this later). In order to function as a camera, some modifications were performed, and some additional components were added to the system. Figure 2 shows a diagram of the final system, the components of which will be discussed in the following section.

3.1 Modifying the projector

As mentioned, we want to replace the projector lamp with a light sensor to "invert" the projector. The first obstacle to doing this is that the projector does not run without a lamp connected. After consulting various resources on the internet, and a trial-and-error approach, it was found that the projector logic can be circumvented by shorting out a permanent 3.3V pin and a pin that is used to determine whether the lamp is lit or not. These pins were found on a connector between the power supply board and main board of the projector. The shorted pins on the connector is shown in figure 3.

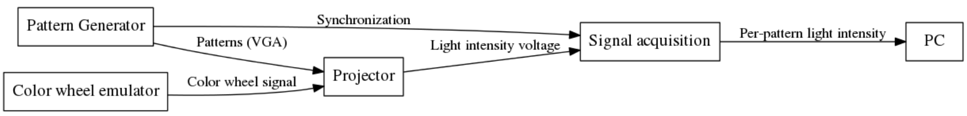

Figure 2: This shows how the various components described are connected together.

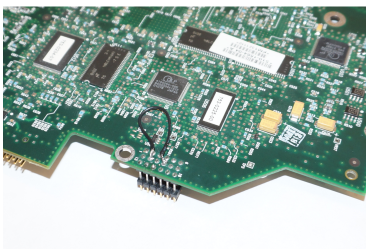

Figure 3: Pins 7 and 10 (3.3V) on the 18-pin connector between the main board and power supply board of the projector were shorted in order to make the projector believe that the lamp is lit, even though it is not present.

We also decided to remove the color wheel, since our prototype was only going to capture grayscale images. With the color wheel removed, we can use the full duration of a video frame to capture one single-pixel sample. Here we encountered the next problem. The projector will not run without a color wheel! A picture of the lens

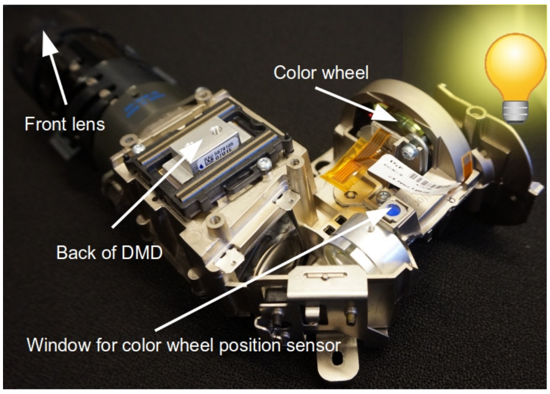

Figure 4: The lens assembly of the projector, with the locations of the lamp, color wheel, DMD, and final output lenses marked. Also shown is the small tinted window which is used by a photodiode on the main board to measure the position of the color wheel.

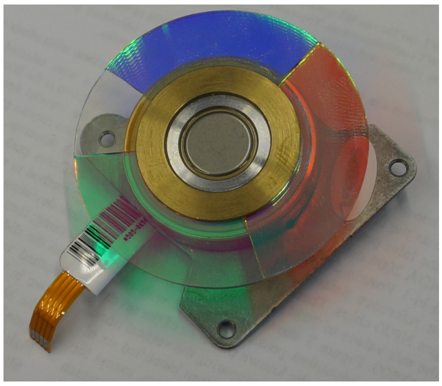

Figure 5: The color wheel of the LP350 projector.

assembly is shown in figure 4, where the placement of the lamp, color wheel, DMD, and final lenses are shown. After some examination, we discovered that there is a small window mounted on top of a light tunnel between the color wheel and DMD housing. Some light will escape through this tinted window and onto a photodiode on the main board of the projector. The color wheel can be seen in figure 5. It has three color sections for blue, red, and green, as well as a small clear section. Using an oscilloscope, we measured that the photodiode produced high pulses when the clear and red areas were lit, while the blue and green areas generated low output.

Our solution to this was to remove the photodiode from the main board, and instead feed this sequence of pulses to the main board using a microcontroller GPIO pin. A voltage divider circuit was placed on the output pin of the microcontroller to lower the output voltage of the pin to less than 700 mV in order to hopefully avoid damaging the projector.

Once the modifications mentioned above had been made, the projector would turn on and stay on despite the missing components. In order to turn the projector into a camera, we needed a light sensor to replace the lamp. Several alternatives were tested, and were more or less suitable. We first tried using a simple photoresistor, which seemed to react too slowly to changes in light intensity. Photodiodes are much faster. We found that light was focused on an area of roughly $5\,\text{mm} \times 5\,\text{mm}$ behind the lens closest to the color wheel position sensor window (see figure 4). This area was larger than most of the photodiodes we had on hand. We decided to use a *FDS10X10* [6] $10\,\text{mm} \times 10\,\text{mm}$ large surface area photodiode from Thorlabs. The photodiode was mounted by cutting a groove in the metal chassis of the lens assembly to hold the diode.

The photodiode outputs a current proportional to the amount of light. An amplification circuit using an op-amp was created and placed between the photodiode and the ADC input of a Texas Instruments Tiva C development board used for data aquisition.

3.2 Pattern Generation

The VGA input of the projector was used to supply the patterns to the DMD array. A PC would be the obvious choice for supplying the pseudo-random patterns to the projector. An issue here is, however, that we want to synchronize the signal acquisition board with the pattern generator since the reconstruction algorithm needs to connect measured light to which specific pattern was displayed. Due to many layers of abstraction and indeterministic behavior, as well as a lack of general purpose output ports, supplying the information on exactly when a frame is output to the acquisition board is not straightforward.

We did however, have access to software[1] for the STM32F4 Discovery development board to generate a VGA signal in software. The software uses DMA (direct memory access) transfers to supply an 8-bit VGA signal through a resistor ladder from the IO pins on the board. This software was used to generate the pseudo-random patterns, and as the signal was generated in software on a microcontroller, we are able to send a precise synchronization signal to the signal acquisition board when a new pattern begins.

[1]The software was originally part of a "demo" at Revision 2012, available at `http://www.pouet.net/prod.php?which=59095`

3.3 Image reconstruction

A typical workstation PC was used to receive the pattern-wise light intensity readings and construct an image from a set of pattern-intensity pairs. The reconstruction was performed using the TVAL3 (TV minimization by Augmented Lagrangian and ALternating direction ALgorithms) algorithm [3]. The reconstruction algorithm uses a set of measured light intensities for different patterns to reconstruct an output image. In the prototype the pattern resolution was $64 \times 64 = 4096$ pixels. The program controlling the image reconstruction was written in python, calling the Matlab code provided by Li[2]. The python program continuously received light intensity values from the acquisition board, and forwarded the desired amount of samples (depending on which "compression ratio" was desired) to the reconstruction algorithm. The resulting image was then output on a monitor.

4 RESULTS

The main goal of our single-pixel camera project was to build a proof-of-concept prototype to be displayed at the "Mission: Better Life" exhibition in Helsinki, in late spring 2014. This goal was accomplished, however in this chapter we will share some early results in terms of picture quality and performance of the system described in section 3.

4.1 Picture quality

In figure 6, an example image captured using the single pixel camera is shown. It is a portrait of a sitting person. Figure 7 shows a comparison between a digital camera photo and a single-pixel camera photo of an "A" taped onto a wooden surface using duct tape. Figure 8 shows six reconstructed images depicting the same "A" character. The images are reconstructed using different numbers of light intensity measurements. The number in the corner of each image is the number of samples used to reconstruct the $64 \times 64 = 4096$ pixel image. At the top left, we use 4096 samples to reconstruct the image, while the image at the bottom right uses only 128 samples, which corresponds to a 32:1 compression ratio.

4.2 Performance

The software of this prototype was not optimized for speed and there are several opportunities for improving the system performance. Since the pattern generation was set to 60 Hz, we could capture a maximum of 60 samples per second, i.e. roughly 2 to 70 seconds for the pictures seen in figure 8. The reconstruction algorithm speed is dependent on parameters used. Execution time for the Matlab algorithm, run by the GNU Octave software on an Intel i5 CPU-based laptop was roughly 0.5 to 2 seconds depending on the number of samples used.

Figure 6: An example portrait of a sitting person using the single-pixel camera, with a resolution of 64×64 pixels.

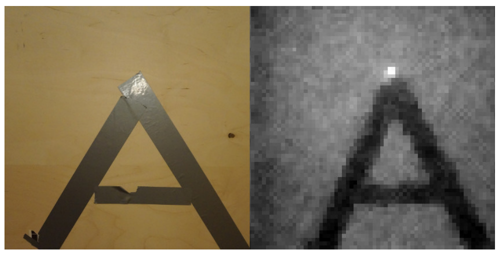

Figure 7: A "regular" digital camera and single-pixel camera photo side by side depicting an A figure.

5 CONCLUSION

This paper has presented how an old DLP video projector was transformed into a single pixel camera. Normally a lamp is generating the light that is used for projecting a picture, via a micro mirror device, onto a wall. In the modified version, the light from the outside is projected through the micro mirror device to a single light sensor. The approach exemplifies two things; the process of compressing sampling and the possibilities for alternative camera designs.

[2]The code for TVAL3 can be obtained from the following url: http://www.caam.rice.edu/ optimization/L1/TVAL3/

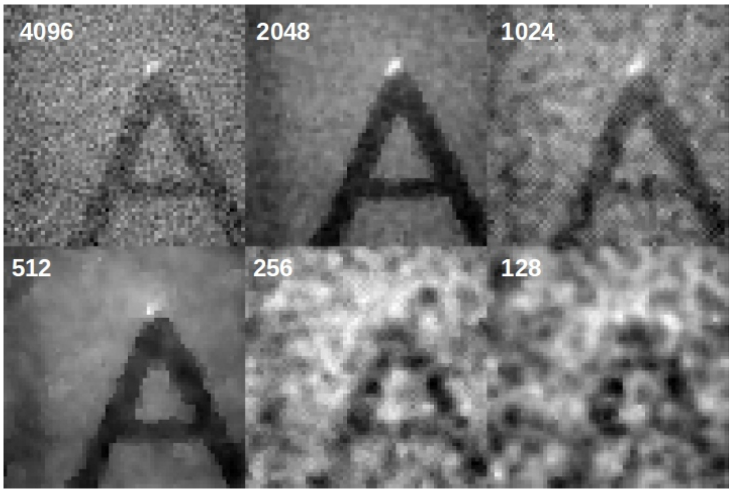

Figure 8: An "A" figure captured and recovered with varying amounts of samples. The number of samples used to reconstruct the image is shown in the upper left corner of each image.

The paper presents the hardware modifications done, the supporting hardware and the software system for reconstructing the image. The design was originally constructed for the Millennium Pavilion in Helsinki 28 April to 14 May 2014, and was used for live demonstrations of alternative approaches in signal processing.

This prototype was created with standard components, and rather inexpensive supporting electronics. The prototype did however not reach high performance, as sampling and reconstruction of a still picture took up to 60 seconds. Clearly, this should be improved. The technical bottle neck at the moment is the Digital Micro mirror Device that in the configuration only supporting a frequency of 60 Hz. This is not a limitation of the DMD itself, but of the configuration used. A more integrated DMD would enable sample rates of up to tens of kHz. The reconstruction algorithm could also likely perform significantly faster, if it were re-implemented with speed as a priority.

As a normal visual light camera, this single pixel approach is inferior to a normal camera. However, the approach makes it possible to make a camera that is sensitive to almost any wavelength of light that is reflected by the DMD, ranging from UV to IR light. When combined with optics, e.g. a prism separating wavelengths to different sensors, we can build a spectrum camera that captures several various wave lengths at the same time. This could be used e.g. in quality control in manufacturing processes.

References

[1] Emmanuel Candes and Justin Romberg, *l1-magic: Recovery of sparse signals via convex programming*, URL: `www.acm.caltech.edu/l1magic/downloads/l1magic.pdf`.

[2] Emmanuel J. Candes, *Compressive sampling*, Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures, pp. 1433–1452.

[3] Li Chengbo, *An Efficient Algorithm For Total Variation Regularization with Applications to the Single Pixel Camera and Compressive Sensing*, Master's thesis, Rice University, 2009.

[4] InFocus, *InFocus LP340 LP350 User Guide*, [Online; accessed 31-July-2014] `http://www.infocus.com/resources/documents/LP300/InFocus-LP340-LP350-UserGuide.pdf`, 2000.

[5] Leonid I. Rudin, Stanley Osher, and Emad Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D **60** (1992), no. 1, 259–268.

[6] Thorlabs, *Silicon photodiode fds10x10*, Spec Sheet, `http://www.thorlabs.de/thorcat/24700/FDS10X10-SpecSheet.pdf` [Online; accessed 5.8.2014], March 2014.

[7] Michael B. Wakin, Jason N. Laska, Marco F. Duarte, Dror Baron, Shriram Sarvotham, Dharmpal Takhar, Kevin F. Kelly, and Richard G. Baraniuk, *An architecture for compressive imaging*, Image Processing, 2006 IEEE International Conference on, IEEE, pp. 1273–1276.