Visible Light Communication-based Indoor Positioning with Mobile Devices

Author:

Zsolczai Viktor

Introduction

With the spreading of high power LED lighting fixtures, there is a growing interest in communication schemes that use Visible Light Communication (VLC). Main advantages of this method include the lack of interference with electromagnetic signals, regulation-free spectral domain, eavesdropping-safe channel, no medical risks and, most importantly, multi-functionality.

This paper details a possible solution for indoor positioning systems, where the information is transmitted by modulated light, and the signals are demodulated and decoded by a smartphone or tablet that is equipped with a CMOS camera and sufficiently powerful hardware to perform simple image processing algorithms. My work mainly focuses on the modulation method and the data decoding.

During the last semester, my goal was to set up a demonstrational system that can potentially be used for indoor positioning. In the transmitter, an LED luminaire is controlled by a driver circuit that produces three-state intensity modulation. The corresponding, seemingly non-flickering light is captured with a "smart" device, and by exploiting the "rolling shutter" image capture method used by CMOS cameras, we are able to restore the original information through examining the spectrum of the picture.

Demodulation with smartphone: rolling shutter

One great drawback of VLC is the lack of widespread, already accessible receivers. While the most electronic device is capable of communicating in the ISM band (via Wi-Fi or Bluetooth), in order to demodulate high-speed optical streams a device developed for this specific reason should be manufactured. To eliminate this problem, we have the option of using mobiles and tablets as receivers.

Nowadays the most handheld device is equipped with CMOS cameras, mainly because of their lower price. Unfortunately, contrary to CCD sensors, capturing the pixels of an image does not happen simultaneously, but it is rather done by sweeping through the rows of the picture; this method is called rolling shutter. As a consequence of this procedure, the picture can be distorted in case of sudden changes of the ambient light. The difference between rolling shutter and traditional image capturing is shown on *Fig.1*.

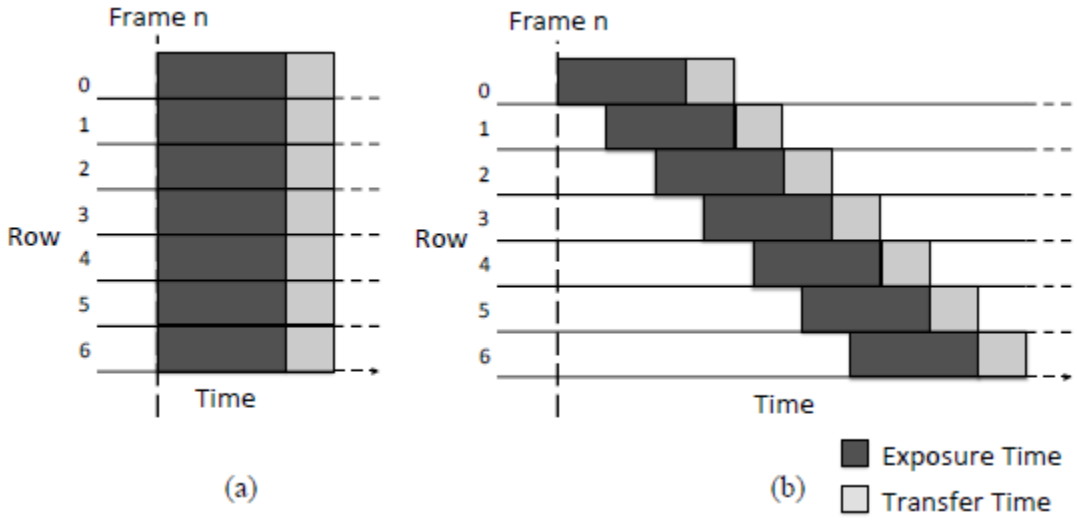

Figure 1: Global shutter left), and rolling shutter (right)

This kind of problem arises mostly in environments where the illumination is periodically flickers; in these cases vertical or horizontal stripes appear in the image, which (considering the shutter's theory of operation) is a completely obvious phenomena. The frequency of the observable darker and lighter stripes depends on the size of the image, the frame rate, the exposure time, and most importantly, the flickering light's frequency. Since the factors regarding the recording device are usually adjustable, or known to be static, the evolving pattern is a function of the illumination only. That is, if we intentionally "disturb" the light of the indoor lighting fixtures, the frequency of the "disturbing" signal can be recovered by image processing procedures. To do this, we need to evaluate the two-dimensional Fourier-transform of the pixels, so we can perform frequency-domain analysis. In our case, when the phenomena to be examined varies only horizontally or vertically, the problem simplifies to one-dimensional FFTs, moreover, we can average the values through rows/columns, and then perform only one FFT.

The absolute values of the resulting complex numbers give us the amplitude spectrum, where the distinct peaks are related to different modulating frequencies. The mapping between the spectrum of the modulated light and the FFT-data is linear, so if the light source is controlled with two different signals with a relation of their frequencies being k/l, the indices of the belonging data in the spectrum will have the same ratio.

Properties of the transmitter

The driver circuit

The light source in the transmitter is a GE LED fixture. In order to drive the luminaire with proper voltage and current, an electrical ballast is used to transform the line power in an appropriate manner. The driver circuit that carries out the modulation of the light can be found between the ballast and the LED module.

The driver circuit applies the XOR and the AND operation on its two inputs; let the resulting outputs be $s_{xor}(t)$ and $s_{and}(t)$. The s_{and} signal is connected to the gate of a MOSFET that is implementing a low-side switch for the LEDs: consequently, the ON state gives full illumination, while in the OFF state the LEDs are turned off. On the other hand, the s_{xor} signal goes to the base of a Darlington-pair through a potentiometer; with the help of the variable resistance, we can adjust the gain of the Darlington amplifier, which is also connected in series with the LED rows. Basically this part of the circuit has the same function as the one mentioned earlier, but while the MOSFET turns the diodes fully on or fully off; in this case we can produce light with intensity half as great as the maximum available. Naturally, the transistors used in this project need to stay operational up to ~60 V of drain-source voltage, and the Darlington has to be able to dissipate a fairly great amount of power. Apart from the usual additional elements, an active Zener voltage regulator is also present, in order to produce a stable 5V supply for the logic gates. *Fig.2.* shows the schematics of the circuit.

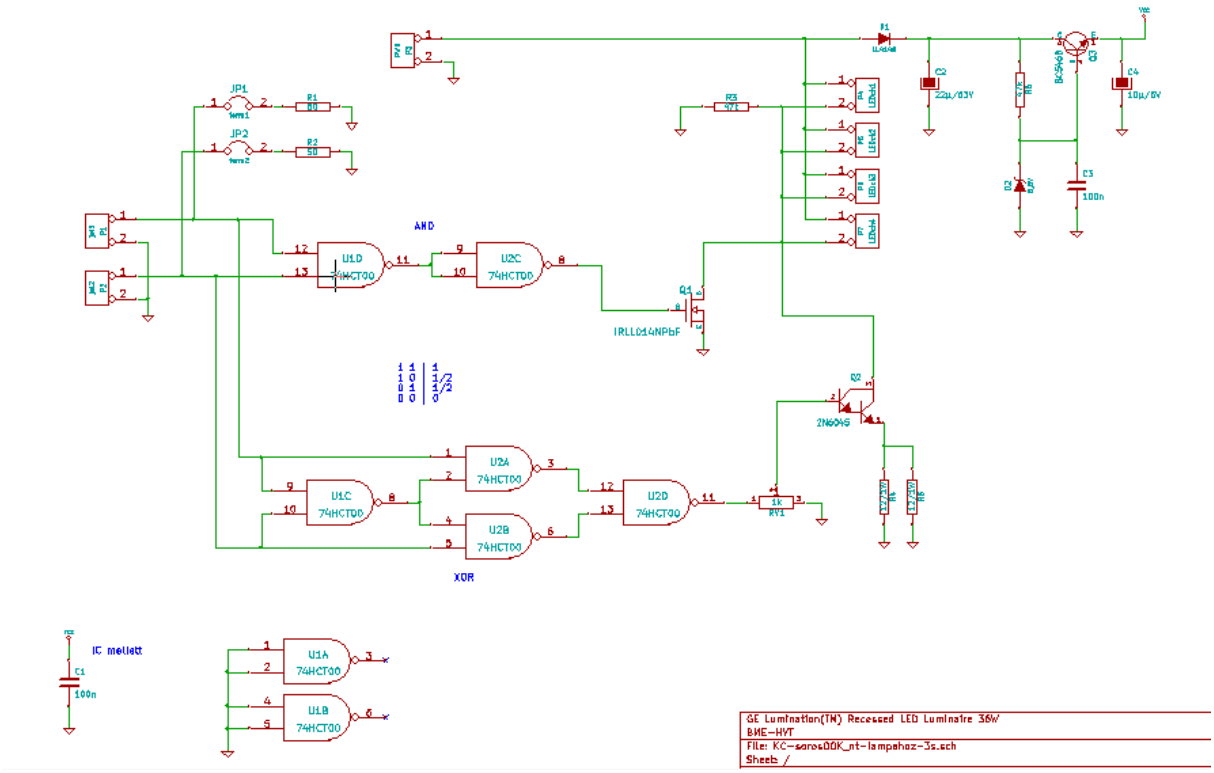

Figure 2: Schematics

Creating control signals, set-up of the arrangement

As it is mentioned in the previous section, the drives needs to input signals, and as a function of those, it alters the power of the lamp. The goal of the modulation is to create two unambiguously distinguishable frequency components in the signal, so that they can be decoded and separated. To achieve that, the implemented system is controlled with two square waves.

To produce the signals, I used Arduino Uno microcontroller boards; the code in the device uses the internal timers and the interrupts connected to them in order to generate the appropriate signals. *Fig.3.* shows the complete transmitter, with the lighting body, the electronic ballast, the driver circuit and the microcontroller. on the picture of the right, a diffuser is placed on top of the LED fixture in order to create more evenly distributed illumination.

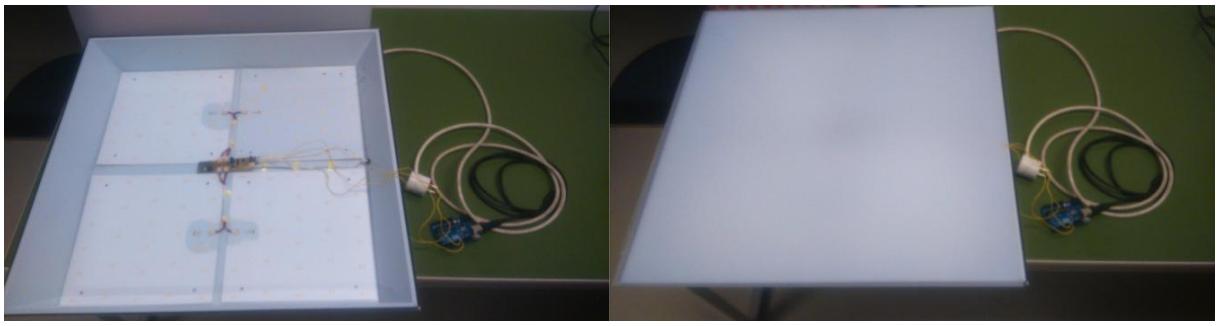

Figure 3: The arrangement on the transmitter's side

The effect of the "half-power" light's actual intensity on the decoding.

Let's consider the case when the modulating signals are two square waves with the same phase but different frequencies. Let the full intensity be 1 as reference; the output of the XOR gate generates a smaller intensity light, denoted with *a*. The connection between the logic values of the inputs and the intensity (denoted with *I*) of the corresponding light can be characterized as follows: if both signals are logic high, I=1, if both are logic low, I=0, and $I=a$ otherwise.

Let the two signals be $s(f_1t)$ and $s(f_2t)$. In the optimum case, f_1 and f_2 appears in the spectrum, and no other components do. Since ideal square waves have only discrete spectral components at odd harmonics, the sum of the two signals will more or less meet the requirements with correct filtering. If we think of the modulated signal as the sum of a square wave changing between *0* and *a* with a frequency of f_1, and another square wave with changing between *0* and *1-a* with a frequency of f_2, the result is almost equivalent with the real output. If the both logic levels are low, the intensity is *0*, if both are high, the output's intensity is $1-a+a = 1$, and in the case when s_1 is high and s_2 is low, the intensity is *a*.

However, if s_1 is low and s_2 is high, the sum of their amplitude is *1-a* while the intensity of the light is *a*. To eliminate the difference, in the function that describes the intensity of the illumination we must add a term that is constant zero when s_1 is logic high, and varies between $a-(1-a) = 2a-1$ and zero with a frequency of f_2 when s_1 is logic high. It is not hard to see that the signal which is constructed in the above mentioned way is nothing else but the product of s_1 and s_2 (with the specified amplitude of course). We can write this in the following form (with omitting amplitudes for the sake of simplicity):

$$s(\omega_1 t) \cdot s(\omega_2 t)$$
$$= [DC_1 + \sin(\omega_1 t) + \sin(3\omega_1 t) + \cdots] * [\mathrm{DC}_2 + \sin(\omega_2 t) + \sin(3\omega_2 t) + \cdots]$$

According to this equation, products of the different harmonics will appear in the light signal, causing components at the sum and difference of the frequencies in the spectrum. The phenomena can be avoided by setting *a* and *1-a* to be equal, thus eliminating the term that originates from the product of the inputs. That is, we should adjust the potentiometer so that the "half-power" light's intensity is exactly the half of the maximum intensity.

Optimal setting of the intensity with the help of analysis in Matlab

According to the conclusions of the previous section, optimum decoding requires the intensity to be half of its full value when the LEDs are driven by the output of the XOR gate. In order to make the setting easier, first I captured a reference video on which the driver circuit's inputs are the same, 2 kHz square waves. This video was processed by a Matlab code detailed below.

After reading the movie file, the programme gets the first frame of the recording and converts it to a grayscale image, since the information which is useful to us is carried in the intensity of the light, not in the colour of it. The next step is averaging by rows, which results in the time-domain signal vector. Finally, the code performs filtering and FFT, and plots the results. *Fig.4.* shows the amplitude spectrum of the reference signal.

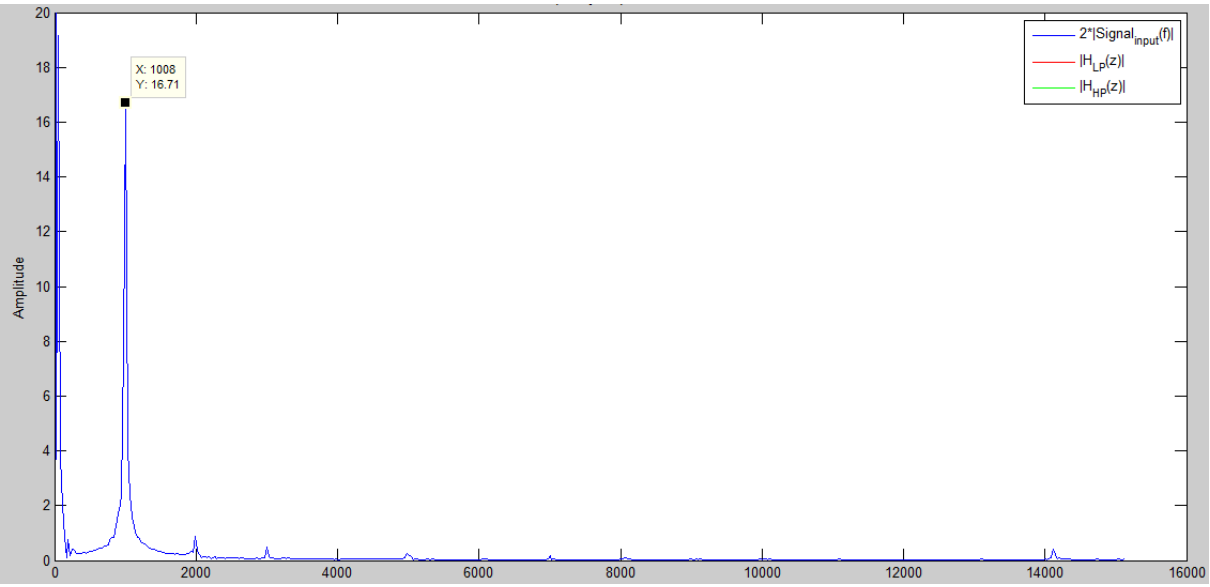

Figure 4: Amplitude spectrum of the reference signal

The graph shows that the amplitude of the spectral line at 2 kHz is ~16.7 units. After this, I modified the configuration so that one of the inputs still remains a 2 kHz square wave, but the other is connected to ground; this way I could examine the intensity set by the potentiometer. I repeated the aforementioned steps, and measured the amplitude of the 2 kHz line in the resulting spectrum. During the steps I tried to adjust the resistance so that the corresponding intensity causes a spectral line at 2 kHz with ~8.3 units of amplitude. However, this process proved to be fairly circumstantial, so I settled at an approximately good value, ~7.5 units. *Fig.5.* shows the final state of the spectrum.

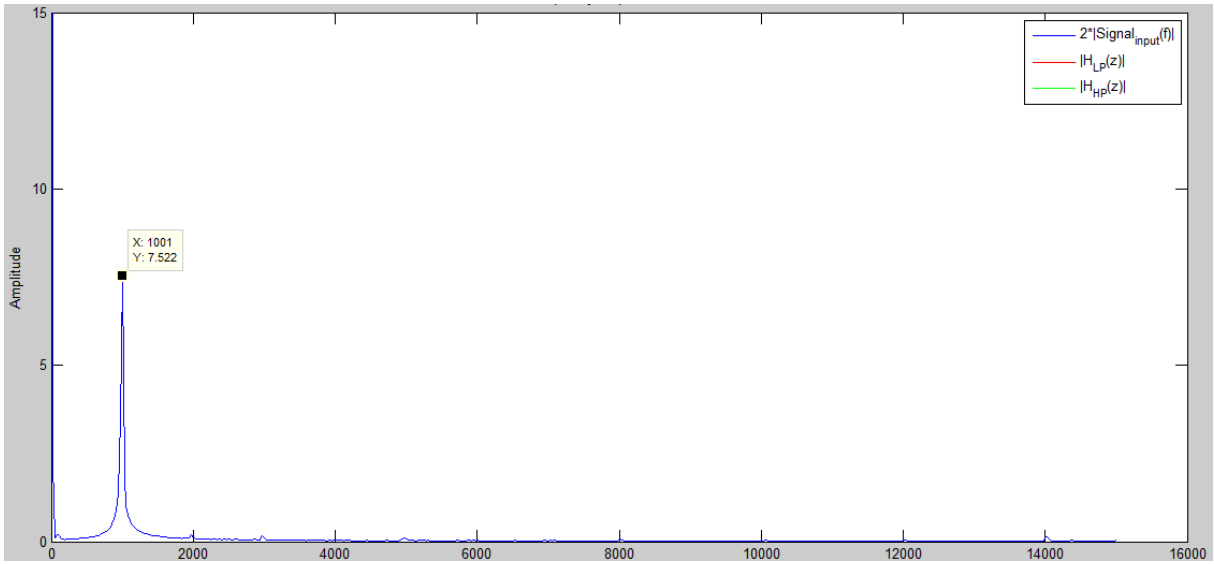

Figure 5: Amplitude spectrum in case of "half-power" light, after configuration

Finally, for comparison's sake, *Fig.6.* and *Fig.7* shows the spectrum with two different configurations, both modulated with a 5 kHz and a 2 kHz square wave.

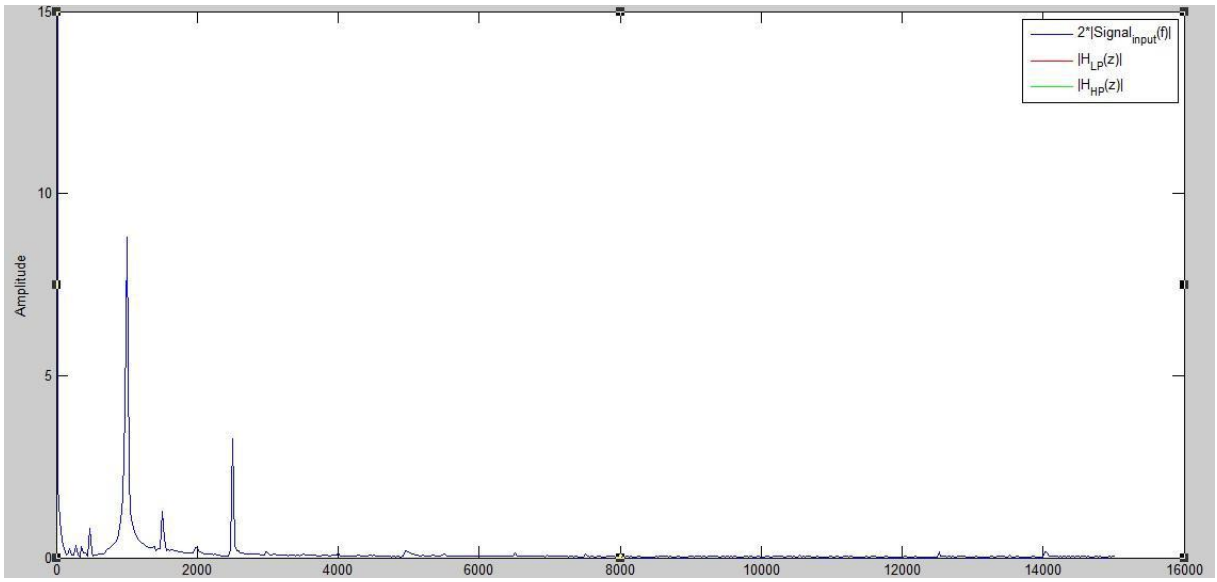

Figure 6:Amplitude spectrum with final configuration; 2 kHz and 5 kHz controlling signals

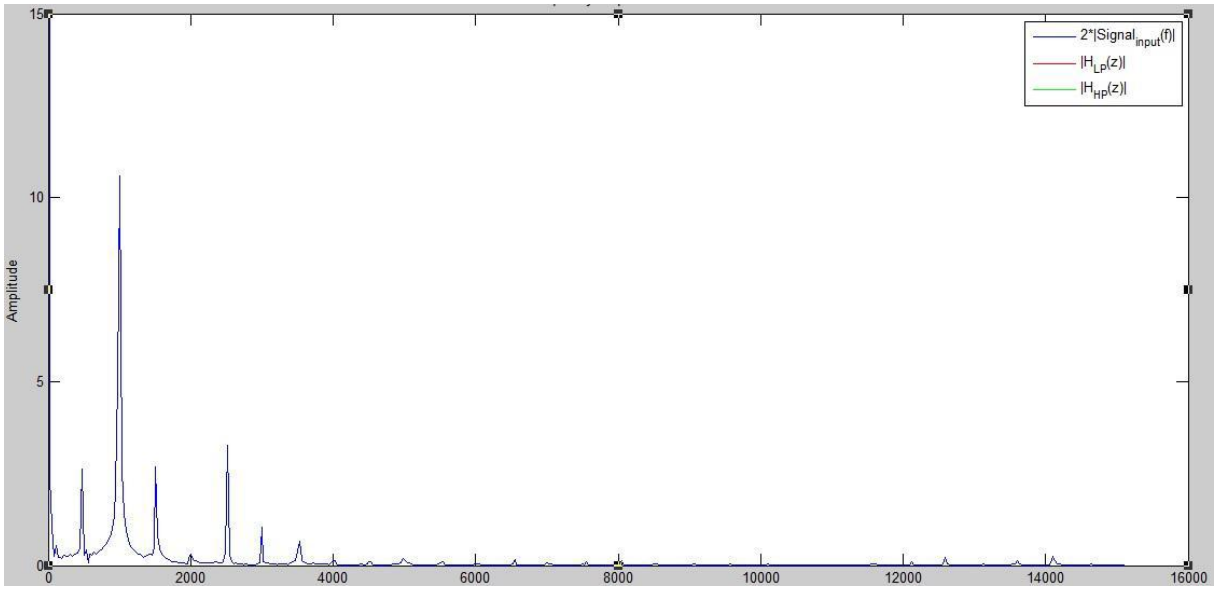

Figure 7: Same modulation as before, but different calibration

As it can be seen on the second figure, the energy of the undesirable components are much higher when using the wrong settings, which can result in errors during decoding.

Implementing the algorithm in the receiver

So far I summed up the steps and final results of the transmitter's setup. My aim was to transmit three different symbols, each having a 2 kHz component, and a second component of

3 kHz, 4 kHz and 5 kHz respectively. The second part of my research was to implement the receiver.

To do so, I wrote a programme for an iPhone 5S device that can perform decoding of the sent data, albeit not in the most sophisticated and robust way. The graphical user interface of the application contains a preview of the device's camera, and buttons that start the recording/analysis.

Instead of taking a picture, the algorithm running on the mobile starts recording a video, stores the first frame, then stops the recording. This seemingly unnecessary method has two advantages: firstly, this way the frame rate of the camera is adjustable, secondly, in case of continuous data streaming we should use this solution anyway.

After creating the image, the algorithm cuts out a 1024x1024 sized rectangle from the middle of the picture and converts it to grayscale. The crop is needed because FFT works the most efficiently on data sets that have the size of a power of two. In the next step I used the appropriate functions of the *vDSP Framework* to average the pixels by columns and perform FFT. Between these two operations I also needed to filter the near-DC components of the data, as these usually dominates in pictures, but in my case made decoding impossible.

The code then generates the absolute values of the obtained complex numbers, and finds the index of the global maximum in the data row. This is followed by a second peak search, but the first maximum and its small neighbourhood must be omitted. Finally, the algorithm calculates the ratio of these two indices; if the result equals to 2/3, 2/4, 2/5 or the reciprocals of these numbers, "3 kHz", "4 kHz" or "5 kHz" appears on the screen respectively. Otherwise we can see "Not a valid code".

Results with the demonstrational system

According to my experiences, the decoding is successful in most cases when the transmitter and the receiver are sufficiently close to each other, however, the results show strong degradation if the distance is increased or the incident angle is too sharp.

The most problems occurred when I tried to use 2 kHz and 5 kHz signals together, which is not surprising, for more reasons. Firstly, the relatively great frequency of the 5 kHz signal means that the corresponding stripes in the image are narrower, they contain fewer pixels, and

consequently the attenuation is high because of the sampling method. Secondly, due to the imperfections in the setup of the light's intensity, a spectral line appears at 3 kHz, which may even exceed the line at 5 kHz; in this case, the programme faults.

Possible future developments

There are many problems that need to be overcome in order to make the system appropriate for indoor positioning. Bit error rate must be brought down; the number of used symbols and the effective range of the device must be increased; and the receiver has to be able to decode signals even if more transmitters are present. Furthermore, aspects like energy-efficiency (the driver circuit dissipates a lot of power), integrability (obviously it is not reasonable to use an Arduino for every single fixture), and easy manageability (for example, the system needs to be easily recalibrated when a change in the environment occurs).

Some optimization methods are already obvious. For instance, the driver could be integrated in the electronic ballast, which not only meant better energy-efficiency and more aesthetic looks, but the usage of the spectrum could also be done in a more appropriate manner. Another possible way is to improve the algorithm in the mobile phone in order to process every single frame recorded by the camera, this way a slide window could be applied to the data, and the effect of the errors could be smoothed; moreover, continuous positioning would be possible. Yet another way to improve the system is to make the exposure time adjustable, as the optimum value of this attribute broadens the detectable frequency range.

In a realistic system that can be used for indoor positioning, there are more than one transmitters, and in some cases the number of codes associated with them can grow really large. To avoid interference, the signals have to be multiplexed in some way. The most straightforward solutions are frequency and time domain multiplexing, since these are easy to implement and still able to raise efficiency.