

# Active Eye-Glasses with Local Dimming

Görkem Özvural, Şevket Gümüştekin

Department of Electrical and Electronics Engineering,  
İzmir Institute of Technology, 35430, Gülbahçe, İzmir, Turkey  
gorkemozvural@gmail.com, sevketcumustekin@iyte.edu.tr

## Abstract

The sense of sight provides the most reliable information about the environment. Although the human eye has a wide dynamic range for brightness adaptation, direct exposure to relatively bright light sources wipes out the total visual perception. The most remarkable example of this situation is driving a car at night, while the other cars are coming from the opposite lane with headlights on. In this work, we have proposed and developed a set of LCD equipped active glasses to solve this light adaptation problem. First, the location of the light sources are determined using an on board camera which is assembled in front of the glasses. After processing image data, the locations of the bright light sources are darkened and the disturbing effect of intense light sources is reduced.

## 1. Introduction

Human vision easily adapts to the environment by controlling the amount of light entering through the pupil. In extreme conditions, people tend to use sunglasses to further reduce the amount of light at the expense of losing details in the dark regions. One of the examples of this situation is driving a car towards a bright light source such as the sunlight or a headlight. LCD (Liquid Crystal Display) technology makes it possible to locally darken a transparent material which can be used as active glasses. By locally dimming high intensity light beams through the eye, without decreasing the light coming from the dark regions, the loss of details in the dark regions is prevented.

For locating light sources in the scene, it is essential to use a camera. Other solutions such as light dependent sensors won't provide sufficient locating resolution of light sources. There are two major design challenges in building active glasses. One of them is designing the hardware that provides interfacing between the camera and the LCD active glasses. The other is the development of the compatible software which provides data interfacing and implements image processing algorithms.

Camera control interfacing provided by the manufacturer makes it possible to achieve desired and appropriate operations of the camera. Usually, cameras have read/write available control registers which are accessible via serial interfaces. Specific binary values release output data and synchronization signals.

Camera data interfaces are used to transfer pixel data. Vertical and horizontal reference signals are especially important to synchronize the receiving party to get the right pixel data. The data transferred by the internal processor of the camera is arranged according to the pre-set control register values. Several options may be available to change output data arrangement. Different output formats such as RGB and YUV are most common formats that are supported by most of the

commercially available camera chips. In this project, the only information we need is luminance data, so YUV output format is used.

Thresholding is applied to the raw data to obtain binary image. Binary data of each 8 pixel are packed into bytes. This provides compression and makes it possible to save the image to the limited memory of the microcontroller. Then perspective transformation is applied to the packed image. The main contribution of this work is the efficient use of low cost hardware with a limited amount of memory.

The LCD units which were used as active glasses are dot matrix displays with chip-on-glass driver. LCD control and data transfer are achieved via serial interface. Data and control bits are transferred to the LCD driver via the same interface with a different prefix.

## 2. Hardware Design

A camera module, LCD driver and microcontroller having the common operation voltage of 3.3V DC are the essential parts of the hardware. At the power supply block, 2 AAA batteries are used to provide power to the whole circuit. A step-up DC-DC converter is used to convert AAA battery voltage of 1.0-1.5V DC to the operation voltage of 3.3V DC.

The camera module chosen for this project needs external clock, so a simple inverter based oscillator circuitry is designed to provide 16 MHz clock signal. Microcontroller oscillation signal is provided by an oscillator built in the breakout board.

The serial camera control bus of our camera is I2C capable and the microcontroller has dedicated I2C interface. So the camera is controlled easily using standard I2C functions. On the other hand, the LCD module uses 3-wire serial bus. Hence the control of LCD driver is achieved by a customized program.

The block diagram of active glasses is shown in Fig.1. The circuit boards containing these parts are tightly assembled on laboratory protection spectacles to form active glasses.

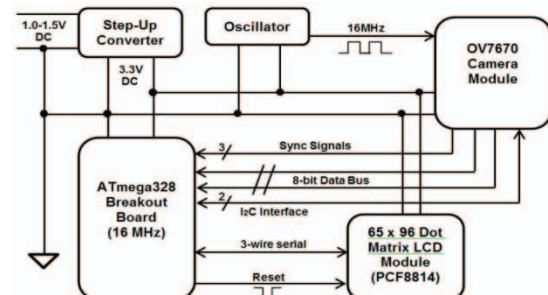


Fig. 1. Block diagram of active glasses hardware

## 2.1. Step-Up Converter

NCP1400 PWM step up switching regulator breakout board is used to convert AAA battery voltage of 1.0 - 1.5V DC to 3.3V DC. The breakout board accepts voltage inputs between 1.0 and 2.0V and gives a constant, low ripple 3.3V output capable of sourcing up to 100 mA.

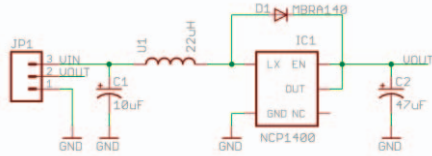


Fig. 2. NCP1400 DC Step-Up Converter breakout circuitry

## 2.2. Camera Oscillator

In the oscillator circuit, high speed Si-gate CMOS inverter 74HC04 is used [1]. An inverter, a 16 MHz quartz crystal, 2 resistors and 2 capacitors form a simple Pierce oscillator circuit. The selection of resistor and capacitor values depends on several factors such as crystal characteristics, frequency stability, inverter gain and start up time [2].

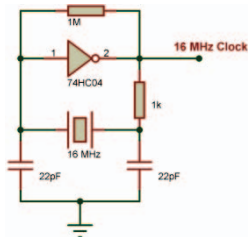


Fig. 3. Pierce oscillator using CMOS inverter

## 2.3. Atmega328 Breakout Board

The control unit used in this project is low power AVR 8-bit microcontroller that has advanced RISC architecture [3]. It has powerful instructions and most of them take single clock cycle execution. This provides up to 16 MIPS throughput at 16 MHz operation frequency in this application. Atmega328 has 32K bytes flash memory and 2K bytes RAM. On chip in system programming (ISP) flash allows uploading the program through an SPI serial interface. This feature is very useful for tightly assembled active glasses application, because we don't need to separate the controller for programming each time an update is needed.

## 2.4. OV7670 Camera Module

OV7670 camera module is a breakout board that contains an OV7670 camera chip, driving circuitry and an on-board camera lens. OV7670 camera chip is a low voltage CMOS image sensor manufactured by OmniVision. It needs an external clock generator with the minimum frequency of 15 MHz. The chip has its own analog-digital signal processors and provides 8-bit images up to 30 fps. The output formats and output data transfer are controlled via I2C capable Serial Camera Control Bus (SCCB) [4]. The camera chip has device control registers which are accessible through SCCB interface.

## 2.5. Dot – Matrix LCD Module

The LCD module used in this project is a 65x96 active binary dot matrix display with chip-on-glass driver. This LCD module is manufactured for mobile phones with different drivers for each model. Only by removing the reflector coating of the rear polarizer, we get transparent active dot matrix display. This feature is critical for the purpose of this project, because we need local darkening of the visual field. The following figure shows the layers of the liquid crystal display.

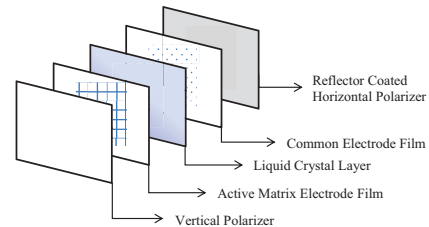


Fig. 4. Active dot-matrix LCD layers

LCD driver uses 3-wire serial interface. Although the driver chip is I2C capable, I2C wires are not available on this module. The serial interface between the LCD driver and the microcontroller is achieved by software via I/O pins.

## 3. Software Design

In this study, open-source electronics prototyping platform Arduino is used in both hardware and software design. The Arduino Duemilanove version development board is based on ATmega328 microcontroller manufactured by ATMEL®. The microcontroller can be programmed using open-source Arduino software editor and programmer [5]. Arduino development platform is based on C programming language.

### 3.1. LCD Module Interfacing and Initialization

The selected LCD driver has several timing requirements for its 3-wire interface. A function is developed to carry out the write mode transmission of one byte data or command. Some of the operating modes and application specific device functions must be executed after the system is powered up.

### 3.2. Camera Module Interfacing

In this project, OV7670 camera module is used as an input device which sends real time images to the microcontroller via an 8-bit parallel bus. The camera is assembled in front of the active glasses, so it provides the image that covers the field of view of the user. The captured images provide the locations of intense light sources which are used to locally darken the corresponding areas on the active glasses.

The control interface of the camera uses I2C capable SCCB. ATmega328 chip has dedicated I2C interface and the required functions to send and receive bytes are provided by the Arduino platform.

The output pixel data frequency is determined by the external oscillator and the internal clock divider. In this application, 16MHz external clock and the clock divider value ( $D_{CLK}$ ) of 3 are used. The camera output signal frequencies and periods can be calculated as follows.

$$\text{Internal CLK} = XCLK/D_{CLK} \quad (1)$$

$$\text{INTCLK} = \frac{16 \text{ MHz}}{3} \cong 5.333 \text{ MHz} \quad (2)$$

$$f_{\text{Pixel CLK}} = \text{INTCLK} = 5.333 \text{ Mhz} \quad (3)$$

$$T_{PCLK} \cong 188 \text{ ns} \quad (4)$$

Since there are two data bytes for each pixel in YUV (4:2:2) format,

Y-byte and U/V-byte:

$$T_P = 2 * T_{PCLK} \quad (5)$$

In this application, the camera operates in 640x480 VGA resolution. In the following calculations, line and frame periods are found for this specific resolution and frequency of INTCLK.

$$T_{LINE} = (640 * T_P) + (144 * T_P) \cong 294 \mu s \quad (6)$$

$$T_{FRAME} = (480 + 3 + 17 + 10) * T_{LINE} \cong 150 \text{ ms} \quad (7)$$

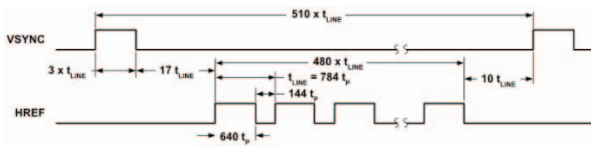


Fig. 5. Frame VSYNC and HREF timing

The VGA output of the camera needs to be down-sampled to be handled by the limited memory of the microcontroller. Under these circumstances, the camera signal conditioning has to face not only the down-sampling but also the packaging requirements of the incoming image.

The synchronization of the incoming frame is achieved using polling techniques. Interrupt management is avoided to meet the real time down-sampling requirements. In VGA resolution with pixel clock frequency of 5.333 MHz, using interrupts of this microcontroller would waste almost the entire processing power only for the down-sampling of image. So, it is convenient to use polling techniques in this application.

For down-sampling, packaging and saving the packed image to the RAM, First In First Out (FIFO) buffering techniques are used. Time-critical parts in the program are written in assembly language [6] to be able to utilize flexibility and agility of the hardware, increasing timing accuracy. The time-critical events are synchronization of the frame, reading the I/O port and creating a required delay for optimal down-sampling.

Following calculations denote the memory requirement, if we try to save whole image with 1-byte Y component for each pixel.

$$\text{Required memory} = 65 * 96 * 1 = 6240 \text{ bytes} \quad (8)$$

It is clear that we have to pack the image using an algorithm that meets the timing requirements and won't restrict our work. Since the LCD module can only display binary images, we can apply threshold to each pixel data and store it as a binary image. The binary data must be stored in bytes as 8 bits in 1 byte to achieve required packing. For 64 rows, 8 vertically arranged data bytes are used. If we consider the memory requirement after packing process, following calculations shows the result. Each bank stores 96 horizontal pixels.

$$\text{Total requirement} = 8 * 96 = 768 \text{ bytes} \quad (9)$$

So we are allocating memory of 768 bytes instead of 6240 bytes, having free memory for image processing requirements.

### 3.3. Camera Field of View (FOV)

The frame alignment is an important process for this application. First we must determine the visual coverage angle of active glasses to carry on frame adjustment. We roughly determined the visual angle of two spectacles combined as 45 degrees. In the following figure, the setup for visual angle measurements is shown.

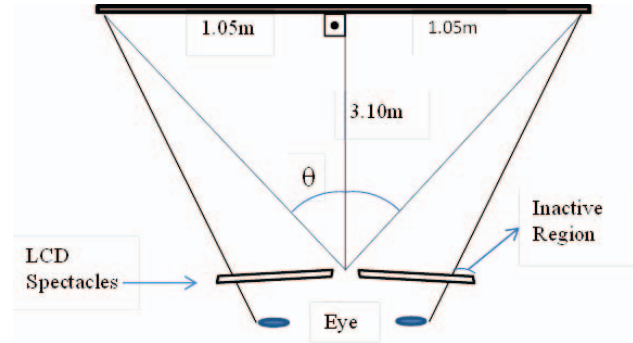


Fig. 6. Measuring the visual coverage angle of active spectacles

The angle  $2\theta$ , illustrated in Fig. 7. is the effective horizontal visual FOV angle of the camera & lens which is placed in front of the glasses between two eyes. The lens of the camera is selected so that it has a FOV of at least the size of this angle.

$$2\theta = 2 * \tan^{-1}(1.05/3.20) \cong 36 \text{ degrees} \quad (10)$$

The image captured by the camera must be aligned with the scene visible in the LCD module. Fig. 7. shows the setup used to experimentally calculate the FOV of lens.

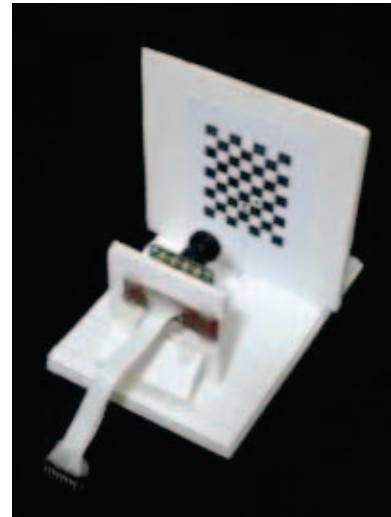


Fig. 7. Calibration setup

The binary data is transmitted to a computer via serial interface, and recorded as an image file after unpacking operations.



Fig. 8. Recorded image

In Fig. 8, the side length of each square on the grid pattern is about 0.68 cm. The camera distance from the grid is 4.1 cm. Using this setup, we can calculate the actual horizontal and vertical visual FOV angles of the camera. It is clear from the output image that, camera can display about 6 squares horizontally and 4.5 squares vertically.

$$\text{Vertical length} = 4.5 * 0.68 = 3.06 \text{ cm} \quad (11)$$

$$\text{Horizontal length} = 6 * 0.68 = 4.08 \text{ cm} \quad (12)$$

Horizontal and vertical FOV angles can be found as:

$$V. \text{ vis. ang.} = 2 * \tan^{-1} \left( \frac{3.06}{2} / 4.1 \right) \cong 41 \text{ degrees} \quad (13)$$

$$H. \text{ vis. ang.} = 2 * \tan^{-1} \left( \frac{4.08}{2} / 4.1 \right) \cong 52 \text{ degrees} \quad (14)$$

The horizontal-vertical FOV angle of selected lens meets the requirements of glasses depicted in Fig. 6. Since the optical characteristics of the camera are not identical to the characteristics of each eye, it is not possible to align real scenes with the images captured by the camera without geometric transformations. Perspective transformations are used to remap image points so that they are aligned with the actual scene points with marginal errors.

There are two separate LCD modules used for right and left eye and only one central camera for both. It is appropriate to use two microprocessors for left and right spectacles because this provides flexibility and parallel processing of two different perspective transformations at the same time. In fact, the memory requirement alone makes it obligatory to use two separate ATmega328 chips for two LCD modules.

### 3.4. Perspective Correction

Using the assumption that objects in the scene are far away, and ignoring lens distortions, we can relate camera view and “through-the-glass eye view” using perspective transformation. Following projection matrix equations are used to apply perspective transformation [7], [8].

$$\begin{bmatrix} X' \\ Y' \\ W \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (15)$$

$$x' = \frac{X'}{W} \quad y' = \frac{Y'}{W} \quad (16)$$

$$x = \frac{ax' + by' + c}{gx' + hy' + 1} \quad (17)$$

$$y = \frac{dx' + ey' + f}{gx' + hy' + 1} \quad (18)$$

There are 8 unknowns in the transformation matrix, so 4 points provide enough information to solve for the parameters of perspective transformation. Equations can be rewritten as follows [8] using four point correspondences (*i.e.*  $(x_i, y_i) \leftrightarrow (x'_i, y'_i) \quad i=1,2,3,4$ ).

$$\begin{bmatrix} x'_1 & y'_1 & 1 & 0 & 0 & 0 & (-x'_1 * x_1) & (-y'_1 * x_1) \\ x'_2 & y'_2 & 1 & 0 & 0 & 0 & (-x'_2 * x_2) & (-y'_2 * x_2) \\ x'_3 & y'_3 & 1 & 0 & 0 & 0 & (-x'_3 * x_3) & (-y'_3 * x_3) \\ x'_4 & y'_4 & 1 & 0 & 0 & 0 & (-x'_4 * x_4) & (-y'_4 * x_4) \\ 0 & 0 & 0 & x'_1 & y'_1 & 1 & (-x'_1 * y_1) & (-y'_1 * y_1) \\ 0 & 0 & 0 & x'_2 & y'_2 & 1 & (-x'_2 * y_2) & (-y'_2 * y_2) \\ 0 & 0 & 0 & x'_3 & y'_3 & 1 & (-x'_3 * y_3) & (-y'_3 * y_3) \\ 0 & 0 & 0 & x'_4 & y'_4 & 1 & (-x'_4 * y_4) & (-y'_4 * y_4) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \quad (19)$$

The  $(x,y)$  coordinates are taken from the displayed image, and  $(x',y')$  coordinates are taken from the intersection of LCD plane and view direction. Four points are marked on the LCD screen. The  $x$  and  $y$  coordinates are measured with respect to the top left corner of the LCD's active region. Fig. 8. demonstrates the points used for the right LCD.

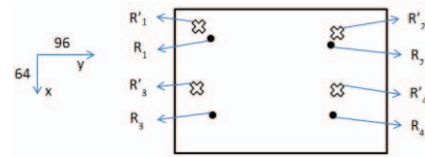


Fig. 9. R and R', old and new coordinates for the right LCD

$$R_1 = (16, 21) \rightarrow R'_1 = (7, 11) \quad (20)$$

$$R_2 = (18, 84) \rightarrow R'_2 = (11, 87) \quad (21)$$

$$R_3 = (59, 21) \rightarrow R'_3 = (36, 7) \quad (22)$$

$$R_4 = (59, 84) \rightarrow R'_4 = (39, 87) \quad (23)$$

These parameters are used to project camera image to the right spectacle plane. Using the reference points the transformation parameters for the right LCD are found as:

$$a = 1.6026 \quad b = -0.0559 \quad c = 5.6030 \quad (24)$$

$$d = 0.1544 \quad e = 0.8398 \quad f = 10.9519 \quad (25)$$

$$g = 0.0018 \quad h = 1.9 * 10^{-6} \quad (26)$$

The transformation of the image for left LCD is achieved similarly using different point correspondences. Following locations shows the old and new points of left LCD transformation.

$$L_1 = (19, 21) \rightarrow L'_1 = (9, 34) \quad (27)$$

$$L_2 = (22, 81) \rightarrow L'_2 = (12, 82) \quad (28)$$

$$L_3 = (52, 19) \rightarrow L'_3 = (49, 30) \quad (29)$$

$$L_4 = (57, 80) \rightarrow L'_4 = (57, 93) \quad (30)$$

The projection parameters of left LCD are found as follows.

$$a = 1.2478 \quad b = -0.0030 \quad c = 9.0571 \quad (31)$$

$$d = 0.2173 \quad e = 1.3458 \quad f = -25.4040 \quad (32)$$

$$g = 0.0071 \quad h = -5.7 * 10^{-5} \quad (33)$$

The images are remapped using the precalculated perspective transformation parameters and displayed on the LCD screens.

#### 4. Encountered Difficulties and Experimental Results

While implementing active glasses, not only hardware and software design challenges but also some optical problems are experienced which require further research. The use of an LCD display as active glasses revealed another concept called “optical diffraction” [9]. The vertical and horizontal spaces between two successive rows and columns of LCD form a square aperture. When the LCD is darkened, the diffraction pattern becomes visible on the screen. Following figure shows a square aperture diffraction pattern.

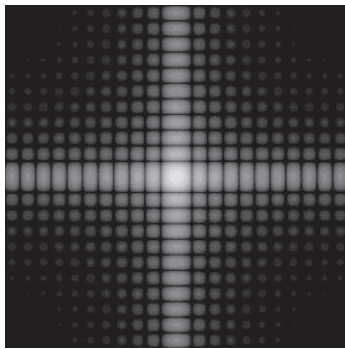


Fig. 10. Square aperture diffraction pattern

This problem must be investigated to achieve better results for active spectacle applications. Different LCDs with higher filling factors can reduce the effect of diffraction. Another approach can be the analysis of square aperture in the frequency domain. This kind of analysis may lead to construction of different dimming patterns which may reduce the diffraction visibility. For further development of active glasses, this concept must be revisited.

Another critical case is the locating the dimming pattern to the right place with an appropriate size. Since the pupil of the eye changes size according to the light conditions, we have to determine the size of the darkening object adaptively. This process is another design challenge and requires dynamic control algorithms. In this prototype design, the adaptation to different illumination levels is achieved by the internal DSP of the camera which performs histogram based automatic exposure control.

In this study, a prototype design and implementation of active glasses are completed. Real time localization of bright light sources is achieved and dimming is applied to corresponding points on the LCD spectacles. A more detailed calibration for image projection parameters must be carried out to achieve precise results.

#### 5. Conclusions

This work proposes a new solution for light adaptation problems of human eye. The disturbing effect of intense light sources is reduced using LCD glasses. The usage of transparent LCDs as eyeglasses is a new approach and it can be used for wide range of applications. This concept requires not only a comprehensive research in the fields of optics and electronics but also manufacturing technologies in these fields.

In this project, active glasses design is implemented using simple components. Two dot matrix LCD modules, one on board VGA camera and two microcontroller breakout boards are

attached on standard laboratory glasses to form a prototype design. In Fig. 11., our active glasses implementation is shown.

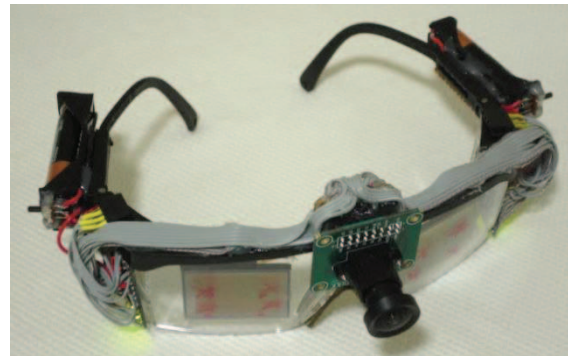


Fig. 11. Active eye-glasses design

The hardware design is made intentionally as simple as possible and for convenience the components are easy to find. Hence the cost of the project is very low compared to the similar vision based projects. The main contribution of this work is the implementation of a practical and original vision problem on a hardware with minimal capabilities. In this case, both the hardware and software design must be efficient and compatible to produce reliable outputs with sufficient quality.

For future applications, this design can be adapted to different fields. With inclusion of proper lenses, it can be utilized to display text messages or images on the screen. For medical purposes, it is possible to re-design it for people who have light adaptation problems (e.g. people suffering from ‘macular degeneration’ disease).

#### 6. References

- [1] Texas Instruments, “Use of the CMOS Unbuffered Inverter in Oscillator Circuits”, Application Notes, 2004.
- [2] MXCOM INC., “Crystal Oscillator Circuit Design”, Application Notes, 1997.
- [3] ATMEL, “8-bit AVR Microcontroller ATmega328”, Datasheet, 2010.
- [4] OmniVision, “Serial Camera Control Bus (SCCB) Functional Specification”, 2007.
- [5] (2005), Arduino, “Open-source Hardware and Software Prototyping Platform”. Available: <http://www.arduino.cc/>
- [6] ATMEL, “8-bit AVR Instruction Set Nomenclature”, Application Notes, 2010.
- [7] R.C. Gonzales, R.E. Woods, “Digital Image Processing”, edited by Addison Wesley, Boston, USA, 1993.
- [8] G. Walberg, “Digital Image Warping”, Wiley-IEEE Computer Society Pr., Hoboken, USA, 1990.
- [9] E. Hecht, “Optics”, Addison Wesley, Boston, USA, 2001.