HALO HOLOGRAM SOFTWARE

Denis Džačko[1] Samuel Polakovič[2] Filip Suchán[3] Erik Kaľavský[4] , Gregor Rozinaj[5]

[1]Faculty of Electrical Engineering and Information Technologies, Institute of Multimedia Information and Communication Technologies, Ilkovicova 3, 812 19 Bratislava, Slovakia

kalavsky.erik6@gmail.com

Abstract. Today's communication systems offer limited possibilities of image processing and subsequent its presentation. Most systems offer an image that is presented only in 2D. With a 2D image, the third dimension disappears, which is the depth of the image. In this article, we deal with a system with the ability to transform an image from 2D to 3D while maintaining in-depth information. This creates a 3D impression and improves the quality of communication, because we create the impression that the person we are communicating with is in the room with us, even if this is not true. Such a system requires real-time hardware and software implementation. In our case, we deal with both hardware implementation and software solution in real time.

Keywords: Atmel, connection, LED strip, HALL-effect sensor, small rotate engine, USB, Graphic user interface, stream, IP camera

1 Introduction

Today, information technology [1] is increasingly used in various fields and on various occasions. These technologies are especially used between people for audio or video communication. This communication is largely restrictive and insufficient, due to the transmission of only the audio signal during telephone calls or the transmission of only the 2D image during video transmission.

In video communications, the depth of the image is lost, giving the impression that the communication is taking place with a moving 2D image on the screen. In order to achieve the perception that the person with whom we communicate through information devices is next to us, in-depth information is one of the most important parameters. To preserve this information, various algorithms are created [2] how to preserve this information and then present it in order to create an image with a 3D effect.

In this work we will focus on how to create such an image, then process it, distribute it over the Internet, and additionally processing such an image where the result will then be presented in 3D.

2 BASIC PROPOSAL

1. Our hardware design is based on the Halo hologram concept [3]. Halogram is a tool for displaying an image that appears as a 3D image. Such an image gives us the

impression that the person with whom he communicates via video communication is next to us, even though he is not in the room with us.
In addition to the hardware design, software image processing is required to make the created image as realistic as possible.

The design consists of two main parts and these are the software implementation and the hardware implementation. Figure 1 shows the basic design concept.

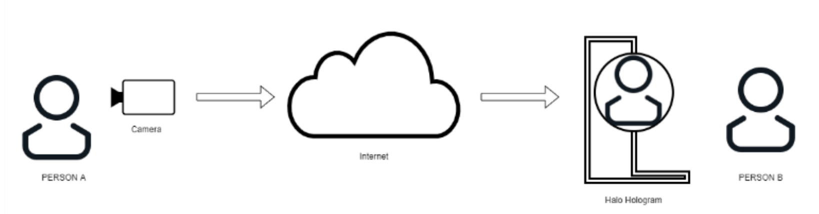

Figure 1 Basic scheme

In Figure 2 it is possible to see the basic blocks from the image capture, then its processing and display.

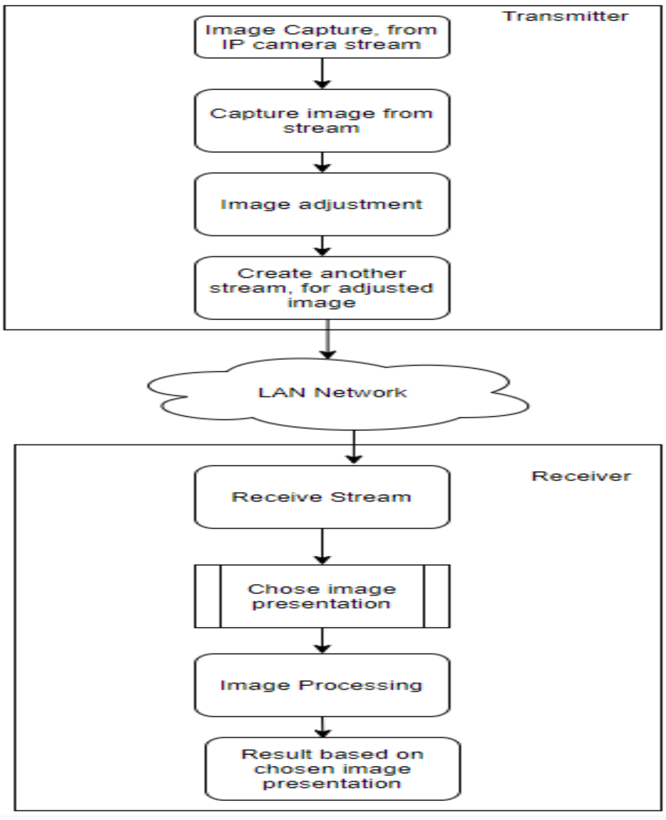

Figure 2Elementary blocks of realization

We will analyze the individual blocks in Figure 2 in more detail and approach the principle of operation.

3 Transmitter

In this section, we discuss the methods by which an image can be captured, subsequently edited, and sent over a LAN to a receiver. This part is software implemented in the C ++ programming language, which has been extended with a library for OpenCv image processing and also with a library for G-Streamer streaming.

1. Image capture

For image capture, we chose an IP camera that can be easily modified and customized as required. The main requirements are the size of the scanned image and the frame rate per second. Another advantage of IP cameras is that it creates its own stream of transmission that is easy to capture and then work with the captured screens.

We are also able to use a WEB camera, which is part of every computer, instead of an IP camera.

2. Capture the image stream

In this part we capture individual image frames, whether they are from an IP camera or a WEB camera in real time.

3. Image adjustment

In this block, the individual captured image frames are resized to the required size so that they can be subsequently sent over the network

.4.Create a new stream with the edited image

After capturing and editing each frame in real time, we will create a new separate stream of such an image. The new stream is created via TCP confection on a server with a dedicated free port. In our case, we work in one closed LAN network, so the IP address of the local computer serves as a server.

Such a stream can also be captured by third-party applications after entering the correct URL of the line on which the stream is presented. The newly created stream has a delay of approximately 2-3 seconds

4 Reciever

The main functionality of the software solution is in the receiver because, here the choice is made onwhich device the given image will be presented.

4.1 Receiving the stream from the transmitter

In this section it is necessary to enter the required parameters for reception and this is the IP address of the server and also the port on which the stream will be received. Subsequently, a URL address is generated based on the selected parameters for receiving the stream from the transmitter.

4.2 Display selection

This part serves in what form the image should be displayed. In our software implementation, we have more options than the resulting image can be displayed. Since some parts are still in development, we decided to point out two bases that are implemented.

4.3 Pyramid image display

In this view, the image will be divided into four quadrants as shown in Figure 3, with the same image being displayed in each quadrant.

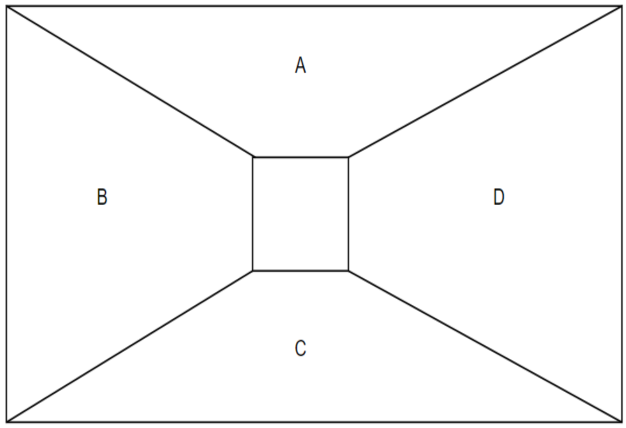

Figure 3split of screen A,B,C,D quadrant

Subsequently, in the middle of the placement, the pyramid that will reflect the image will be created, adding that the image will float inside. The image on each side will be the same. The location of such a pyramid is shown in Figure 4.

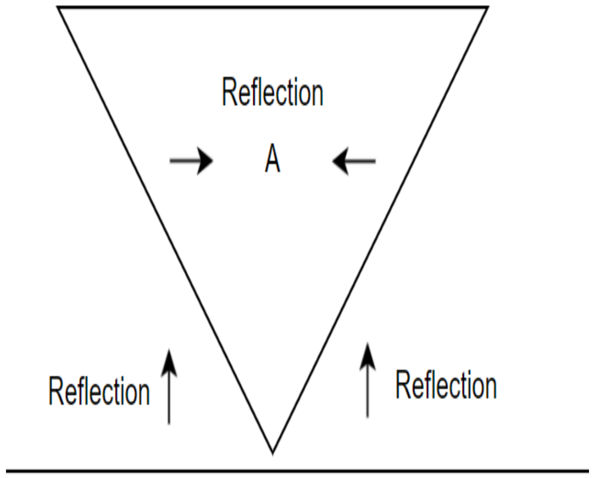

Figure 4Display mirrored image

4.4 Display on Halo hologram

The display on the globe model will be described in more detail in Chapter V Hardware Implementation

4.5 Image processing

After selecting the display, real-time image processing then takes place. In this section, each farm received is adjusted according to the selected display option. Each display option requires specific image adjustments.

4.6 Image display based on the selected method

After receiving and processing the image, the resulting image is then presented on a display device. In the resulting image, the depth of the image is preserved and a 3D impression is created.

5 Software implementation of the Hardware part

To display the image in the best possible quality, we built a circle that is seated in the base. The display of a given image is created by a rapid rotation of a circle around its own axis. On the circle are the location of the LED strip that creates the image. The hardware implementation can be divided into three parts, which are the transfer of data from the receiver to the microcontroller, editing the received data and then sending them to the display unit, displaying the data on the built circle. These parts are shown in Figure 5.

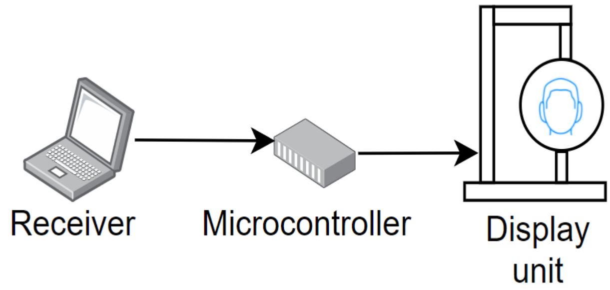

Figure 5Elementarry scheme of hardware realization

5.1 Data transfer from the receiver to the microcontroller

After image processing on the receiver side, one frame is then sent via the serial USB connector to the motherboard of the Atmel UC3-A3 Xplain microcontroller.

After establishing a connection between the microcontroller and the receiver, each frame of the received image is then sent to the microcontroller in blocks. One block is a 120x65 nut.

For one such picture frame it is necessary to send 3 picture blocks. This is because the resulting image contains all 3 RGB color components and each component is sent separately. The transmission speed is 125000 Kbaud, which is not enough speed for the given image to be received in a sufficiently short time for its further processing. We could not overcome this speed.

5.2 Editing the received data and then sending it to the display unit

This part is the most important part of the hardware implementation, even though it is performed on the microcontroller itself. Due to the control of the display unit itself. After receiving the input data, a 120x65x3 matrix is created, which is the resulting image matrix, which will then be projected on the display unit.

The resulting image matrix is then sent via the SPI interface. Sending speed is 30Mhz / 1s. The resulting array is sent in blocks of 4 bytes. A 32-bit start packet is sent to establish a connection with the LED strip. Subsequently, the matrix is sent one pixel at a time. Since the size of the sent block is 32 bits, the first 8 bits indicate the basic properties so that the first 3 bits are controllable, 5 bits indicate the intensity at which the LED on the LED strip should light up. The intensity range of the LED strip is from 0 - 31, in our solution we use the maximum intensity of 2 because we use a very bright LED strip. And the remaining 24 bits represent information about the pixel itself in RGB.

The main feature of the display unit is the rotation of the circle. Timers are used for the correct rotation of the circle, which will ensure synchronization based on the frequency of rotation.

The display itself is by means of a slice counter strip which has a timer set to 120 seconds. Subsequently, the resulting image matrix is read from the beginning to the end and also at the same time from the end to the beginning. This is so that the resulting image is on both sides of the display unit.

5.3 Display unit

The display unit is constructed of commonly available material. This display unit is shown in Figure 6. In which the image is projected.

Figure 6Display Unit

Due to the limited transmission speed between the receiver and the microcontroller, the resulting image is insufficient and incomplete. Therefore, the main functionality was tested on a predefined still image matrix which was then sent to the display unit.

6 System improvement options

To achieve better results, it is necessary to increase the baud rate for sending data between the receiver and the microcontroller. The proposed speed is 3000000 Kbaudov / 1s. At this speed, the resulting image should be high quality without imperfections.

Another possibility to achieve better results is to use another microcontroller such as RasberryPi or possibly Arduino. These microcontrollers are able to achieve higher speeds and also have a higher memory capacity.

It is also possible to improve the system from a network point of view by adding a publicly available server and communication will not be limited to the LA network.

7 Result

The system can obtain a stream from an IP camera and then, after editing and selected options, present the image on display units.

7.1 Receive and create stream

As can be seen in Figure 7, we obtained the stream from the IP camera and then distributed it over the LAN and displayed it in a third-party application.

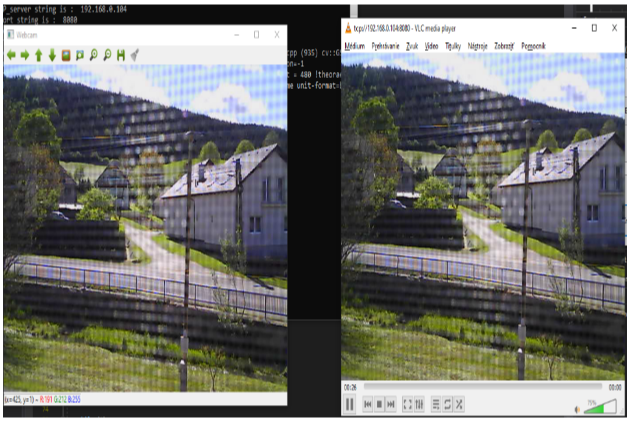

Figure 7Obtain a stream from the IP camera on the right, and then create a new stream on the left

7.2 Displayed in pyramide

After selecting the pyramid display option, one frame is divided into 4 identical quadrants as shown in Figure 8.

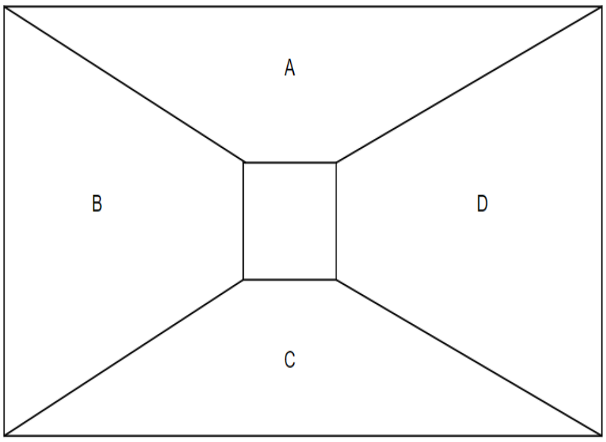

Figure 8Image obtained from an IP camera and then ready for pyramid display

The image thus prepared is then streamed and can be captured via a third-party application on the mobile. Such a representation is shown in Figure 9, with each page presenting the same image. If necessary, it is possible to present a different image on each page

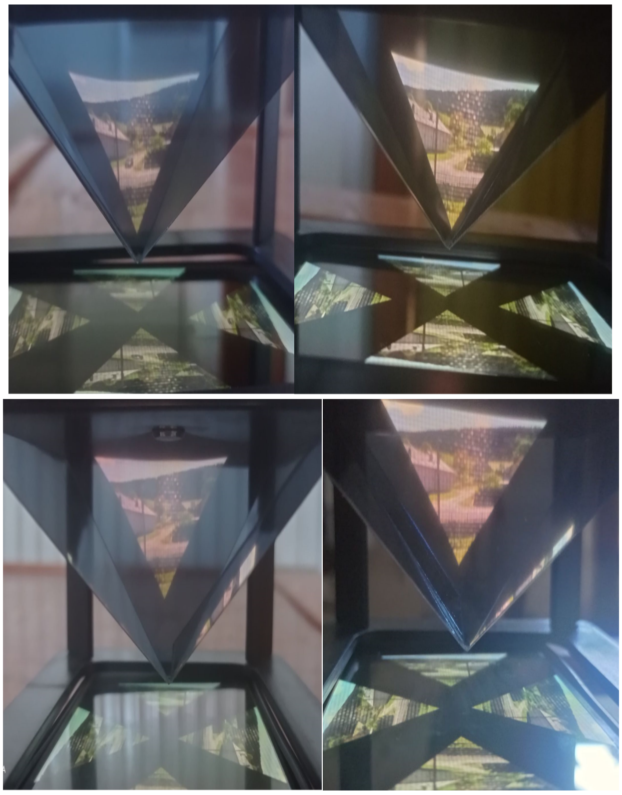

Figure 9age in holographical pyramide from all sides

7.2 Display on rotating circle

In this section, we were the first to test the functionality of this display unit. In Figure 10 and Figure 11 it is possible to see the testing of the functionality on one color and subsequently on two colors. This confirms that the display unit is working properly..

Figure 10. Right figure testing one color on display unit, left figure testing with 2 colors.

After verifying the functionality, we tested a predefined static image, which was then sent to a rotating circle. The main problem in this part was timing because we used Halo sensors that were not enough and created delays. Due to the delay, the image was spread over the entire circle as can be seen in Figure 11.

Figure 11 Face display during time synchronization with Halo sensors.

Until a stabilized image is achieved, we then tried to project the human head, which should be the main result of this system. This result can be seen in Figure 13 and also in Figure 14.
Subsequently, we used timers as functions instead of Halo Sensors. In this way, we were able to stabilize the desired image and direct it directly to the display circle.

Figure 12Stabilized image display

Figure 13Final Display of face

cameras, communication within one VLAN network. We also managed to achieve results in the field of imaging, such that we can display an image from memory for our Halo hologram. Of course, in conclusion, we would like to add that even though we have achieved many tangible results in our project, there is room for further development and improvement. As some, we would like to mention image processing from scanning devices such as napríklad Azure Kinect, Data transfer not only in one network but globally. Next, it is necessary to solve the problem of loading speed of sent data for display on the hologram. The greatest room for improvement occurs in the display where it is necessary to further work on the display of not only the stationary image but the real time video image itself to create quality communication technology of the future.

Acknowledgment

This publication was created thanks to support under the Operational Program Integrated Infrastructure for the project: International Center of Excellence for Research on Intelligent and Secure Information and Communication Technologies and Systems - II. stage, ITMS code: 313021W404, co-financed by the European Regional Development Fund

References

[1] EGBU, Charles O.; BOTTERILL, Katherine. Information technologies for knowledge management: their usage and effectiveness. Journal of Information Technology in Construction (ITcon), 2003, 7.8: 125-137.

[2] FEHN, Christoph. A 3D-TV approach using depth-image-based rendering (DIBR). In: Proc. of VIIP. 2003.

[3] SUCHÁN, Denis Džačko1 Samuel Polakovič2 Filip; KAĽAVSKÝ, Erik; ROZINAJ, Gregor. HALO HOLOGRAM HARDWARE. 2021.

[5] BARANNIK, Vladimir, et al. The video stream encoding method in infocommunication systems. In: 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET). IEEE, 2018. p. 538-541.

Redžúr 2021

PROCEEDINGS

15th International Workshop on Multimedia Information and Communication Technologies

Edited by:
Ivan Minárik
Gregor Rozinaj

STU FEI

ZDRUŽENIE POUŽÍVATEĽOV TELEKOMUNIKÁCIÍ SLOVENSKA

ISBN 978-80-227-5113-1

SPEKTRUM STU

PROCEEDINGS
Redžúr 2021

15th International Workshop on Multimedia Information and Communication Technologies

4 June 2021, Bratislava, Slovakia

EDITED BY:
Ivan Minárik
Gregor Rozinaj

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering and Information Technology
Institute of Multimedia Information and Communication Technologies
Ilkovičova 3
812 19 Bratislava
Slovak Republic

Published by:

Vydavateľstvo SPEKTRUM STU of the Slovak University of Technology in Bratislava

in cooperation with

Združenie používateľov telekomunikácií Slovenska

ISBN 978-80-227-5113-1

General Chair

Gregor Rozinaj

Slovak University of Technology, Bratislava, Slovak Republic

Program Committee

Aura Conci

Fluminense Federal University, Brasil

Mislav Grgic

University of Zagreb, Croatia

Gerhard Gruhler

Heilbronn University, Germany

Jaromír Hrad

Czech Technical University, Czech Republic

Juraj Kačur

Slovak University of Technology, Slovakia

Jarmila Pavlovičová

Slovak University of Technology, Slovakia

Pavol Podhradský

Slovak University of Technology, Slovakia

Markus Rupp

Vienna University of Technology, Austria

Yevgeniya Sulema

National Technical University of Ukraine, Kyiv Polytechnic Institute, Ukraine

Radoslav Vargic

Slovak University of Technology, Slovakia

Tomáš Zeman

Czech Technical University, Czech Republic

Branka Zovko-Cihlar

University of Zagreb, Croatia

Review Committee

Abreu, Raphael
Aguilera, Cristhian A.
Araújo, José Denes
Aung, Zeyar
Bergamasco, Leila
Bergo, Felipe
Bezerra, Eduardo
Bozek, Jelena
Bravenec, Tomas
Bujok, Petr
Burget, Radim
Casaca, Wallace
Ciarelli, Patrick Marques
Copetti, Alessandro
Costa, Tales Fernandes
Čepko, Jozef
Davídková Antošová, Marcela
Devamane, Shridhar
Galić, Irena
Gonçalves, Vagner Mendonça
Grgic, Sonja
Habijan, Marija
Haddad, Diego Barreto
Henriques, Felipe
Hocenski, Zeljko
Hrad, Jaromír
Jakóbczak, Dariusz Jacek
Juhár, Jozef
Kačur, Juraj
Karwowski, Damian
Kominkova Oplatkova, Zuzana
Körting, Thales Sehn
Kos, Marko
Kultan, Matej
Laguna, Juana Martinez
Latkoski, Pero
Lima, Alan
Londák, Juraj
Lopes, Bruno
Lopes, Guilherme Wachs
Lourenço, Vítor
Malajner, Marko
Mandic, Lidija
Marana, Aparecido Nilceu
Marchevský, Stanislav
Marinova, Galia
Markovska, Marija
Matos, Caio
Medvecký, Martin
Minárik, Ivan
Mocanu, Stefan
Mustra, Mario
Nyarko, Emmanuel Karlo
Paiva, Anselmo
Papa, Joao Paulo
Podhradský, Pavol
Polak, Ladislav
Prinosil, Jiri
Rakús, Martin
Rodriguez, Denis Delisle
Rozinaj, Gregor
Rybárová, Renata
Silva, Aristófanes
Silvestre, Santiago
Slanina, Martin
Sousa De Almeida, Joao Dallyson
Sousa, Azael Melo E
Stasinski, Ryszard
Tcheou, Michel
Toledo, Yanexis Pupo
Trúchly, Peter
Turan, Jan
Vargic, Radoslav
Veras, Rodrigo
Vitas, Dijana
Vlaj, Damjan
Vukovic, Josip
Wajda, Krzysztof
Zamuda, Ales
Zeman, Tomas

Organizing Committee

Juraj Londák
Slovak Republic

Ivan Minárik
Slovak Republic

Šimon Tibenský
Slovak Republic

Marek Vančo
Slovak Republic

Table of Contents