

A Novel Cross-Dimensional Image Processing Technique: Toward a 3D View of a 2D Image



Dipnarayan Das and Sumit Gupta

Abstract The domain of image processing has been very popular among researchers since a very long time. It is a well-known fact that there are myriad objectives that can be met with this subject. Through this paper, we are proposing a novel technique whereby we will train our system by processing images on a two-dimensional plane and subsequently produce a resultant three-dimensional view of an image. We have attained this transformation through cross-dimensional image processing by studying, understanding and analyzing the various attributes and properties of an image. This cross-dimensional image processing technique explores the existing digital image processing domain where the inputted 2D image is mapped to produce a 3D view of the image in 3D space.

Keywords Image processing · Z-Buffer · Depth of field · LED cube · Surface detection

1 Introduction

In the field of computer science, digital image processing is a well-known term, both in the research community as well in the academic circle. It is one of the most crucial categories of digital signal processing, capable enough in dealing with distortions and noise that occur when images are processed. Since images are defined over two dimensions (or perhaps more), digital image processing is generally modeled as multidimensional systems. But post-processing, what we observe is the change in the image attributes, keeping the dimension of the image unaffected.

D. Das (✉) · S. Gupta
Department of Computer Science & Engineering, University Institute of Technology,
The University of Burdwan, Golapbag (North), Burdwan 713104, West Bengal, India
e-mail: dipnarayan.das35@gmail.com

S. Gupta
e-mail: sumitsayshi@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
S. Bhattacharyya et al. (eds.), *Recent Trends in Signal and Image Processing*,
Advances in Intelligent Systems and Computing 922,
https://doi.org/10.1007/978-981-13-6783-0_8

The primary difference between 2D and 3D objects is that 3D objects have three dimensions, viz. height, width, and depth, whereas 2D objects have only two dimensions, viz. height and width. The real-world objects are three-dimensional in nature because they have depth as an extra dimension along with height and width. A drawing on a paper is often 2D, but on analyzing an image from a linear perspective, a 2D image can be made to appear as having three dimensions. So, what is crucial and worth mentioning over here is that even if the linear perspective helps to create a 3D view of a 2D image, yet the resultant image lacks the z -value or the depth value.

Through this paper, we are proposing a novel cross-dimensional image processing technique for obtaining a 3D view of a 2D image. Our system is based on a hardware layer where the main microcontroller is Arduino Mega. As we know, open source microcontrollers do not have an inbuilt library for image processing, so we have developed a novel indigenous algorithm that detects the color spaces for converting the 2D images into a 3D physical (e.g., 3D hologram) view.

The rest of the paper is organized as follows: Sect. 2 discusses the previous related works. In Sect. 3, we have proposed our methodology in terms of architecture, system components required, working principle, and theories related to the concept of cross-dimensional image processing. Section 4 discusses the implementation part and the results so obtained. Section 5 analyzes our work through a discussion on the merits and demerits of our technique. Section 6 concludes the paper by pointing out some future scope of improvements that can be taken up.

2 Previous Related Works

We have studied that in most of the existing systems [1–3], the stereoscopic vision technique is generally used to capture the depth of real-time view, and/or depth value is used for constructing the 3D shape. In papers [4, 5], the authors have proposed 2D-to-3D conversion algorithms that fairly deal with the depth quality and computational complexity but suffer due to the limited size of the training dataset. Some existing systems [6, 7] are in fact manual, cumbersome and lack automation. Moreover, all the current systems utilize the system-defined or inbuilt library functions, but we have developed a new image processing library compatible with low-performance microcontrollers like Arduino.

3 Our Proposed Methodology

3.1 System Architecture

See Fig. 1.

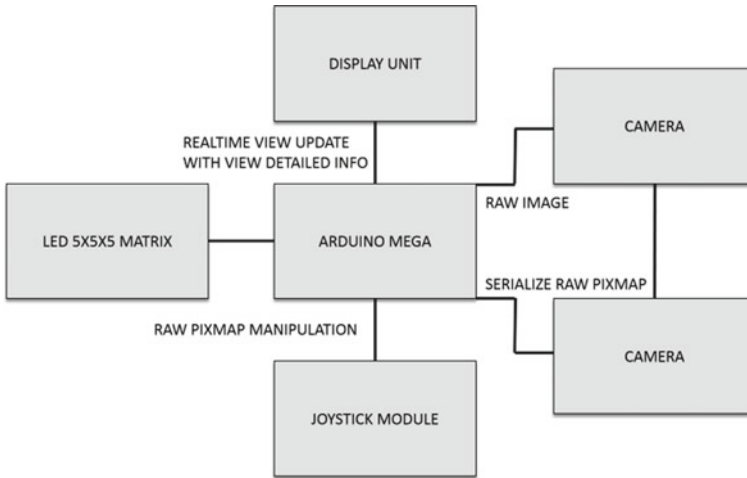


Fig. 1 Architecture of our proposed system

3.2 System Components

Our proposed system (see Fig. 2) comprises the following components and/or modules:

1. **Arduino Mega:** The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 [8].
2. **TFT Color LCD:** As we know that interfacing a TFT LCD with an Arduino is pretty easy, so we require an Arduino Board and a 2.4 in. TFT Shield to manage the hardware section and an Arduino IDE and the TFT Library for managing the software section. Several libraries exist for operating upon the TFT shield, but we need to identify that particular TFT driver out of multiple available TFT LCD drivers for which we will install our library to serve our purpose [9].
3. **Joystick Module:** The joystick module has 5 pins, viz. VCC, Ground, X, Y, and Key. The thumb stick is analogous in nature and provides more precise readings compared to the other joysticks with buttons and/or switches. Further, the joystick can be pressed down for activating a ‘press to select’ push button [10].
4. **5 × 5 × 5 LED Matrix:** The light emitting diodes (LEDs) have been arranged as 5 × 5 matrices in five different planes. The number of LEDs can be extended as per need.

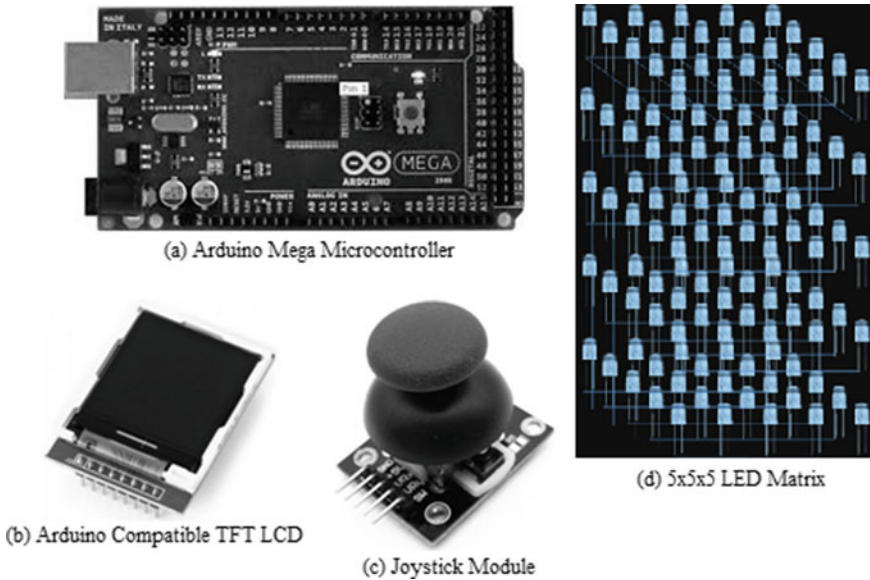


Fig. 2 Components of our proposed system

3.3 Working Principle

Our proposed system works as follows: Firstly, the raw image of any object will be entered into the Arduino Mega with raw (or unchanged) format after acquiring the image directly from any standard camera. As an alternative, we can also send serialized data which will be further formatted by the Arduino microcontroller. Further, as we know that Arduino does not have any inbuilt image processing library, so we have created a library termed as color space library, which is compact because microcontrollers have very less amount of built-in storage space. The color space library is able to detect a color region out of the three regions for 16,777,216 types of colors. We have used a specialized technique to detect the colors (that works in the same fashion as the boundary-fill algorithm in graphics designing for detecting the sudden change of values) and the variable rate of three zone colors. Our proposed library has the following features:

1. Two-dimensional image/space to 3D space conversion.
2. Texture detection and object boundary evaluation.
3. Depth recognition.
4. Real-world live streaming from 2D to 3D.
5. Object symmetry detection and rest production.
6. Color space detection.

In our work, in lieu of using a camera module, we are using a mobile phone to capture a real-world image. The pixmap is converted into serialized data and then sent

to the Arduino. The serialized data is also in raw format, and the actual processing will be done on the raw image. The Arduino Mega is attached to a TFT LCD for providing all the information about the raw image and its transmission, viz. transmission rate, image resolution, FPS (frames per second), PPI (pixels per inch), etc. The joystick module is used to simulate the image in TFT LCD so that operations such as cropping, resizing, texturing, and zooming can be performed while converting an image from 2D to 3D mode. The 3D object will be shown from the 2D plane onto the $5 \times 5 \times 5$ LED cube. The LED cube is in the monochrome mode, and so, all the pictures or video that are captured will be threshold by the microcontroller during the time of processing. For depth analysis, the LED intensity is variable as well as the level of the cube is changed.

3.4 Related Theories

1. **Surface Detection:** In graphics designing, we have heard about the visible surface detection methods, namely the object-space method (implemented in the physical coordinate system) and the image-space method (implemented in the screen coordinate system). We are not concerned about the object-space method because in the input panel itself, the 3D coordinate system is already used. But in case of the image-space method, there is one method (depth-buffer method) which can recognize the depth and by that detection, it removes the back faces.
2. **Depth-Buffer Method (or Z-Buffer Method):** This method compares surface depths at each pixel position on the projection plane. An object's depth is generally measured from the view plane along the Z-axis of a viewing system. It requires two buffers, viz. the image buffer and the Z-buffer (or the depth buffer), each having the same resolution as the capturing image [11]. During the process, the image buffer stores the color values of each pixel position, and the Z-buffer stores the depth values for each (x, y) position. But here the system using a 2D image has all the z -values for each pixel which helps us to determine the actual back and front faces.
3. **Image-specific attributes:** We observe that for a 2D image, the Z-buffer method will not work to display in 3D space. Now to resolve this 2D image to 3D view objective, our system focuses on few image-specific attributes, viz. image pixel values, rate of change of color values, color region detection, boundary detection, and object symmetry detection.
4. **Depth of Field:** Depth of field is basically the distance between the nearest and the farthest objects in a photograph that appears to be acceptably sharp. It is known that a camera can focus sharply on only one point. Scientifically, it is based on the circle of confusion. Using these properties, every object is considered individually, and also the whole image is converted into a 3D object whose properties are structured into a table and depending upon the user activity, different types of 3D image-cum-night hologram are made in the LED cube. Based on the LED cube configuration (see Fig. 3a), the structured values made from the image are

again re-formatted to fit into the display by image processing operations like cropping, resizing, etc., and according to that values, the microcontroller sends the monochromatic color lookup table to the display controller to show the 3D space information as well as to activate the digital pins of the LED cube.

This method is very effective in case when there are not enough pins in the microcontroller. But this method also suffers from a problem which is that if we connect the common lines vertically and horizontally, the level selector will be connected, and then there will be no possible way to create an object angle in the vertical direction. Only horizontal axis can be used for any inclination then. So to solve this problem, we have to use a matrix connection due to which although the number of pins will increase, but still it will be quite less when compared to a normal setup. In the matrix connection, there will be 25 pins spread horizontally as well as vertically. So, in total 50 pins will be needed to enable cent percent working of the system. In this case, we observe that there still lies the problem that there are not sufficient pins to make a connection. So here we have used 74HC595 shift registers IC (see Fig. 3b) to convert three pins to (8* number of shift registers) pins [12]. So now we can connect the whole circuit. The shift register has three control pins, namely the latch clock, the shift clock, and the data line. Here, the latch clock is used to enable the rest of the control pins. The shift clock will change the pin sequentially, and on using the data line, we can pass the data or a command to that pin. Hence, when connected, we can extend the pins from 3 to 8. And when there will be two shift registers then there will be multiple 74HC595 connection with 3–16 pins orientation allowed. It is worth noting that the LED cube can be extended into any dimension by adding an extra IC but the dimension should have an odd order for achieving an accurate range approximation.

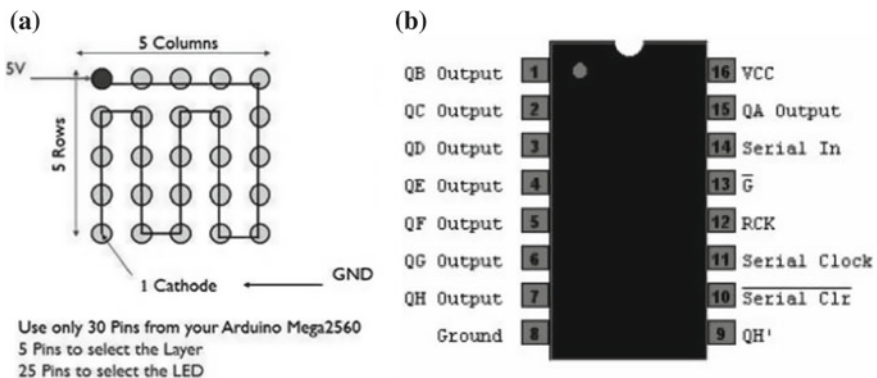


Fig. 3 a LED cube configuration, b 74HC595 IC

4 Implementation and Result

We have implemented our system to observe, test, and analyze the performance of our proposed approach. To understand the working of our system, we have initially taken the 2D view of a 3D Rectangle object (see Fig. 4a). Here, our algorithm first checks where the color deviation works and how it occurs. When the system finds that the same color space is becoming darker, then it considers it to be the depth of the object. Many other factors and conditions are also significant for mapping the object into a 3D display and to visualize it like a 3D hologram which we can see in the following image (see Fig. 4b) where the system approximates based on the fact that the object is symmetric in nature.

We have also considered one real-world image of volleyball (see Fig. 4c) downloaded from the Web [13]. This picture consisted of $201 * 251 = 50,451$ pixels in the 2D layer. Now our physical device consists of only 25 pixels in the 2D layer and there are a total of 125 pixels in the three layers. So when we have fed this image into our device, then the occupied cross-sectional area has a resolution of $5 * 5 = 25$ pixels which can be moved across the whole image using the joystick. We also have another option to scale the image to 25 pixels in order to get the $5 * 5$ pixels resolution. Depending upon the shades of the pixels, firstly our system checks for the changes in the color space. Our system analyzes on the percentage of same value pixels, new color pixel, direction of change, and how color is varying in a specific direction. Depending on the result, the data is mapped onto our physical system. Here, the spread or the percentage of the white color is maximum, so the white color is considered to be the background of the image, and the resultant view is depicted in the adjoining figure (see Fig. 4d).

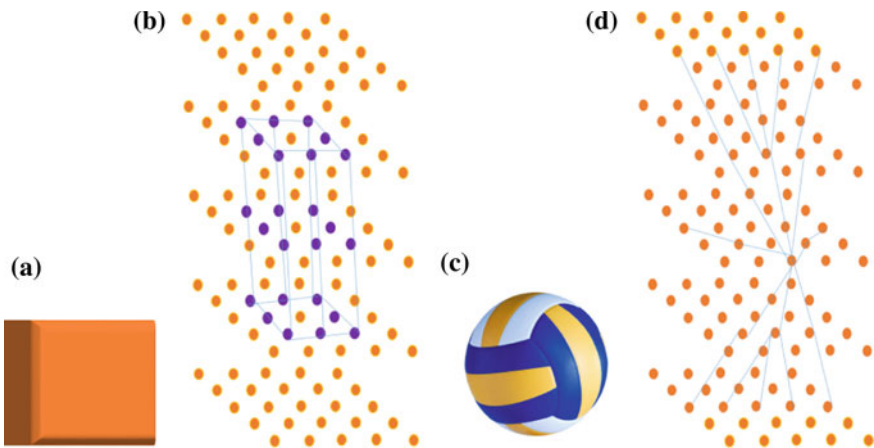


Fig. 4 a A 2D view of a 3D rectangle object, b corresponding 3D rectangle view in the LED cube, c a real-world volleyball's image, and d corresponding 3D inclined view of volleyball in the LED cube

5 Analysis and Discussion

There are various situations where our proposed approach can prove to be useful. For instance, NASA, ISRO, and other such organizations send different types of robots and intelligent devices into space for observing the nature, activity, land texture, etc. of the planets. But the device on reaching the spot transmits a continuous frame of the picture to the space control room, which is further processed to detect if there is any unknown surface contact or not. Our device can show all the land curves and offer different types of other processing in real-time 3D systems.

To watch a 3D movie we use 3D glasses. But there will be no need for any special glass or modification of the 3D movie by using our proposed device. This device can be used in defense for 3D mapping with hidden spy because during that moment, the monochrome LEDs are replaced with the infrared LEDs. Our radar system provides 2D plotting to a good extent for different objects whose position details (altitude, longitude, latitude) are shown in the display. If we split every level of this display with a specific height, then we can easily convert this device into an advanced 3D monitoring system. In ultrasonography, the imaging is in monochromatic mode situated in a 2D display. So, an observer (unlike an expert) cannot understand where the actual problem lies. On using our device, the depth analysis can be done efficiently. In endoscopy, the camera goes into our body channel for taking images of our internal system. But if we have to target any internal organ and operate on it, then we have to perform the approximation of its depth relative to the camera, resulting in the chances of getting hurt but our device can be pretty helpful.

Our proposed system provides a low cost, portable solution for performing 3D imaging for any 2D view. We can easily extend the depth of the display by adding an extra layer and by changing the value of the mapping function. So, our system is very flexible. The limitation of our system is that the LED cube is not very strong in tolerating any physical hazard or heavy jitter. Our system is currently not so effective in performing complicated computer vision processing on dynamically changing images.

6 Future Work and Conclusion

In the future, we aim to incorporate the notion of deep learning and integrate a knowledge base with our system so that any specific object can be detected from a real-time video. The resultant 3D view can then be used to observe and analyze frames that possess objects like guns or other important objects. This paper attempts to present a system that uses a novel cross-dimensional image processing technique to produce the 3D view of a 2D image. This approach can surely benefit the society and will help one think and understand differently, control his/her reflex rate and consider each scene from a bigger perspective by getting the 3D view of a 2D image.

References

1. Varga M, Jadlovsky J (2016) 3D imaging and image processing—literature review. In: 16th scientific conference of young researchers of faculty of electrical engineering and informatics technical university of Košice (SYCR) 2016. Herľany, Slovakia, pp 12–15
2. Mathai S, Mathai PP, Divya KA (2015) Automatic 2D to 3D video and image conversion based on global depth map. In: 2015 IEEE international conference on computational intelligence and computing research (ICCIC). IEEE, Madurai, India, pp 1–4
3. Konrad J, Wang M, Ishwar P (2012) 2D-to-3D image conversion by learning depth from examples. In: 2012 IEEE computer society conference on computer vision and pattern recognition workshops. IEEE, Providence, RI, pp 16–22
4. Angot JL, Huang WJ, Liu KC (2010) A 2D to 3D video and image conversion technique based on a bilateral filter. In: Proceedings of SPIE-IS&T electronic imaging, international society for optics and photonics, vol 7526. SPIE-IS&T, pp 75260D1-10
5. Galabov MN (2015) A real time 2D to 3D image conversion techniques. Int J Eng Sci Innov Technol (IJESIT) 4(1):297–304
6. Cheng CC, Li CT, Chen LG (2010) A 2D-to-3D conversion system using edge information. In: 2010 digest of technical papers international conference on consumer electronics (ICCE). IEEE, Las Vegas, NV, pp 377–378
7. Gonzalez-Huitron V, Ramos-Diaz E, Kravchenko V, Ponomaryov V (2014) 2D to 3D conversion based on disparity map estimation. In: Bayro-Corrochano E, Hancock E (eds) Progress in pattern recognition, image analysis, computer vision, and applications (CIARP 2014), vol 8827. Lecture Notes in Computer Science. Springer, Cham, pp 982–989
8. Arduino Webpage, <https://www.arduino.cc/en/Main/ArduinoBoardMega2560?setlang=en>. Last accessed 15 Aug 2018
9. CircuitDigest Webpage, <https://circuitdigest.com/microcontroller-projects/arduino-tft-lcd-interfacing-tutorial>. Last accessed 15 Aug 2018
10. Arduino Webpage, <https://www.arduino.cc/en/Tutorial/JoyStick>. Last accessed 16 Aug 2018
11. Adams A (1980) The camera. The new Ansel Adams photography series/Book 1. New York Graphic Society, Boston
12. Autodesk Webpage, <https://www.instructables.com/id/3-Arduino-pins-to-24-output-pins/>. Last accessed 15 Aug 2018
13. IndiaMART Webpage, <https://www.indiamart.com/proddetail/sports-rubber-volleyball-17443322497.html>. Last accessed 15 Aug 2018