How Varied Linear Spacing in Autostereograms Affects Time Taken to Reveal Hidden Image

Fengke Yan

Introduction

Autostereograms are a unique form of stereograms that allow for the perception of three-dimensional (3D) depth effects without the need for special viewing equipment such as a stereoscope.[1] They rely on the natural process of binocular fusion, where information from both eyes is combined to create a single coherent image.[2] It is noteworthy that some of people will unfortunately never be able to see stereograms due to binocular or stereo vision impairments such as deviations or misalignments in one or both eyes, astigmatism, or cataracts.

The generation of autostereograms involves programming techniques that manipulate factors such as repeating pattern, linear spacing that determines the perceived depth and depth map that determines the hidden image that can be revealed through proper viewing techniques.
In this report, our aim is to investigate the impact of varied linear spacing on the process of revealing hidden images in autostereograms. By examining how different spacing patterns influence the time taken to perceive the hidden image, we seek to gain insights into the role of linear spacing in the perception of autostereograms.

By understanding the relationship between linear spacing and the ease or difficulty of revealing hidden images, we can contribute to the field of visual perception and potentially enhance the design and creation of autostereograms.

Background Information

Depth Perception Relies on Binocular Vision

The left eye perceived a different vision with the right eye, it is knowns as disparity. When we focus on an object, there is a deviation of position comparing the left and right vision. Due to foreshortening, nearby objects show a larger disparity than farther objects, so disparity can be used to determine distances. The brain can automatically analyse the size of angle of disparity to obtain depth information. [3]

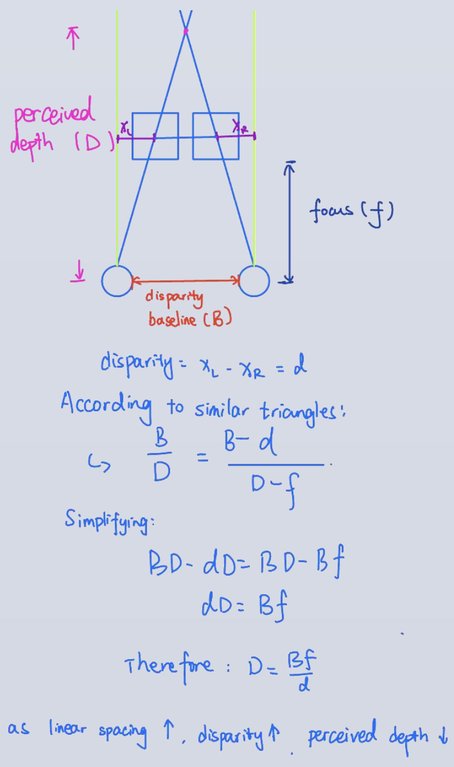

Fig. 1 The relationship between disparity and perceived depth

Autostereograms work by adjusting pixels in the repeating pattern. The repeating part will be considered as no disparity by brain, so it is perceived as 2D-image. The adjusted part creates disparity, which creates depth perception. When your eyes converge at an imaginary point behind the image, your brain matches the points seen with your left eye with a different group seen by your right eye, and you see these points lying on a plane behind the image. The perceived distance to this plane depends on the amount of spacing (linear spacing) in the pattern.[4] The relationship between the perceived depth and disparity can be derived via mathematical method shown in Fig. 1.

Methodology

Algorithm to generate autostereograms

PIL (Python Image Library) is used to read, create and alter images.

```
from PIL import Image, ImageDraw
```

Function createTiledImage(tile, dims) shown in Fig. 2.1 returns an image for the background according to the tile given and dimensions given.
Function createAutostereogram(dmap, tile) shown in Fig. 2.2 returns the autostereogram generated according to the depth map and tile given.

```
# Create a larger image of size dims by tiling the given image
def createTiledImage(tile, dims):
    # create output image
    img = Image.new('RGB', dims)
    W, H = dims
    w, h = tile.size
    # calculate # of tiles needed
    cols = int(W / w) + 1
    rows = int(H / h) + 1
    # paste tiles
    for i in range(rows):
        for j in range(cols):
            img.paste(tile, (j * w, i * h))
    # output image
    return img
```

Fig. 2.1
Implementation of the function createTiledImage

```
# with pixels shifted according to depth
def createAutostereogram(dmap, tile):
    # convert depth map to single channel if needed
    if dmap.mode != 'L':
        dmap = dmap.convert('L')
    # if no tile specified, use random image
    if not tile:
        tile = createRandomTile((100, 100))
    # create an image by tiling
    img = createTiledImage(tile, dmap.size)
    # create shifted image
    sImg = img.copy()
    # get pixel access
    pixD = dmap.load()
    pixS = sImg.load()
    # shift pixels output based on depth map
    cols, rows = sImg.size
    for j in range(rows):
        for i in range(cols):
            xshift = pixD[i, j] / 30
            xpos = i - tile.size[0] + xshift
            if xpos > 0 and xpos < cols:
                pixS[i, j] = pixS[xpos, j]
    # return shifted image
    return sImg
```

varying

Fig. 2.2
Implementation of the function createAutostereogram

Approach of altering the linear spacing

Linear spacing can be determined when creating an autostereogram. xshift (in the code) determines the shift value (linear spacing) according to the depth map. If it is deep, the shift value will be greater, so it is easier to recognise when wall-eyed view due to disparity. The divisor determines difference of shift between far and close pixels, the greater the divisor the harder to recognise. The shift (linear spacing) is proportional to the value of current pixel in the depth map. We are investigating how varied linear spacing of one specific pixel affect the depth perception. So greater divisor should have less spacing.

Research Procedure

A questionnaire is used to carried out a survey. Each interviewee was provided with three autostereogram images with the same repeating pattern but different depth maps. The autostereograms have different linear spacing of images. When they received each autostereogram and attempted to discern the hidden image, a timer recorded the time taken for them to perceive it. There is a half-minute break after every autostereogram. The recorded times were used as data to test whether the hypothesis - Decrease in linear spacing cause increase in time taken to recognise the image - can be accepted.

Statistical Analysis Approach

The statistical analysis approach employed involves the utilisation of linear regression to examine the relationship between the divisor and the time taken, measured in milliseconds, by using STATA. We already knew that as the divisor increases, the linear spacing decreases. So the objective is to assess if the divisor and the processing time shows a positive correlation (Gradient of regression line and the R-squared value [Goodness of Fit]).

Results

A total number of 132 questionnaires are collected, 90 of them are valid and 42 of them are invalid since they could not recognise two or more images. After eliminating extreme values, 250 data could be used for further analysis.

Quantitative Analysis

STATA software is used to perform regression analysis on the data collected. The scattered diagram and the regression line is drawn by package matplotlib in Python. The result is shown in Fig. 3.1 and Fig. 3.2 below:

Source	SS	df	MS		
Model	4.3974e+09	1	4.3974e+09	Number of obs	= 250
Residual	3.5903e+11	248	1.4477e+09	F(1, 248)	= 3.04
				Prob > F	= 0.0826
Total	3.6343e+11	249	1.4596e+09	R-squared	= 0.0121
				Adj R-squared	= 0.0081
				Root MSE	= 38049

time	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
Divisor	524.0828	300.7087	1.74	0.083	-68.1858	1116.351
_cons	20639.83	6310.586	3.27	0.001	8210.656	33069.01

Fig. 3.1 Results from STATA of time taken and divisor

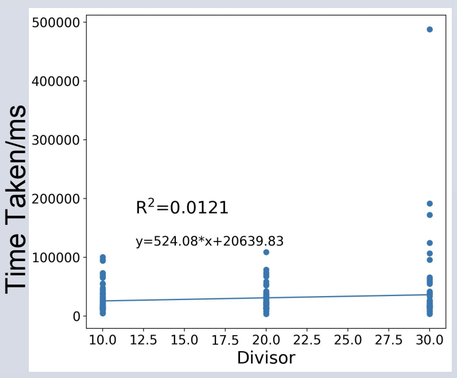

Fig. 3.2 Scattered diagram of data and the regression line drawn with gradient, y-intercept and R-squared value

For linear regression, the aim is to match the data points into a linear function, while divisor as x-axis (independent variable) and time taken as y-axis (dependent variable). So it can be written as: $Y = \alpha + \beta X$. After regression, we obtained $\alpha = 20639.83$ and $\beta = 524.0828$. $\beta > 0$ and as the gradient of regression line, suggests the divisor is positive correlated to the time, as shown in Fig. 3.2.

A test of linear hypothesis can be used to verify the significance about β. [H_0: $\beta = 0$]
The p-value (P>|t|) associated with the t-test is 0.083, which indicates weak evidence against H_0, suggesting that the coefficient may not be statistically significant at the conventional significance level, but the coefficient is still acceptable at 10% significance level). An R^2 of 0.0121 suggests the goodness of fit of regression line is relatively low, but it could be accepted since the data is insufficient to further prove the feasibility of the model and time recorded is strongly influenced by any other factors.

To sum up, the model is feasible in small amount of data collected but needs to be further verified after improving the method of data collection and collecting more valid data. Therefore, we can say that divisor is positive correlated with the time and linear spacing determined by divisor is negative correlated with time taken to reveal the image.

Future Works

- Improve the method of data collection, since the data are greatly influenced by factors like brightness and size of screen. The time taken is very hard to control due to online questionnaire. In the future, these limitations could be minimised by doing research in real life.
- Use more divisor values to collect more data due to current data is insufficient to draw the conclusion.
- To verify if there is any other factors such as colours of tiles affects time taken in perception of autostereograms.

Acknowledgement

- I would like to express my sincere gratitude to my friends and interviewees for their invaluable contributions in providing the data for analysis.
- I am especially grateful to my friend, Mr. Zijian Shao, for his assistance in analyzing the collected data.
- I would also like to extend my heartfelt appreciation to my Computer Science teacher, Mr. Yu Tang, for enhancing my understanding of computer stereo vision.

References

[1] Christopher Tyler (2014) Autostereogram. Scholarpedia, 9(4):9229.

[2] Georgeson, M., A. , Wallis, & S., A. . (2014). Binocular fusion, suppression and diplopia for blurred edges. Ophthalmic & physiological optics: the journal of the British College of Ophthalmic Opticians (Optometrists).

[3] How does disparity change with distance? | Socratic. (n.d.). Socratic.org. https://socratic.org/questions/how-does-disparity-change-with-distance

[4] Venkitachalam, M. (2015). Python Playground: Geeky projects for the curious programmer. http://cds.cern.ch/record/2113797