Single-image random-dot stereogram by cross-talk

Billy T.W. YU and William W.H. YU*

University of Hong Kong, Pokfulam Road, Hong Kong
Industrial & Manufacturing Systems Engineering Department

ABSTRACT

A fast, simple and memory saving algorithm for stereogram generation is presented in this paper. It is a ray-tracing like algorithm making use of the cross-talk effect in stereoscopic computer graphics for generating single-image random-dot stereogram. It actively looks for the *smallest equivalent class* of points with the same color so that it gives the greatest freedom of coloring for artistic design with stereogram.

Keywords: autostereogram, cross-talk, ray-tracing, stereogram

2. INTRODUCTION

Stereograms help to demonstrate the amazing power and abilities of the human mind and visual system. From a stereogram, a picture which at the first glance appears to be a mere display of random dots, most viewers have no trouble perceiving a three-dimensional scene of objects with a true sense of depth. This perception is accomplished by simple movement of the eye balls known as the oculomotor skill. The early Stereograms are computer-generated random-dot image pairs,[1,2] shown in Figure 1. When viewed monocularly (one eye at a time), these are but random-dot patterns which appear to have no meanings. When viewed with both eyes with proper image fusion and stabilized-image conditions, three-dimensional structures will merge.

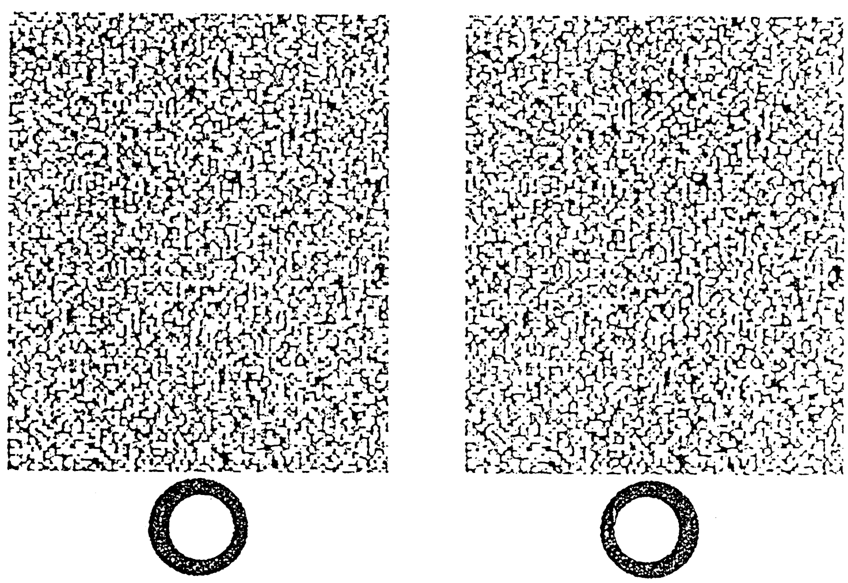

Figure 1 Julesz's Random dot stereogram

* William W.H. YU: E-mail: wwhyu@hkucc.hku.hk, and Billy T.W. YU: E-mail: twyu@hkusub.hku.hk; WWW: http://hkusub.hku.hk:8000/~twyu/

Studies showed that these pairs of images can be combined into one image producing similar effects as the random-dot pairs.[3,4] The single-image random-dot stereograms (SIRDS) are also commonly known as autostereograms. Theoretically, it is possible to produce a large tiling surface giving the viewer a feeling of continuos variation of depth.[5] However, the stereograms produced only surfaces of discrete depths or staircase-like surfaces. These stereograms are generated based on the psychological findings of Julesz using 'pixel shifting' algorithms. These algorithms work on the principles and repetitive properties in random-dot patterns.

Thimbleby first solved the discrete depth tiling problems using a stereo-vision geometry approach.[6] His algorithm generates random-dot stereogram (RDSG) based on 3D modeling of objects and is a breakthrough from the pixel shifting algorithms used by the earlier work. To determine the color of a new dot in the stereogram, the algorithm first scans the generated line of dots starting from its nearest neighbor to confirm if it is visible. If it is a visible dot, its color is defined by the color of a corresponding dot already existed, which is equivalent to setting up the *repetition widths* of different depth levels, *pixelwise*. When no corresponding dots are found, a free choice of color is assigned. In order to avoid repetitive computation procedures, Thimbleby used a double array to hold the depth data corresponding to every pixel. Though the Thimbleby algorithm is based on 3D object models, the procedure shares little common grounds with the popular 3D graphics procedures of the computer graphics industries.

Ray-tracing algorithms are frequently used to generate realistic 3D graphics. Considering the principle of stereograms, each dot, like 3D graphics, represents a ray reflected from the object and crossing the image plane to the viewer's eye, or view point. The difference is that in 3D graphics, the same point is intended to be presented, giving the same impression, to both eyes of the viewer. For each point of the 3D object, the stereogram requires two points or rays, one for each eye, to give the viewer a correct impression of depth. This paper presents a ray-tracing like algorithm for generating stereograms.

3. ALGORITHM

General principles. To apply the ray-tracing principles for generating stereogram is simple. The ray may originate from either the right eye or the left eye. The random dots are mapped from left to right if the ray starts from the left eye and vice versa. When the ray starts from the left eye and intersects the back plane or the surface of an object at point A, it returns to the right eye as a reflected ray. The rays intersect the image plane at points a and b, as illustrated in Figure 2. The following rules apply to the generation of dots and color assignment on the image plane.

New dots. A set of rays (see Figure 2) generates dots a and b on the image plane. Dots a and b represent the same object point A seen by the left and right eyes, respectively, and are assigned the same color of a free choice. When the two dots merge properly in the brain of the viewer, the depth of point A is perceived.

Recurring dots. Dots a and b are visible to both eyes. This means that dot b is the intersection between the return ray from point A to the right eye and the image plane, as well as the outgoing ray from the left eye to another object point or back plane, point B, as illustrated in Figure 3. The return ray from point B to the right eye intersects the image plane at c. Thus dot b presents object points A to the right eye and B to the left eye at the same time. Because dot b has already been assigned a color, dot c must be assigned the color equal to dot b.

Hidden points. If the reflected ray intersects the surface of another object at C as illustrated in Figure 4, point A is hidden from the right eye and only dot a is generated and assigned a color. Whenever this happens, the depth of the object point cannot be discerned because of the loss of stereo disparity information. However, the 3D structure of the stereogram as a whole will not suffer loss.

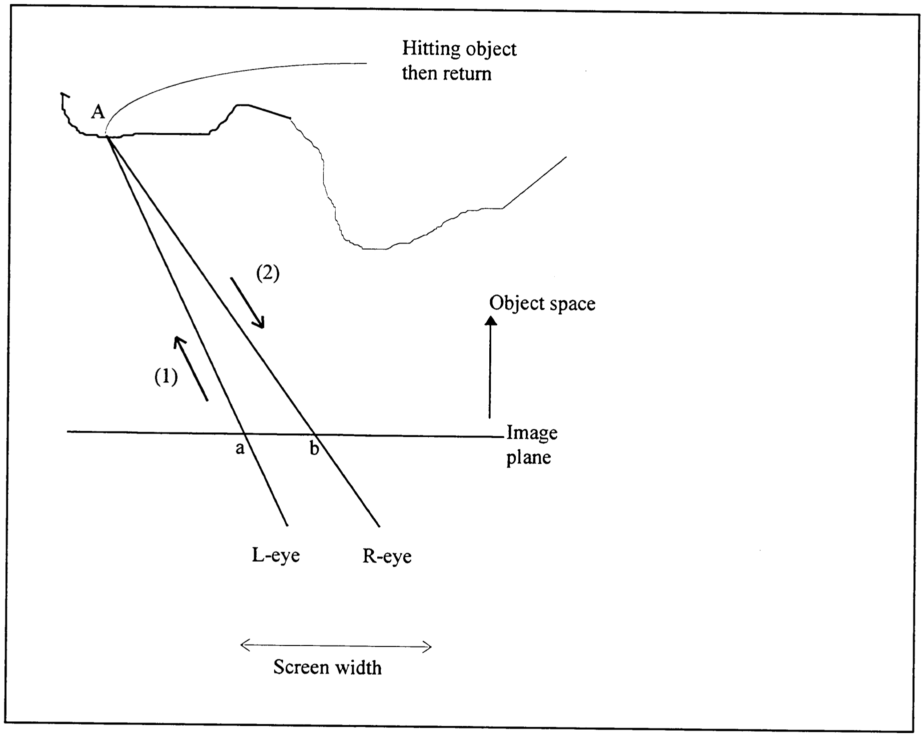

Figure 2 Color assignment by ray-tracking and it is starting from (1) to (2) yielding 2 points

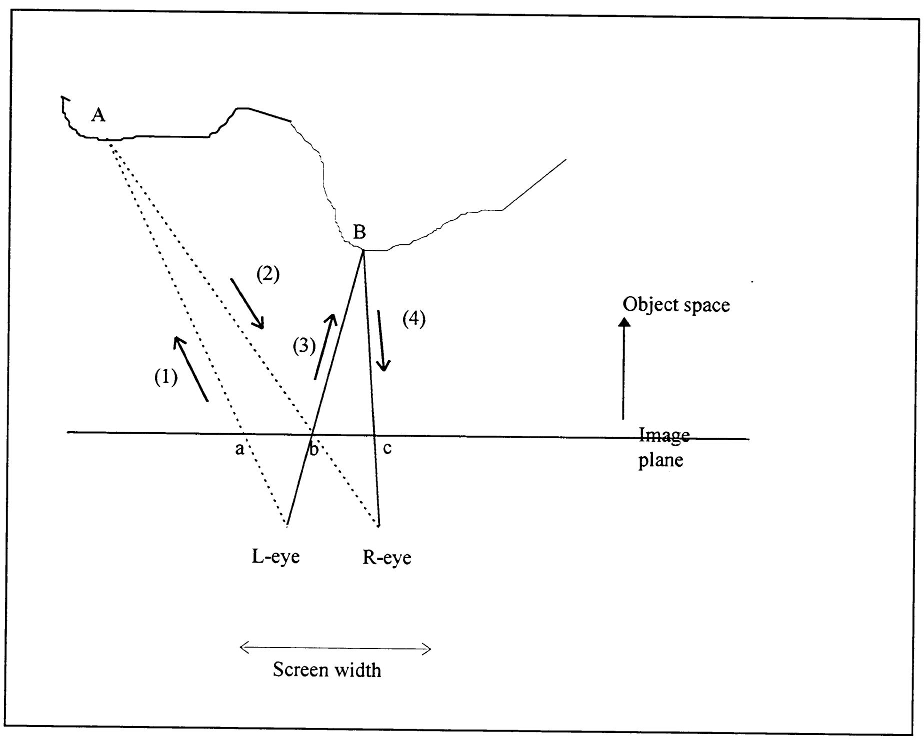

Figure 3 Track path staring from (1) to (4) yielding 3 points from 2 object points (A and B). Path (1) and (2) are dotted as they were previously located in Figure 2. When viewed stereoscopically the 3 points merge in our brain, 4 point will be seen but only the 2 middle points are steady and they will give rise to the depth perception as A and B as desired.

Overlapping pixels. Overlapping pixel problem occurs in the situation illustrated in Figure 5. The reflected ray from point A coincides or nearly coincides with the reflected ray from point C. Dot b has already been assigned a color equal to dot a due to the reflected ray from point A at an earlier mapping operation. The second set of rays starts from the left eye intersecting the image plane at c and the object surface at C. Dot c being a new dot is assigned a color of a free choice. The color of dot c may be different from the color of dot a. This creates a problem of the choice of color to be assigned to dot b, since both the reflected rays from point A and C intersect the image plane at dot b. The color of choice in this case is determined by the nearer object point to the image plane. Thus the color of dot b is overwritten with the color equal to dot c.

Finally, the pseudo code of the proposed algorithm is given in Figure 6.

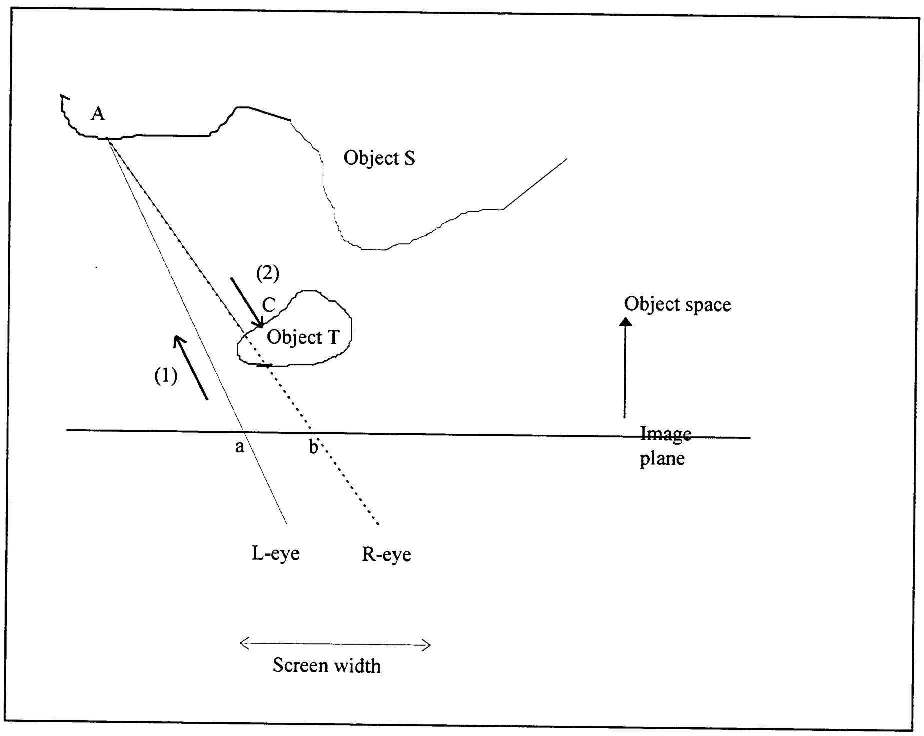

Figure 4 An obstacle, Object T, hides Point A from the right eye. The color of b remained undefined.

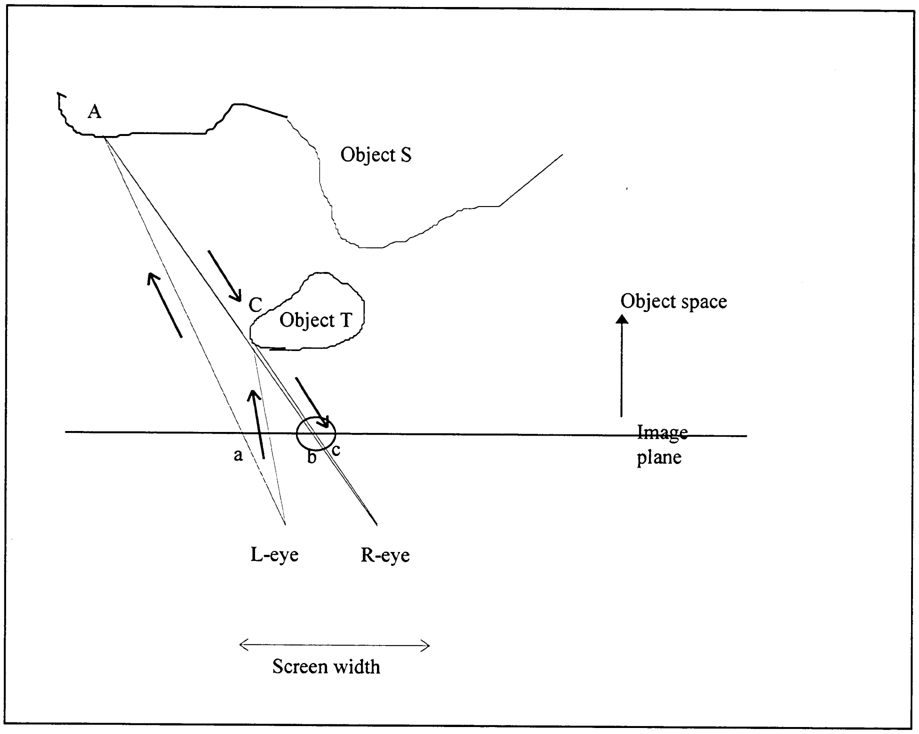

Figure 5 Overlapped pixel problem during color assignment process, two return rays hit the same pixel after round off near b.

```
BOOL shot_ray_from_L-eye_to(Point x) {
        shot ray from L-eye through x to object space to find intersection P
        return ( shot_ray_to_R-eye_from( P ) )
}

BOOL shot_ray_to_R-eye_from(Point x) {
        shot ray from P to R-eye
        if ( nothing hit in object space) find intersection I with screen plane
                if( I < MaximumX) display it with assigned color
        else return FALSE
        shot_ray_from_L-eye_to( I )
        return TRUE
}

Draw_stereogram( ) {
        For every scan-line
                For(x=0;x<MaximumX; x++) {
                        if ( already colored ) next x
                        arbitrary assign color for x
                        if( ! shot_ray_from_L-eye_to( x ) ) next x
                }
}
```

Figure 6 The pseudo code of the proposed algorithm

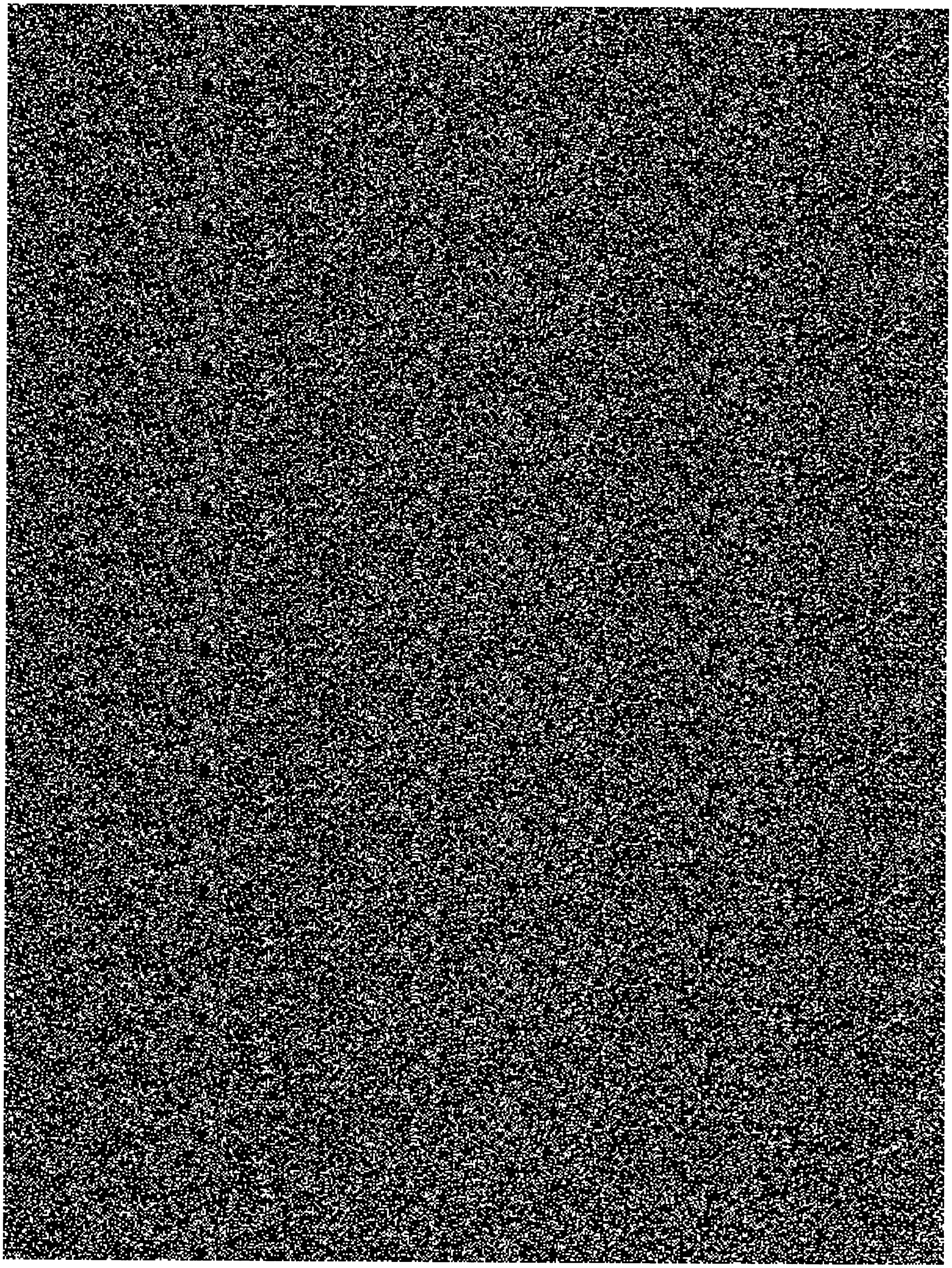

Figure 7 A stereogram of 6 spheres generated by the proposed algorithm

4. EVALUATIONS AND DISCUSSIONS

The algorithm is symmetric in effects. The same result is obtained whether dots are generated from left to right or vice versa.

The algorithm uses the ray-tracing approach. The ray-object surface intersection computations are the major overhead costs of the algorithm. The Thimberbly's algorithm uses the separation distances of two corresponding points to search and determine the depth of an object point. The speed of the proposed algorithm has advantages over the Thimberbly's algorithm up to 6 spheres. The performance comparison between the two algorithms are tabulated and diagrammed in Table 1 and Chat 1, respectively. Figure 7 shows a stereogram of 6 spheres generated by the proposed algorithm.

Ball Number	Time / sec	
	Thimbleby's	This paper's
1	15.87	12.09
2	16.20	13.02
3	16.53	13.78
4	16.97	14.45
5	16.75	15.71
6	17.03	16.97
7	17.85	17.85
8	18.40	18.73
9	18.62	19.77
10	19.01	20.55

Table 1 The first column is the number of ball; the second is the time for generating a stereogram with Thimbleby's method and the third is that for this paper proposed algorithm

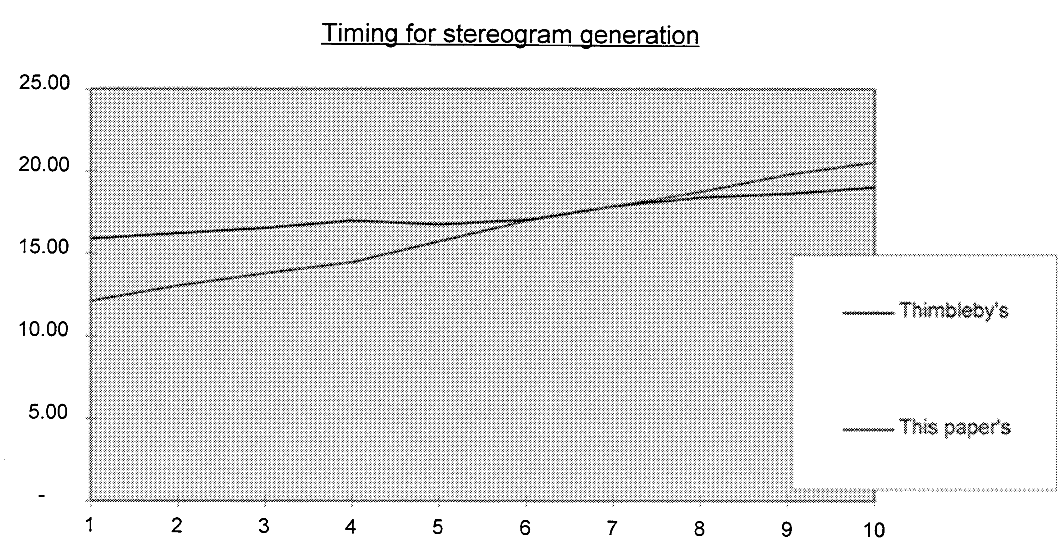

Chart 1 Actual timing for both stereogram generation algorithm

The proposed algorithm has definite advantage over the earlier stereogram generation algorithms in terms of simplicity and being compatible with familiar computer graphics techniques such as ray-tracing. This means that the acceleration techniques developed for ray-tracing can be adapted to further accelerate the proposed algorithm. It is believed, for example, that by applying the Glassner space subdivision method the algorithm will be accelerated by at least 80%.[7]

The algorithm produces every point in the stereogram representing itself only. This plainly implies its reflexivity. $p = p$ and $\Re(p) = \Re(p)$, where p is a point location, and $\Re$ is an abstract function returning a point matching the input, and they should have the same color. $\Re$ here works from left to right.

Geometrically speaking, the algorithm works properly both from left-to-right and vice versa, although the algorithm here implemented in the forward direction. Employing the left point to locate the right point or in reversal produces the same result. $\Re(p_L) = p_R$ and $\Re^{-1}(p_R) = p_L$ and $\Re = (\Re^{-1})^{-1}$. Thus, the algorithm is symmetric.

The recursive way of actively searching points of same color generates two successive points of equal color, $p_2 = \Re(p_1)$, $p_3 = \Re(p_2)$ and $p_3 = \Re(\ \Re(p_1)\ )$. This showed the transitive attribute of the algorithm. With all these properties, the equivalent quality of the algorithm is established.

This proposed algorithm introduces a recursive stereogram generation technique. Each recursive process breeds one *smallest equivalent class* as addressed by Thimbleby.[8] This is the class of least restriction and thus allowing the greatest freedom of coloring for artistic design.

5. FURTHER WORKS

At this stage, stereograms remain chiefly as random dot patterns. Further investigation can be undertaken to produce stereograms with more realistic visual effects. It will be worthwhile to investigate the problem associated with animated stereogram. Different acceleration techniques can be examined to improve the performance of the algorithm.

ACKNOWLEDGEMENTS

This work was supported by the Research Grant Committee and the Industrial & Manufacturing Systems Engineering Department of the University of Hong Kong, RGC Code: 345/063/0174

REFERENCES

1. Marr, D. and T. Poggio, "A computational theory of Human Stereo Vision" Proc. Royal Soc. of London, Vol. B204, Royal Society, London, 1979, pp. 304-28
2. Julesz, B., "Binocular depth perception of computer generated pattern", Bell Syst. Tech. J. 39, 1125-62.
3. Julesz, B., Foundations of Cyclopean Perception, University of Chicago Press, Chicago, 1971.
4. Tyler, C.W. and Chang, J.J., "Visual echoes: The perception of repetition in quasi-random patterns" Vision Res. 17, 109-16
5. Tyler, C.W. and M.B. Clarle, "The Autostereogram," SPIE Stereoscopic Displays and Applications, Vol.1258, 1990, pp.182-96
6. Thimbleby, H. "Displaying 3D image: algorithms for single-image random-dot stereogram" Computer Oct 94' pp.38-48
7. A.S. Glassner, "Space Subdivision for Fast Ray Tracing," CG&A, Oct. 1984, pp. 15-22.
8. Thimbleby, H. "An equivalent class algorithm for drawing autostereogram", Software-Practice and experience March 96 V.26(3), pp. 309-25
9. Bishop, P.O., "Can RDSG serve as a model for perception of depth in relation to real 3D object", Vision Res. V.36 I10, pp. 1473-7