

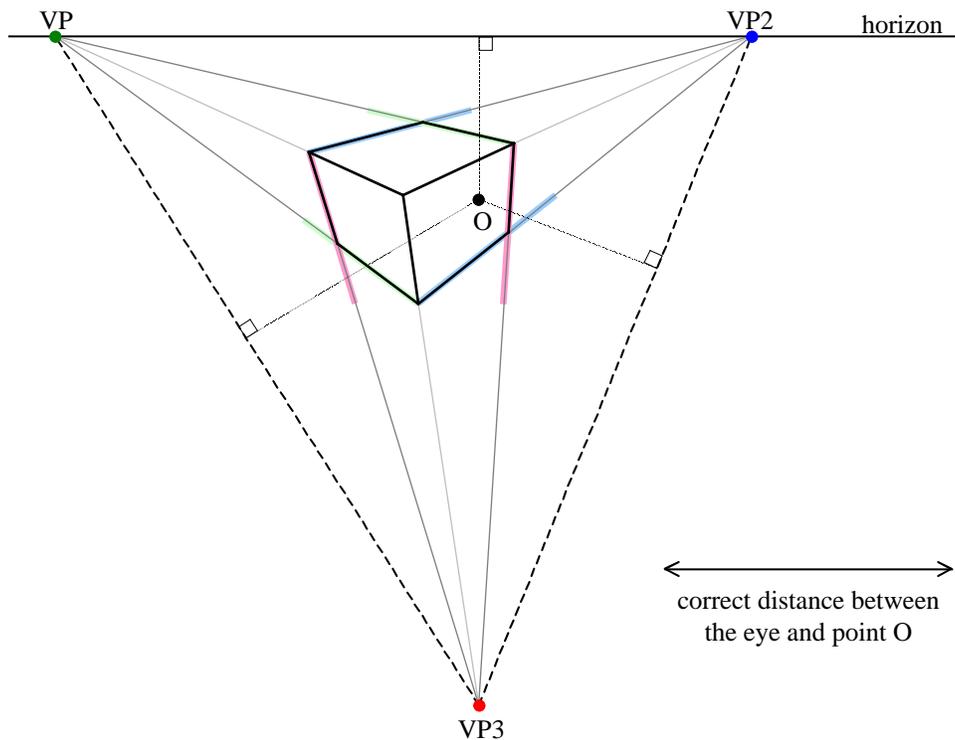
# Determination of the correct eye position for viewing perspective images of 3D scenes

Justin Jang  
[jang@cc.gatech.edu](mailto:jang@cc.gatech.edu)

Jarek Rossignac  
[jarek@gvu.gatech.edu](mailto:jarek@gvu.gatech.edu)

College of Computing and GVVU Center  
Georgia Institute of Technology

July 6, 2001



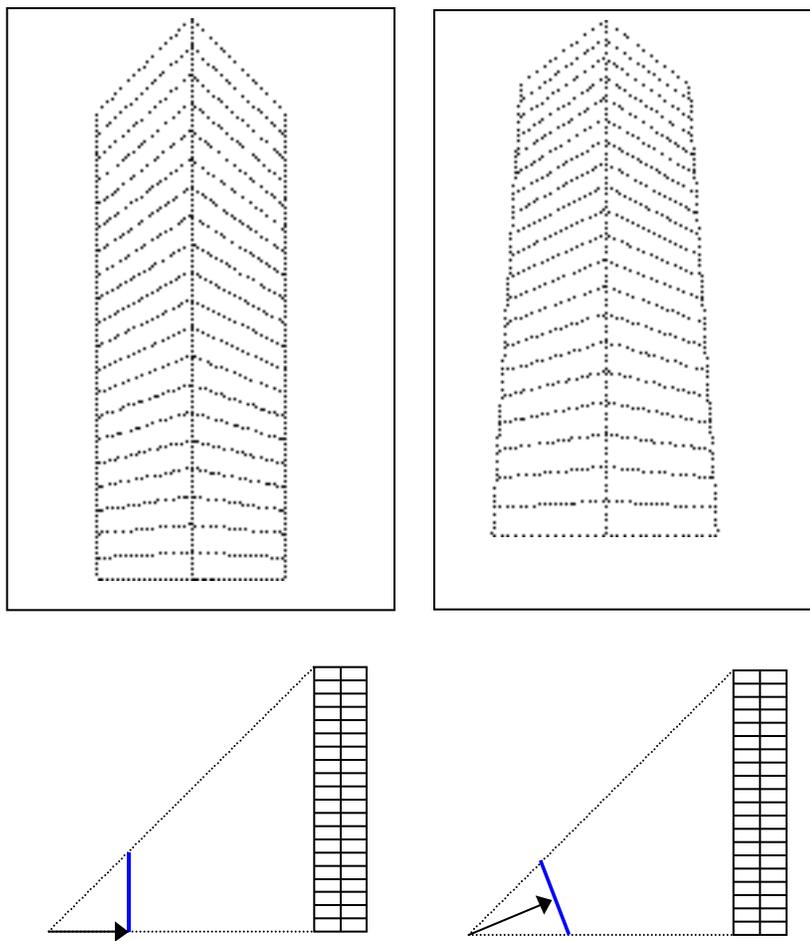
## Abstract

*Realistic viewing of perspective images requires placing the viewer at the correct location. In this paper, we present a technique for determining the correct viewing location for perspective images of 3D scenes. The technique is based on the determination of three vanishing points in mutually orthogonal directions. We also use this technique to devise a simple method for manually drawing images of rectangular prisms in correct perspective with respect to a viewpoint.*

# 1. Introduction

A perspective image is created by projecting objects in 3D space onto a 2D image plane with respect to a viewpoint. This point, also known as the center of projection, is the proper location for viewing the image. A perspective image viewed from any other point may cause the scene to appear distorted.

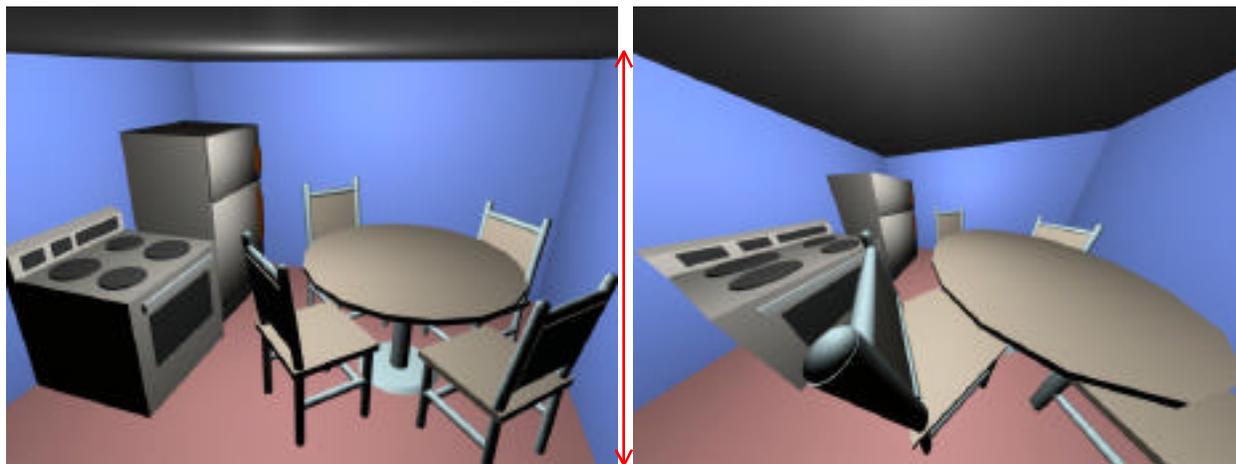
Figure 1 shows two views of the same skyscraper. When viewed from a viewpoint close to the center of the image, the skyscraper on the top left appears distorted. It is actually the top half of a projection centered at the bottom or the skyscraper. As illustrated below (bottom-left), the camera is pointed parallel to the ground. Thus, a correct perspective is obtained by positioning the viewer close to the bottom of the image. The same part of the skyscraper is also shown on the top-right. In this case, however, the camera is looking up at a  $22.5^\circ$  angle from the ground as depicted below (bottom-right). The center of the top-right figure is the correct projection of the viewpoint, so it provides a more realistic depiction when viewed at the center.



**Fig 1:** The image on the top-left appears distorted unless the eye is placed close to the bottom center of it, simulating a human looking horizontally towards the bottom part of a sky-scraper. The image on the right appears correct when the viewpoint is close to the center of the image. It simulates a person looking up towards the center of a skyscraper.

The distance of the viewpoint from the image also affects the distortion of a 3D scene. Figure 2 shows the same scene with different viewpoint distances from the image plane. The red lines to the right of Figures 2a and 2b indicate the correct viewing distance from the center of each image, respectively. A

large distance results in a small field-of-view and a small distance results in a large field-of-view. Figure 2b has a wider field-of-view than Figure 2a, thus it appears considerably more distorted than Figure 2a, when viewed from a distance greater than the correct viewing distance for Figure 2a.



**Fig 2:** Both images should be viewed by placing the eye in front of their center at a distance respectively indicated by the vertical double-headed arrows to their right. Clearly, although correct, the image on the right will appear distorted from any comfortable viewing condition.

In this paper, we address the problem of finding the center of projection of a picture or image given only the image itself. Of course, this is only possible if the appropriate clues exist in the image.

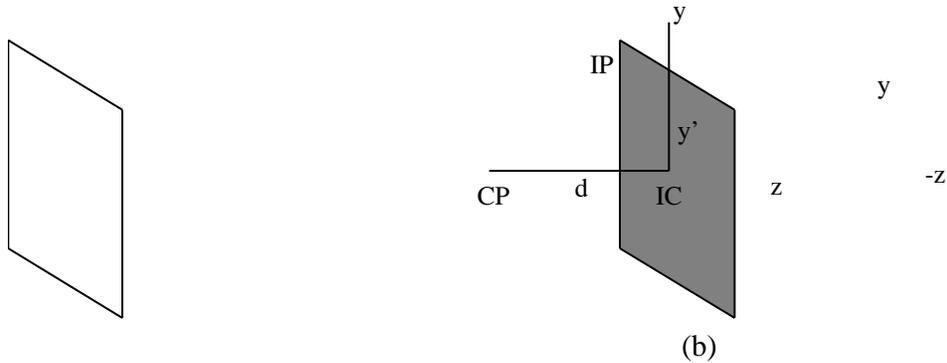
Our solution is to identify three pairs of edges such that the two edges of each pair are, in the three-dimensional space, parallel to each other and orthogonal to edges in other pairs. We expect the user to manually identify these pairs by selecting points on the image. (Automatically picking correlated lines and features falls beyond the scope of this paper.) We have developed a graphical interface that simplifies this manual step and have found that it takes less than a minute per image. Of course, this approach is limited to images containing three mutually orthogonal pairs of edges and thus is most appropriate for images of streets, buildings, or interiors. Furthermore, each set of lines cannot be parallel on the image plane because they will not converge at a vanishing point.

Once the three pairs of edges are identified, our program computes the corresponding vanishing points, from which it derives the center of projection, defined by its projection  $O$  on the image plane and by its distance  $d$  to  $O$ .

The remainder of this paper has the following structure. Section 2 defines the terminology and notations used in this paper. Section 3 lists relevant prior work. Section 4 describes our method of finding the viewpoint, which contrasts with prior art. Section 5 shows the results of applying our method to real images. Section 6 presents a technique for manually drawing rectangular prisms in 3-point perspective. Section 7 summarizes the paper and provides suggestions on future work.

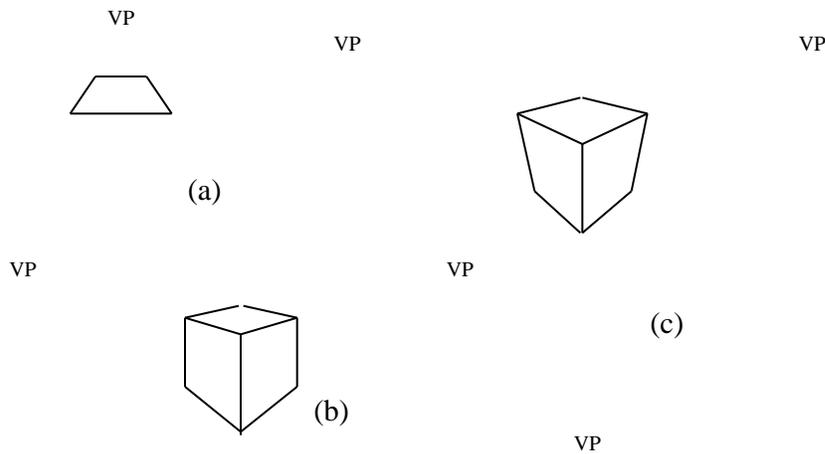
## 2. Background

We first define and explain our terminology and notation. **Image plane**, view plane, and projection plane are used synonymously and will be denoted by the capital letter  $P$ . The **viewpoint** is located at the center of projection, sometimes called the projection center. This is the correct viewing position and will be denoted by the capital letter  $E$ . The **image center** (IC), also sometimes called the center of view or view center, is the orthogonal projection of the center of projection onto the view plane. The image center will be denoted by the capital letter  $O$ , and the lowercase letter  $d$  will refer to the **distance between  $E$  and  $O$**  (see Figure 3).



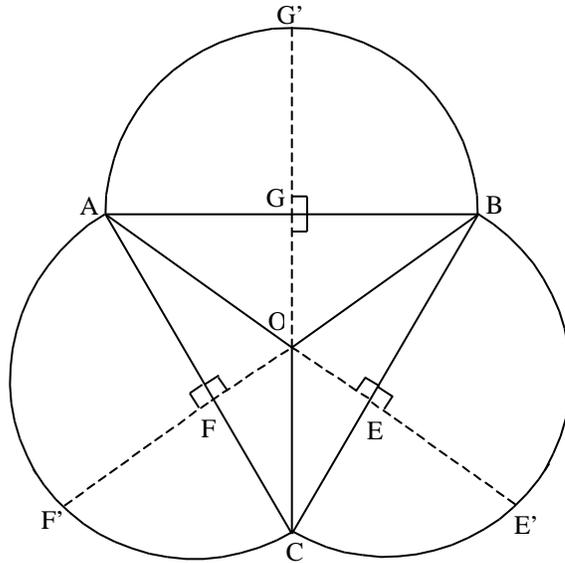
**Fig 3:** The eye should be positioned at the center of projection (a). As shown right (b), its orthogonal projection (CP) onto the screen is called the image center (IC) and is at distance  $d$  from it.

A **vanishing point** (VP) is the point at which a set of parallel lines appears to converge, when placed under perspective [Hea97]. Sets of lines that are parallel to the view plane have parallel projections and thus have vanishing points at infinity. In 3D, a maximum of three sets of lines can be mutually orthogonal to each other and up to two of these sets can be parallel to the view plane. As a result, there are cases where 1, 2, or 3 vanishing points exist with respect to a given set of three mutually orthogonal directions, as for example the vertical and two horizontal directions defined by the orientation of building facades on a street. These are referred to as one-point, two-point, and three-point perspective, respectively (see Figure 4) [Mau85, Gil75]. Note that these terms apply to a given set of three mutually orthogonal directions. Thus, some parallelepipeds in an image may appear under a one-point perspective, while others appear under three-point perspective.



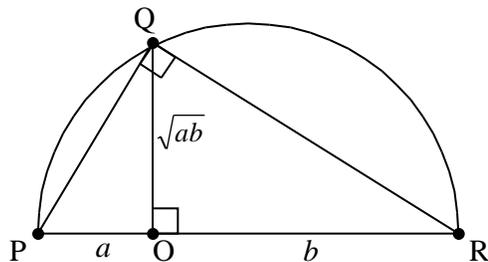
**Fig 4:** One-point (a), two-point (b), and three-point (c) perspectives for a given parallelogram.

In the case of three-point perspective, the image center is located at the **orthocenter** of triangle T, where the vertices of T are the three vanishing points [Tru98, Mau85]. Therefore, we find it convenient to label both the orthocenter and image center with the capital letter **O**. (The orthocenter of a triangle is the intersection of its three altitudes. An **altitude** of a triangle is a line that runs through a vertex and is perpendicular to that vertex's opposite edge. See Figure 5.) Mauldin [Mau85] calls triangle T the **vanishing point triangle**.



**Fig 5:** The orthocenter, O, of a triangle (A,B,C) is the intersection of their attitudes, which each pass through a vertex and are orthogonal to the opposite edge of the triangle. *Semi-circles are shown for each edge.*

Mauldin [Mau85] gives a geometric method for determining the view center and viewpoint from this vanishing point triangle. This method employs the geometric mean. The **geometric mean** of two distances  $a$  and  $b$  is defined as  $\sqrt{ab}$ . A semicircle construction can be used to graphically derive a line segment of length  $\sqrt{ab}$ , given segments of length  $a$  and  $b$  (see Figure 6). We will refer to this as the **geometric mean construction**.



**Fig 6:** Graphical construction of the geometric mean using a semi-circle.

Finally, a **regular parallelepiped** or **rectangular prism** is a rectangular box that has three mutually orthogonal sets of parallel lines delimiting its shape [Mau85]. For simplicity, we will refer to this as a “**box**”.

### 3. Prior Art

Freese [Fre30] teaches a simple technique for drawing in 2-point perspective. Mauldin [Mau85] and Gill [Gil75] give similar methods for drawing in 1-point, 2-point, and 3-point perspective. These methods are based on two pre-drawn views: the plan view (top view) and elevation view (side view) [Fol92, Fre30].

The area of computer graphics has matrix-based approaches for projecting points onto a view plane with respect to a viewpoint [Fol92]. At its core, a method for perspective projection has the following equation:

$$(x', y') = \frac{dx}{d+z}, \frac{dy}{d+z}$$

The coordinate  $(x', y')$  is the projection of point  $(x, y, z)$  onto the view plane  $z = 0$ , centered at the origin (see Figure 3b).

There is prior knowledge on how to determine the viewing characteristics from features of the image itself. Trucco [Tru98] states that the image center is located the orthocenter of  $T$ , where  $T$  is the triangle on the image plane defined by the three vanishing points of three mutually orthogonal sets of parallel lines in space. Antone and Teller et. al. [Ant00, Tel01] gave methods for automatically deducing the likely view center of an image.

Mauldin [Mau85] gives mathematical and geometric methods for determining the view center and viewpoint from the vanishing point triangle. Figure 5 illustrates a geometric construction for viewpoint determination described by Mauldin. Segments  $HH'$ ,  $FF'$ , and  $GG'$ , which are derived from geometric means, fold up to meet at the viewpoint, i.e.  $E$  is at distances  $|HH'|$ ,  $|FF'|$ , and  $|GG'|$  from  $H$ ,  $F$ , and  $G$  respectively. Furthermore, the geometric mean construction can be used to find the distance  $d$  of the viewpoint directly out from  $O$  by using any one of the pairs  $OA/OH$ ,  $OB/OF$ , and  $OC/OG$  as  $a/b$ .

We propose and prove a slightly different construction for computing  $O$  and  $d$ .

## 4. Determining the Correct Viewpoint

Given the perspective projection of a box, the “proper” position of the eye (the center of projection) can be determined with the following procedure:

1. Compute the three vanishing points  $A$ ,  $B$ , and  $C$  on the view plane.
2. Compute the orthocenter  $O$  of triangle  $ABC$ . Compute  $d = \sqrt{-OB \cdot OA}$ .
3. The eye should be located at distance  $d$  from  $O$  along the normal to the screen.

In general, any three pairs of parallel lines will work, as long as each pair is orthogonal to the other two. An important constraint is that none of the lines can be parallel to the view plane. This constraint ensures that there are three vanishing points.

### 4.1. Computing the Vanishing Points

Given a 2D image, a vanishing point can be obtained by computing the intersection of lines that are identified by the user as parallel in 3D space (before projection). One way to specify two lines is with four Cartesian coordinates. Let  $((x_a, y_a), (x_b, y_b))$  and  $((x_c, y_c), (x_d, y_d))$  define line 1 and line 2, respectively, on the  $x$ - $y$  plane. It is simple to think of these two lines in slope-intercept form. Compute the slopes of the two lines  $m_1 = (y_b - y_a) / (x_b - x_a)$  and  $m_2 = (y_d - y_c) / (x_d - x_c)$ . If  $m_1$  and  $m_2$  are equal then the two lines are parallel. Otherwise, compute the intercepts  $b_1 = y_a - m_1 x_a$  and  $b_2 = y_c - m_2 x_c$ . The intersection is at  $x = (b_2 - b_1) / (m_1 - m_2)$  and  $y = m_1 x + b_1$ .

In 3D, the vanishing point can be found by computing the location of the point at infinity in the direction parallel to the line, and placing this point under perspective projection. Let  $A$ ,  $B$ , and  $C$  be the coordinates of the vanishing points with respect to the center of projection  $O$ , where  $A = (x_A, y_A)$ ,  $B = (x_B, y_B)$ ,  $C = (x_C, y_C)$ . Let  $L_1$ ,  $L_2$ , and  $L_3$  be vectors parallel to each of the three sets of parallel lines, where  $L_1 = (x_1, y_1, z_1)$ ,  $L_2 = (x_2, y_2, z_2)$ , and  $L_3 = (x_3, y_3, z_3)$ .  $L_1$ ,  $L_2$ , and  $L_3$  correspond with  $A$ ,  $B$ , and  $C$ , respectively, and are orthogonal to each other.

The perspective projection of the point at infinity in direction  $L_1$

$$\text{is: } A = (x_A, y_A) = \frac{dkx_1}{d+kz_1}, \frac{dky_1}{d+kz_1} = \frac{dx_1}{\frac{d}{k}+z_1}, \frac{dy_1}{\frac{d}{k}+z_1} = \frac{dx_1}{z_1}, \frac{dy_1}{z_1}$$

Likewise, vanishing points B and C are computed from direction vectors  $L_2$  and  $L_3$  giving:

$$B = (x_B, y_B) = \frac{dx_2}{z_2}, \frac{dy_2}{z_2}$$

$$C = (x_C, y_C) = \frac{dx_3}{z_3}, \frac{dy_3}{z_3}$$

#### 4.2. Computing the Orthocenter

From the three vanishing points A, B, and C, we get the vanishing point triangle ABC. Let H, F, and G be the points on the edges opposite A, B, and C, respectively, such that AH, BF, and CG are perpendicular to BC, CA, and AB, respectively. Note that AH, BF, and CG are altitudes of triangle ABC (see Figure 5). The following are equations for H, F, and G.

$$H = B + \frac{BA \cdot BC}{\|BC\|^2} BC, F = A + \frac{AB \cdot AC}{\|AC\|^2} AC, G = A + \frac{AC \cdot AB}{\|AB\|^2} AB$$

And using properties of right triangles, we obtain the following expression.

$$O = A + \frac{\|AB\| \|AG\|}{\|AH\|^2} AH$$

Note that there are other symmetric equations for O given A, B, C, H, F, and G.

#### 4.3. Why O lies at the orthocenter of triangle ABC

Let OA, OB, and OC be the vectors from the image center O to A, B, and C, respectively. For each of these three vectors, project the other two onto it. If each pair of projections are equal, then O is the orthocenter (intersection of the altitudes) of triangle ABC (see Figure 5).

$$OB \cdot OA = x_B x_A + y_B y_A = \frac{dx_2}{z_2} \frac{dx_1}{z_1} + \frac{dy_2}{z_2} \frac{dy_1}{z_1} = d^2 \frac{x_2 x_1 + y_2 y_1}{z_2 z_1} = * -d^2$$

(\*Since  $L_1$  and  $L_2$  are orthogonal,

$$L_2 \cdot L_1 = 0 \quad x_2 x_1 + y_2 y_1 + z_2 z_1 = 0 \quad x_2 x_1 + y_2 y_1 = -z_2 z_1 \quad (x_2 x_1 + y_2 y_1) / (z_2 z_1) = -1)$$

Likewise,  $OC \cdot OA = OA \cdot OB = OC \cdot OB = OA \cdot OC = OB \cdot OC = -d^2$ .

#### 4.4. Computing distance d of the viewpoint from the view plane

From the computation above, we get the expression  $OB \cdot OA = -d^2$ , which gives us  $d = \sqrt{-OB \cdot OA}$ . Any two of the vectors OA, OB, and OC can be used to compute d.

Geometrically,  $OA \cdot OC$  projects one vector onto the other, so  $OA \cdot OC = \|OG\| \|OC\|$  and  $d = \sqrt{\|OG\| \|OC\|}$ , as described by Mauldin (see Figure 5). This allows us to use the geometric mean construction (Figure 6) to determine d.

## 5. Results

We wrote an application that allows users to manually place six lines (representing three pairs of mutually orthogonal lines) on an image. Given the user-defined edges, the application computes the image center and distance  $d$  of the viewpoint from the image center.

We tested this approach on a number of images of paintings and photographs, including IPIX captured images [Ipi01], as well as on computer-rendered images. The image center of a photograph, including IPIX images, should be at the center of the image, unless the image was cropped or an asymmetric lens was used. While this should also apply to most computer rendered images, artistic paintings and renderings don't necessarily share this property.

When testing our application on the photographs, the average percent error on the computed image centers (assuming the IC is at the center) was 12%. The average image size tested is 675 (width) by 472 (height). Figure 7 shows the results of running our application on a photograph.



**Fig 7:** *The purple lines are the user-selected lines, the maroon lines are the altitudes of the vanishing point triangle, and the white box encircles the calculated image center (where the altitudes intersect).*

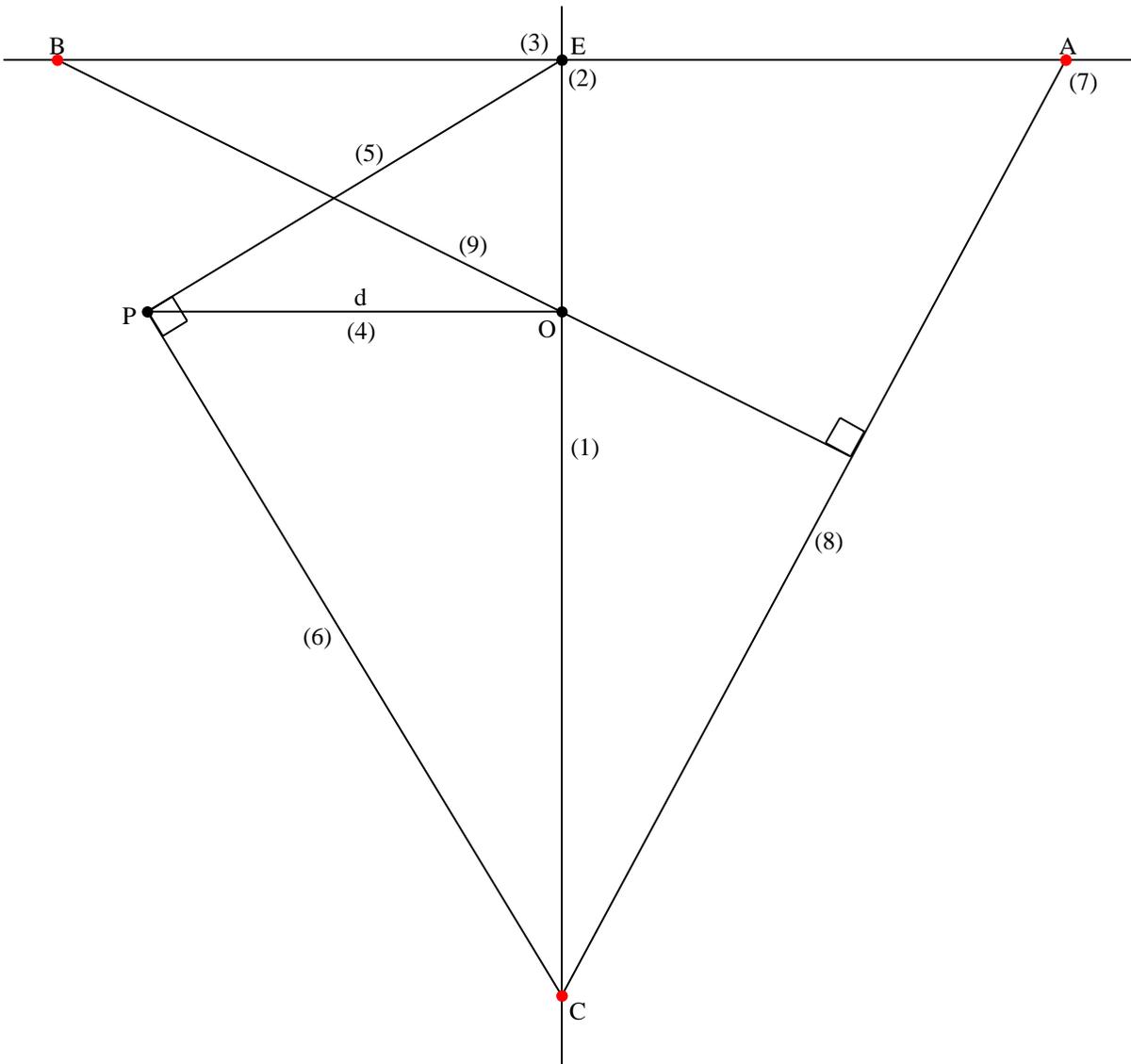
We also tested our approach on IPIX images [Ipi01] of primarily indoor scenes. For the set of images we tested, the average percent error of the computed image centers from the center of the images was approximately 6.4%. The average size of the IPIX images in this test set is 330 by 229. The distance  $d$  with respect to the image size lets us approximate the field of view of the camera  $fov=2\arctan(size/2d)$ , where  $size$  is the horizontal or vertical size of the image. The average calculated horizontal field-of-view was approximately a  $90^\circ$ . Consequently, to view these IPIX images correctly, one must place the eye at a distance approximately equal to half the width of the image (see Figure 8). This is clearly an uncomfortable position for the viewer, even if the IPIX image was blown out to full screen size. For the set of images that we tested, the actual range of horizontal field-of-view values was  $79.7$  to  $103.5^\circ$ . It makes sense that the range of the vertical field-of-view values,  $55.1^\circ$  to  $89.6^\circ$ , is lower than the horizontal values because the tested set of images are larger horizontally than vertically.



When testing images with pairs of lines that were nearly parallel, a slight human error sometimes lead to extremely abnormal results including image centers very far off the image and negative  $d$  values. This is due to the volatility of the line intersection computations. When the slope of two lines approaches each other, the equation of their intersection approaches a singularity. In general, images with 3 highly oblique sets of lines with respect to the image plane yield better results, since their respective projections on the image plane will have larger differences in slope.

## 6. Drawing a Box in Perspective

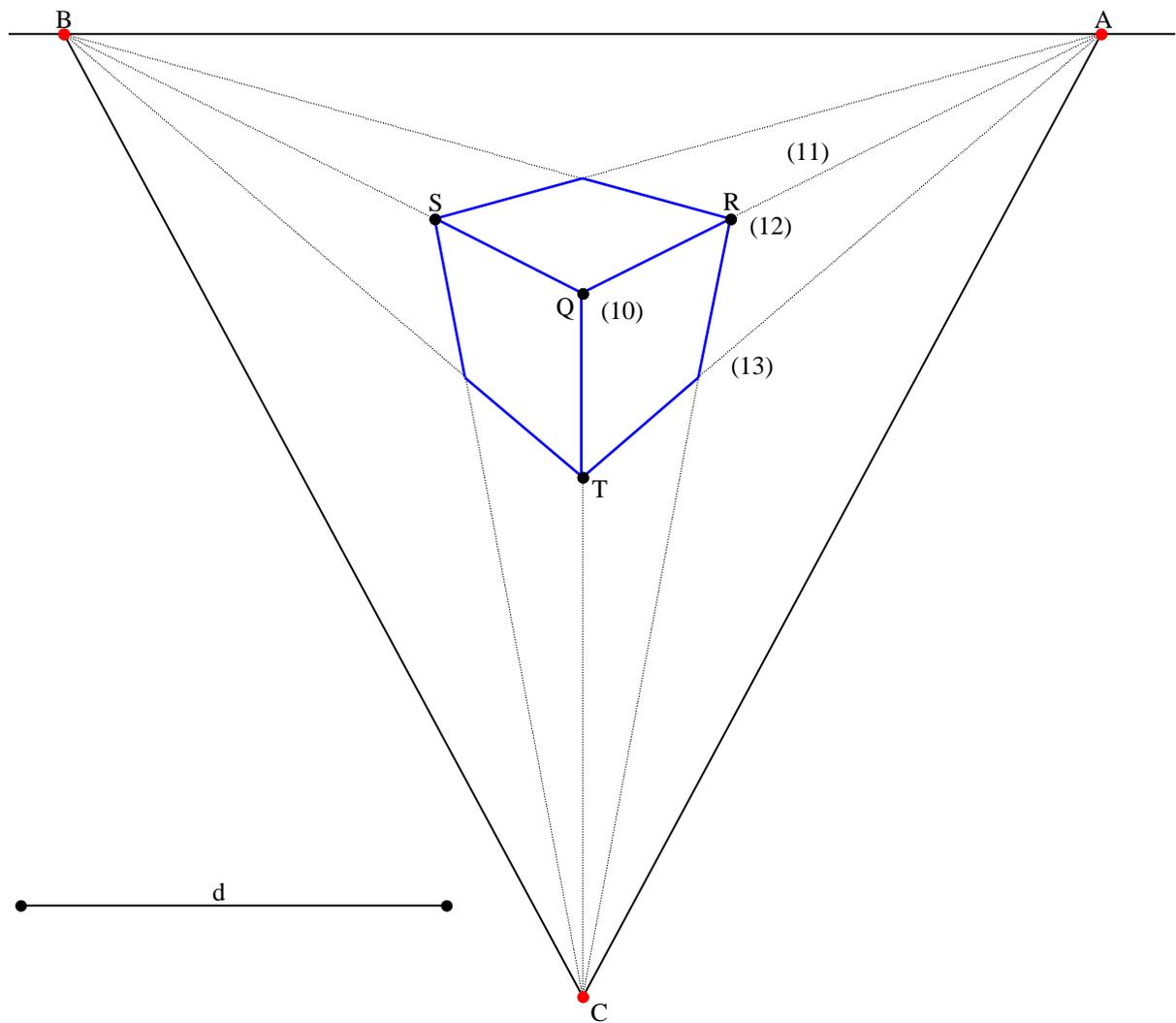
We now describe a method for drawing a box in 3-point perspective. A desired center of view ( $O$ ) and distance  $d$  are assumed to be given (see Figures 10, 11, and 12).



**Fig 10:** Constructing the correct vanishing points,  $A$ ,  $B$ , and  $C$ , for a given  $O$  and  $d$ .

1. Draw a vertical line through  $O$ .
2. Pick a point  $H$  on the vertical line depending on the desired elevation.
3. Draw a horizontal line through  $H$ . This is the horizon line.
4. Draw a horizontal segment of length  $d$  with one endpoint on  $O$ . Label the other endpoint  $M$ .

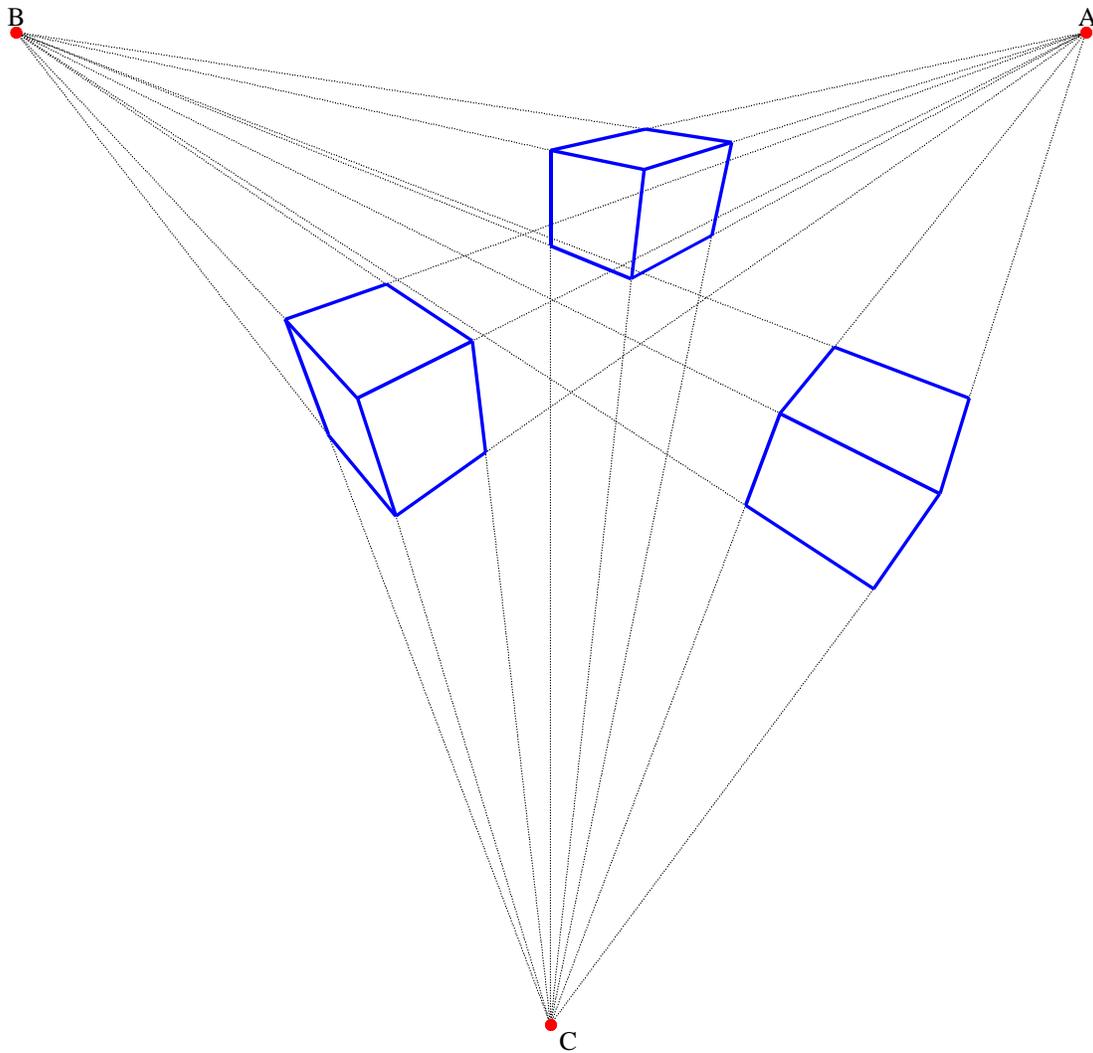
5. Draw segment HM.
6. Draw a line perpendicular to HM that extends from M to intersect the vertical line. Label this intersection as point C.
7. Pick a point A on the horizontal line depending on the desired left-right rotation of the parallelepiped.
8. Draw line AC.
9. Draw a line perpendicular to AC that extends through O and intersects with the horizontal line. Label this intersection as point B.
10. Pick any point to represent the closest corner of the box. Label this as point Q.
11. Draw lines QA, QB, and QC.
12. Pick points R, S, and T on QA, QB, and QC, respectively. These points determine the dimensions of the box.
13. Draw segments RB, RC, SA, SC, TA, and TB.
14. If the hidden edges are desired, draw segments from A, B, and C to the intersections of SC and TB; TA and RC; and RB and SA, respectively.



**Fig 11:** Drawing a parallelepiped, given 3 vanishing points.

Steps 1-9 define the vanishing point triangle as shown in Figure 10. The entire construct can be rotated to achieve a desired tilt about the normal to the view plane. Repeating steps 7-9 allows a different rotation about the y-axis while using the same elevation parameters.

Steps 10-14 draw a box (parallelepiped) as shown in Figure 11. Given a set of vanishing points, steps 10-14 can be repeated to draw boxes that are parallel to each other (see Figure 12).



**Fig 12:** Drawing several aligned parallelepipeds for the same vanishing points.

## 7. Conclusions and future work

Given suitable image features, the proper viewing location of an image can be found.

For small images like the iPIX captured images, the correct viewpoint is often very close to the screen. In particular, this distance is likely to be shorter than the typical viewing distance of the user to the screen. As a consequence, the images appear distorted when viewed under comfortable viewing conditions.

In the case of small images of paintings, it makes sense for them to have short viewing distances since a painting is typically much larger than an image of it on a computer screen. Rather than a few inches from the screen, the viewpoint would be a few feet from the painting, which is probably where the painter was standing

Larger images give more accurate results because the maximum level of accuracy attainable for user-selected lines on a computer image depends on the resolution of the images. Furthermore, it is easier to view a larger image, because of the bigger distance  $d$ , which could otherwise be uncomfortably close to a small image. While applying our technique to smaller images could cause a higher degree of error in the results, this is of little consequence. The human error in placing the eye at the desired location with respect to the image increases to the same degree.

There are several other factors that affect the accuracy of our technique. The presence of 3 highly oblique sets of lines with respect to the image plane as well as a large field of view each contribute to a larger divergence in the slopes of each line pair, thus contributing in a more accurate line intersection result. The presence of crisp and distinctive lines or image features as well as the overall size of the image affect the accuracy of the line selection process. Further work could include a detailed analysis of how each of these factors contributes to the accuracy of the technique.

Our technique has several limitations. The selection of lines requires significant human intervention, and thus is prone to human error. There must be prior knowledge about the subject of the image in order to conclude that a particular edge is actually orthogonal to others in the world space of the scene. While there has been some work in this area [Ant00], further study on extracting appropriate information from images needs to be done.

## References

- [Ant00] Matthew Antone and Seth Teller, *Automatic Recovery of Relative Camera Rotations for Urban Scenes*, Proceedings of CVPR, 2000, Volume II, June 2000, pp. 282-289.
- [Fol92] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA, 2<sup>nd</sup> edition, 1992.
- [Fre30] Ernest I. Freese. *Perspective Projection*. The Pencil Points Press, Inc., New York, 1930.
- [Gil75] Robert W. Gill. *Creative Perspective*. Thames and Hudson, London, 1975.
- [Hea97] Donald Hearn and Pauline M. Baker. *Computer Graphics, C Version*. Prentice Hall, Inc., Upper Saddle River, NJ, 1997, p. 446.
- [Ipi01] Internet Pictures Corporation website, <http://www.ipix.com>.
- [Mau85] John H. Mauldin. *Perspective Design*. Van Nostrand Reinhold Co., New York, 1985.
- [Tel01] S. Teller, M. Antone, M. Bosse, S. Coorg, M. Jethwa, N. Master, *Calibrated, Registered Images of an Extended Urban Area*, MIT Computer Graphics Group, 2001.
- [Tru98] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1998, p. 131-132.