FIGURES FOR THE BOOK

3D Computer Graphics
A Mathematical Introduction with OpenGL

Samuel R. Buss

October 6, 2005

This is an electronic file with figures for the book *3D Computer Graphics, A Mathematical Introduction with OpenGL*, by S. Buss, Cambridge University Press, 2003.

This file of figures is intended to for instructors' use in creating course lecture slides. These figures may be used freely and circulated electronically as long as you acknowledge their source.

Chapter I

Introduction

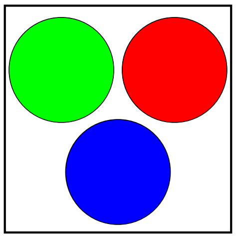
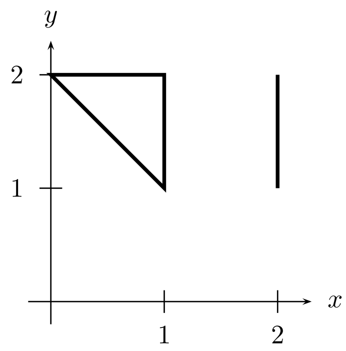


Figure I.1: A pixel is formed from subregions or subpixels, each of which displays one of three colors. See color plate C.1.




```
penup();
moveto(2,2);
pendown();
moveto(2,1);
penup();
moveto(1,2);
pendown();
moveto(0,2);
moveto(1,1);
moveto(1,2);
```

Figure I.2: Examples of vector graphics commands.

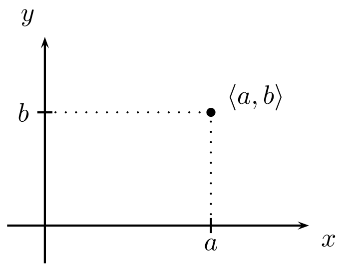


Figure I.3: The xy-plane, $\mathbb{R}^2$, and the point $\langle a, b\rangle$.

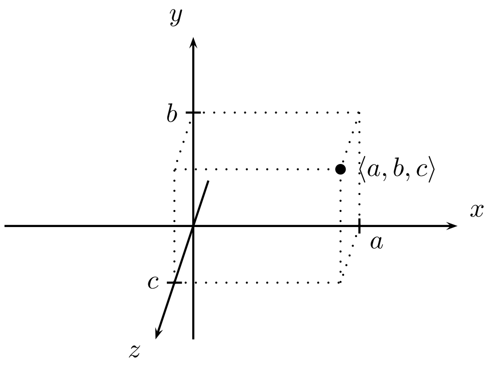


Figure I.4: The coordinate axes in $\mathbb{R}^3$, and the point $\langle a, b, c\rangle$. The z-axis is pointing towards the viewer.

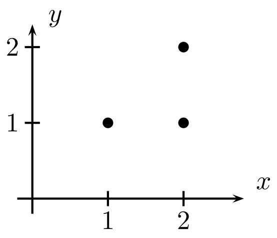


Figure I.5: Three points drawn in two dimensions.

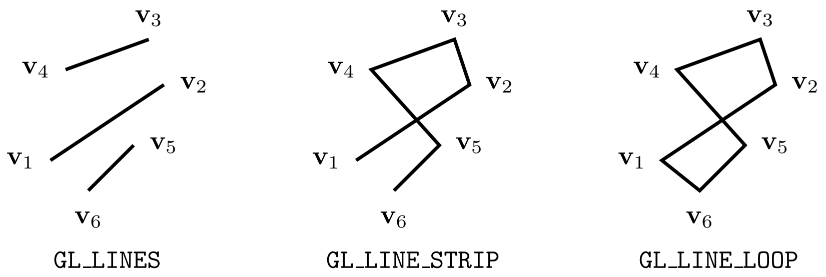


Figure I.6: The three line drawing modes, as controlled by the parameter to `glBegin`.

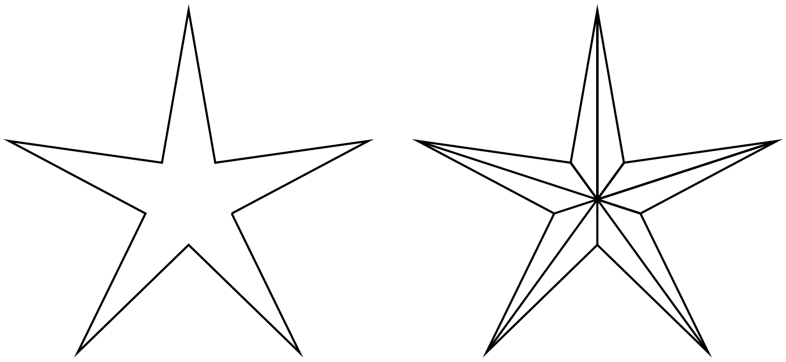


Figure I.7: Figures for exercises I.2, I.3, and I.4.

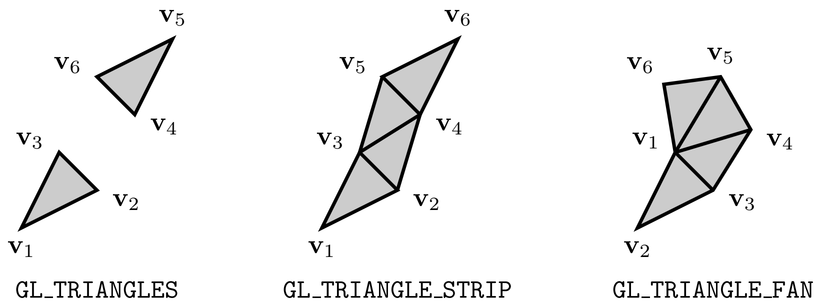


Figure I.8: The three triangle drawing modes. These are shown with the default front face upwards. In regard to this, note the difference in the placement of the vertices in each figure, especially of $\mathbf{v}_5$ and $\mathbf{v}_6$ in the first two figures.

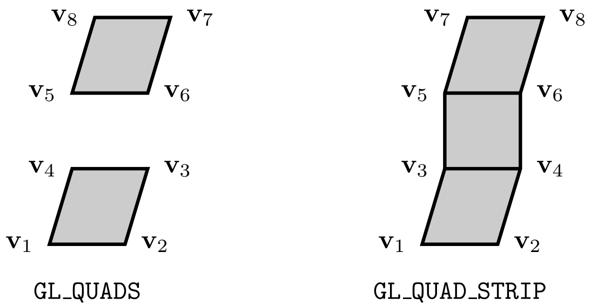


Figure I.9: The two quadrilateral drawing modes. It is important to note that the order of the vertices is different in the two modes!

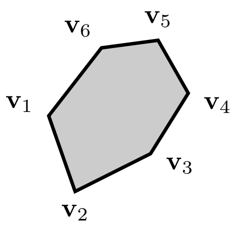
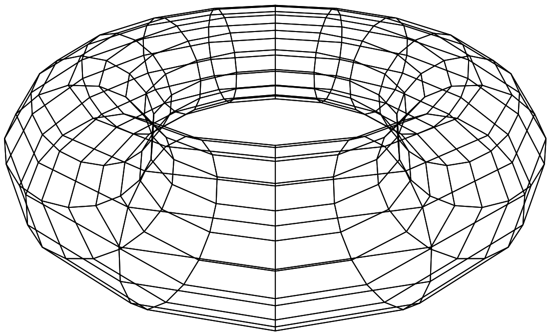
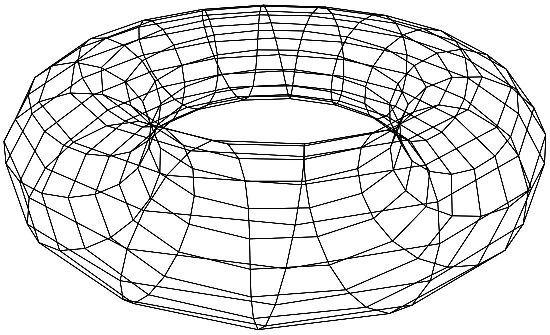


Figure I.10: A polygon with six vertices. The OpenGL standards do not specify how the polygon will be triangulated.



(a) Torus as multiple quad strips.



(b) Torus as a single quad strip.

Figure I.11: Two different methods of generating wireframe tori. The second torus is created with the supplied OpenGL program `WrapTorus`. In the second torus, the quadrilaterals are not quite planar.

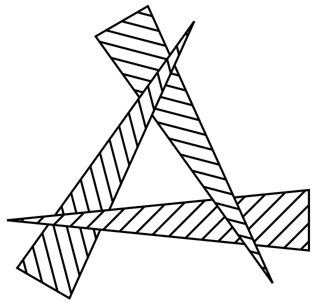
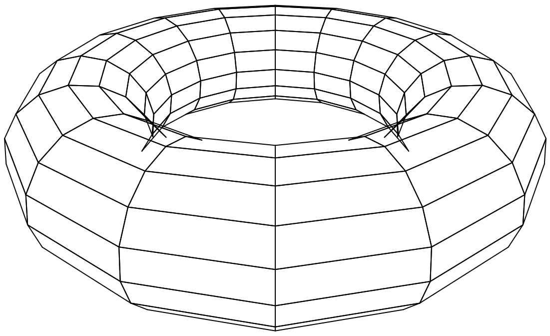
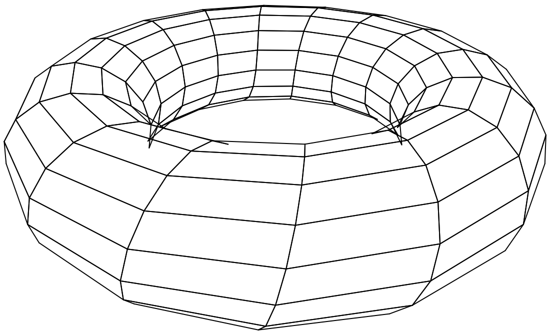


Figure I.12: Three triangles. The triangles are turned obliquely to the viewer so that the top portions of each triangle is in front of the base portion of another.



(a) Torus as multiple quad strips.



(b) Torus as a single quad strip.

Figure I.13: Two wireframe tori, with back faces culled. Compare to figure I.11.

Chapter II

Transformations and Viewing

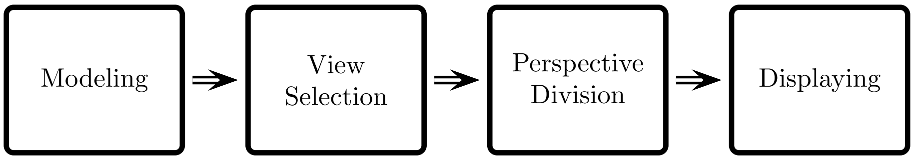


Figure II.1: The four stages of the rendering pipeline in OpenGL.

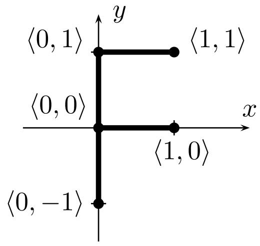


Figure II.2: An "F" shape.

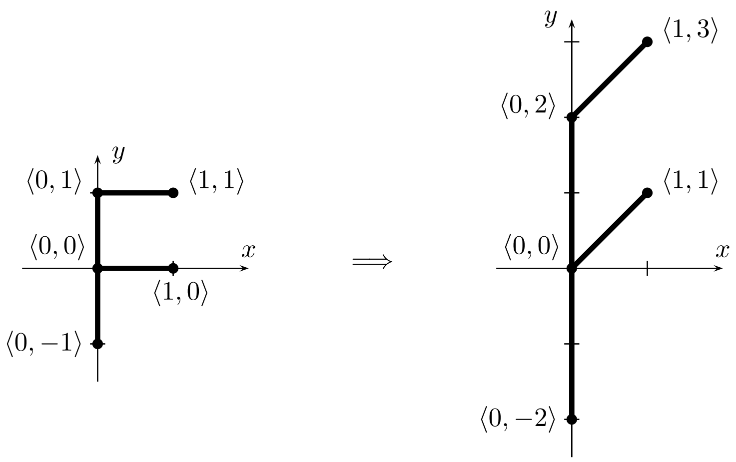


Figure II.3: An "F" shape transformed by a linear transformation.

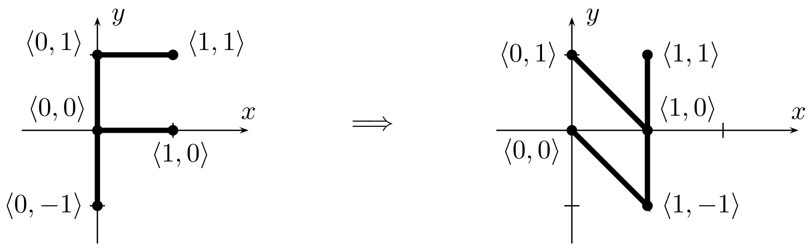


Figure II.4: An affine transformation acting on an "F."

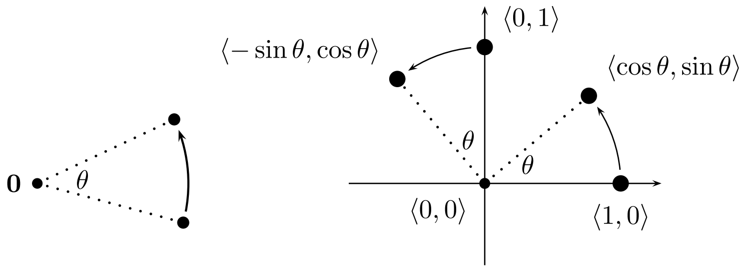


Figure II.5: Effect of a rotation through angle θ. The origin $\mathbf{0}$ is held fixed by the rotation.

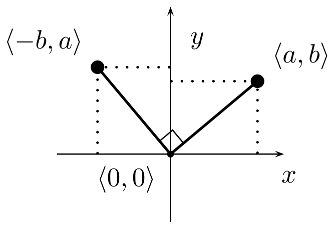


Figure II.6: A rigid, orientation preserving, linear transformation acting on the unit vectors $\mathbf{i}$ and $\mathbf{j}$.

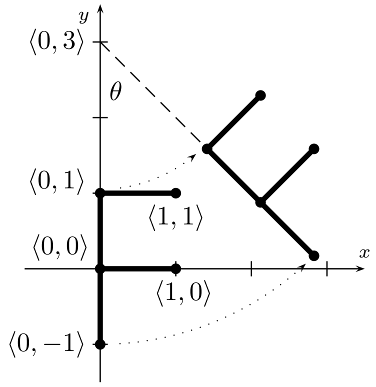


Figure II.7: A generalized rotation $R_\theta^{\mathbf{u}}$. The center of rotation is $\mathbf{u} = \langle 0, 3\rangle$. The angle is $\theta = 45^\circ$.

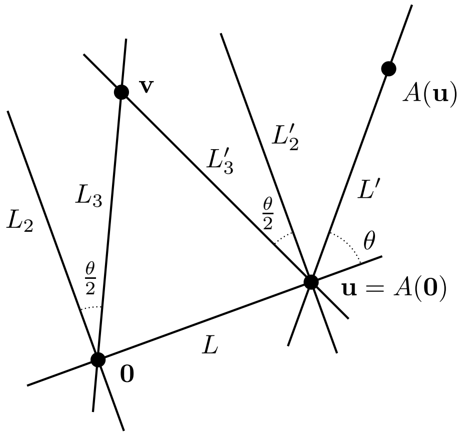


Figure II.8: Finding the center of rotation. The point $\mathbf{v}$ is fixed by the rotation.

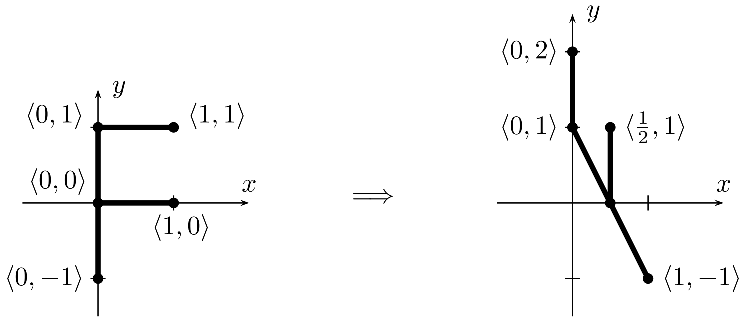


Figure II.9: An affine transformation acting on an "F."

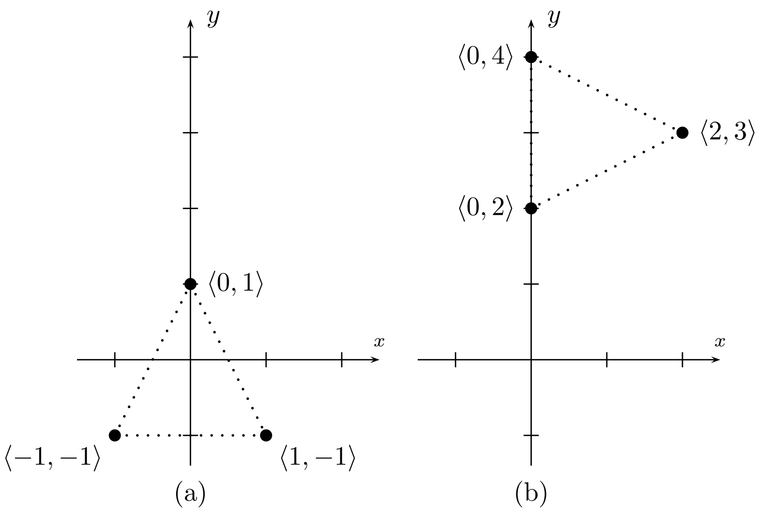


Figure II.10: Drawing points (a) without transformation by the model view matrix and (b) with transformation by the model view matrix. The matrix is as given in the text and represents a rotation of -90° degrees followed by a translation of $\langle 1, 3\rangle$.

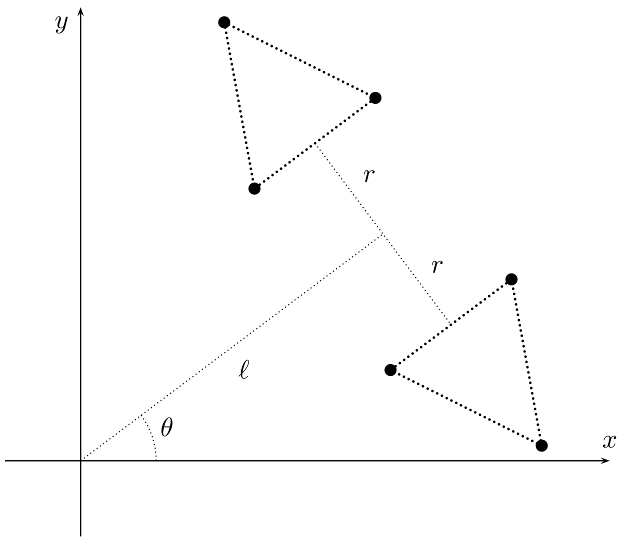


Figure II.11: The results of drawing the triangle with two different model view matrices. The dotted lines are not drawn by the OpenGL program and are present only to indicate the placement.

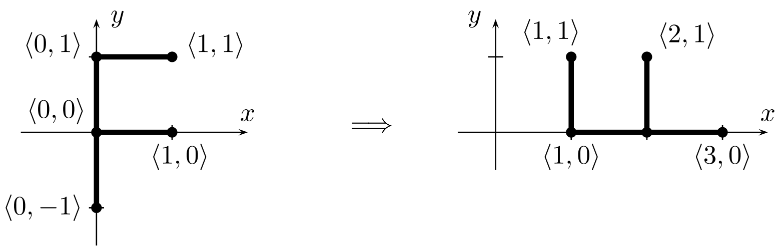


Figure II.12: The affine transformation for exercise II.11.

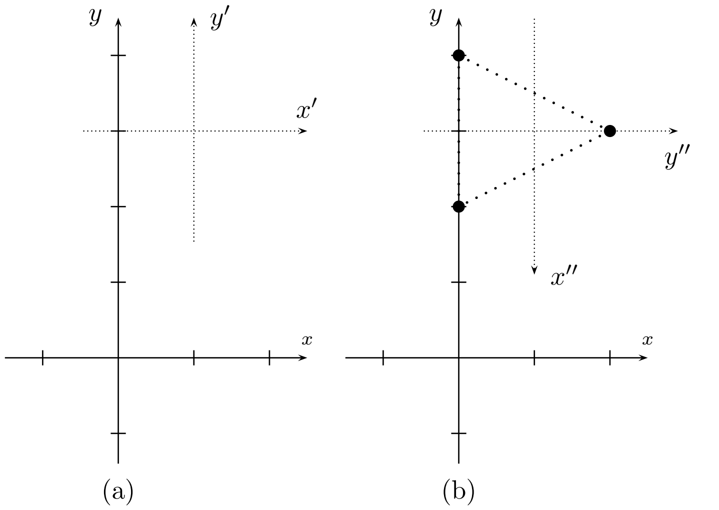


Figure II.13: (a) shows the local coordinate system $x'y'$ obtained by translating the xy-axes by $\langle 1, 3 \rangle$. (b) shows the coordinates further transformed by a clockwise rotation of 90 degrees, yielding the local coordinate system with axes x'' and y''. In (b), the triangle's vertices are drawn according to the local coordinate axes x'' and y''.

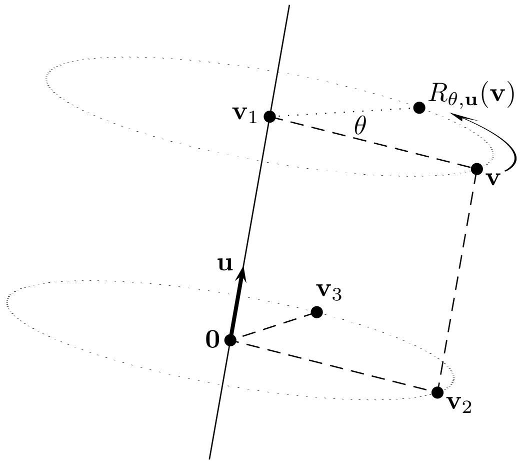


Figure II.14: The vector $\mathbf{v}$ being rotated around $\mathbf{u}$. The vector $\mathbf{v}_1$ is $\mathbf{v}$'s projection onto $\mathbf{u}$. The vector $\mathbf{v}_2$ is the component of $\mathbf{v}$ orthogonal to $\mathbf{u}$. The vector $\mathbf{v}_3$ is $\mathbf{v}_2$ rotated $90°$ around $\mathbf{u}$. The dashed line segments in the figure all meet at right angles.

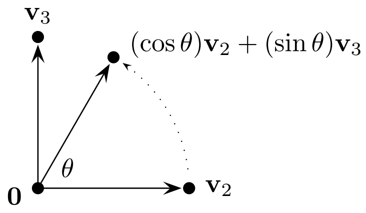


Figure II.15: The vector $\mathbf{v}_2$ being rotated around $\mathbf{u}$. This is the same situation as shown in figure II.14, but viewed looking directly down the vector $\mathbf{u}$.

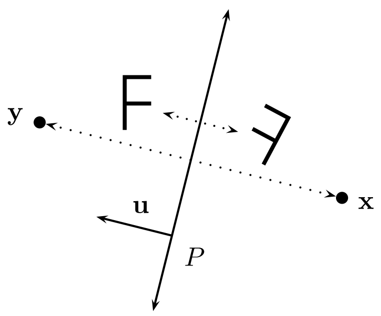


Figure II.16: Reflection across the plane P. The vector $\mathbf{u}$ is the unit vector perpendicular to the plane. A reflection maps a point to its mirror image across the plane. The point $\mathbf{x}$ is mapped to the point $\mathbf{y}$ directly across the plane, and vice-versa. Each "F" is mapped to the mirror image "F".

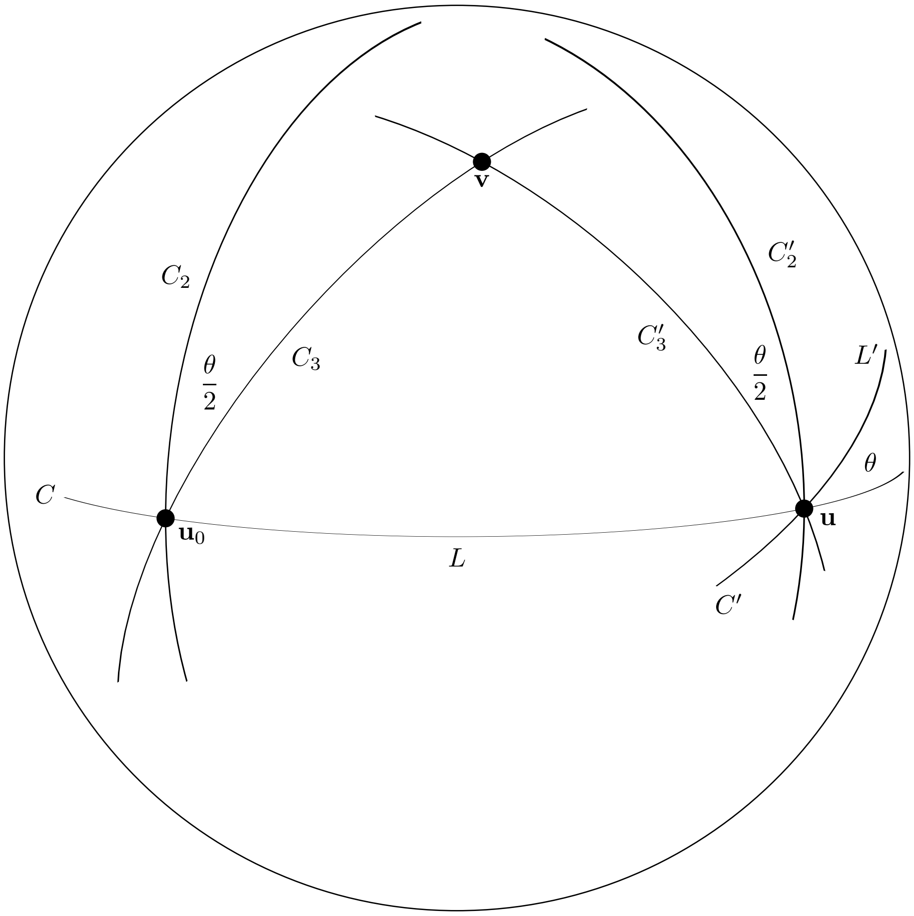


Figure II.17: Finding the axis of rotation. We have $\mathbf{u} = A(\mathbf{u}_0)$ and $\mathbf{v} = A(\mathbf{v})$. Compare this with figure II.8.

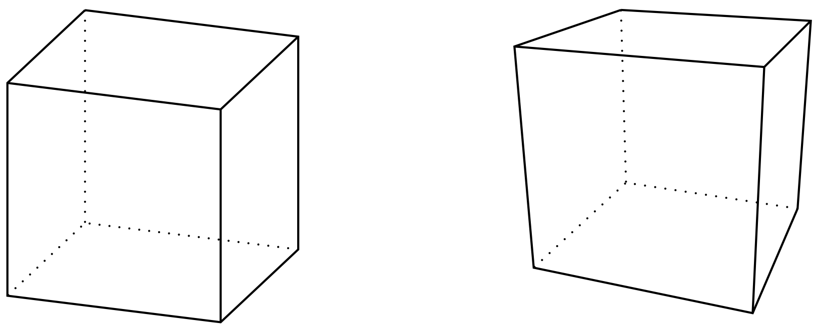


Figure II.18: The cube on the left is rendered with an orthographic projection. The one on the right with a perspective transformation. With the orthographic projection, the rendered size of a face of the cube is independent of its distance from the viewer; compare, for example, the front and back faces. Under a perspective transformation, the closer a face is, the larger it is rendered.

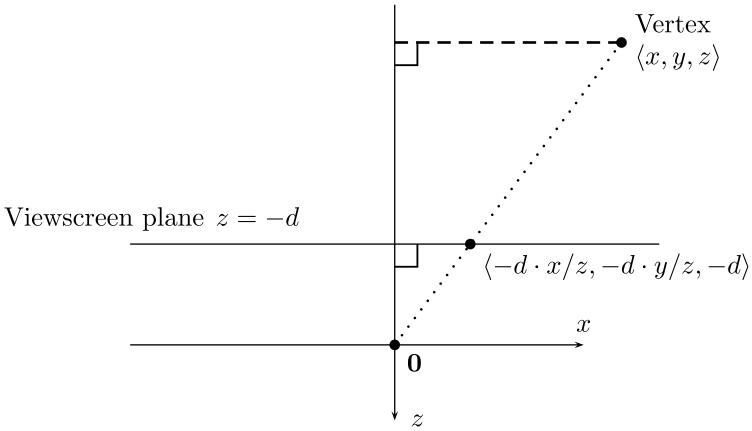


Figure II.19: Perspective projection onto a viewscreen at distance d. The viewer is at the origin, looking in the direction of the negative z axis. The point $\langle x, y, z\rangle$ is perspectively projected onto the plane $z = -d$, which is at distance d in front of the viewer at the origin.

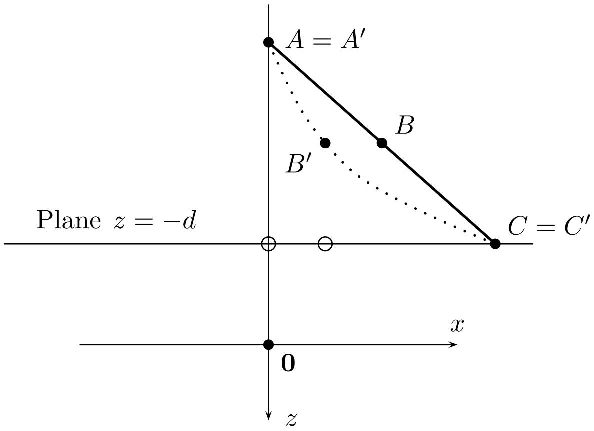


Figure II.20: The undesirable transformation of a line to a curve. The mapping used is $\langle x, y, z\rangle \mapsto \langle -d\cdot x/z, -d\cdot y/z, z\rangle$. The points A and C are fixed by the transformation and B is mapped to B'. The dotted curve is the image of the line segment AC. (The small unlabeled circles show the images of A and B under the mapping of figure II.19.)

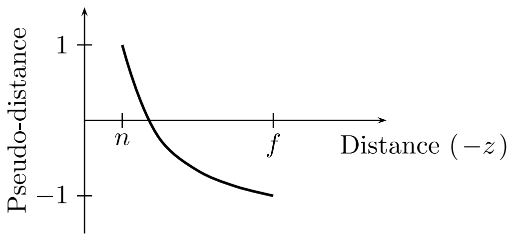


Figure II.21: Pseudo-distance varies nonlinearly with distance. Larger pseudo-distance values correspond to closer points.

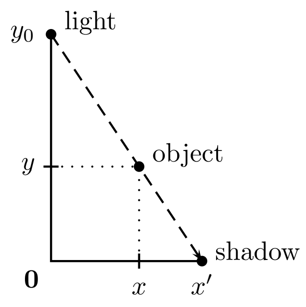


Figure II.22: A light is positioned at $\langle 0, y_0, 0\rangle$. An object is positioned at $\langle x, y, z\rangle$. The shadow of the point is projected to the point $\langle x', 0, z'\rangle$, where $x' = x/(1 - y/y_0)$ and $z' = z/(1 - y/y_0)$.

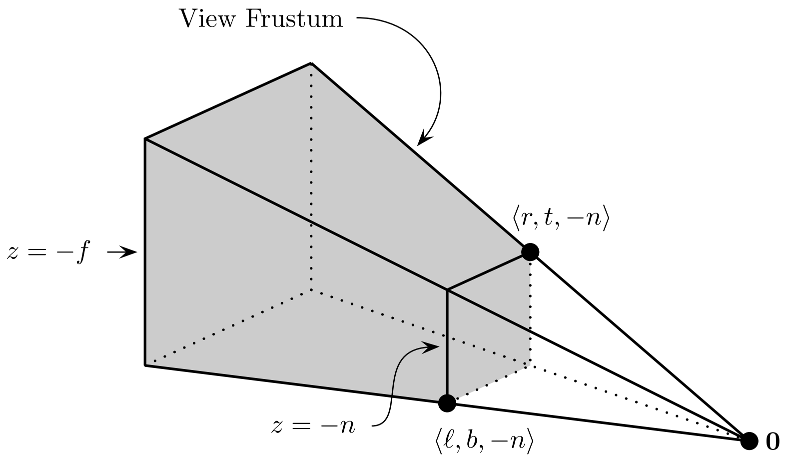


Figure II.23: The frustum viewed with `glFrustum(`ℓ`, `r`, `b`, `t`, `n`, `f`)`. The near clipping plane is $z = -n$. The far clipping plane is $z = -f$. The frustum is the set of points satisfying (II.16) and (II.17).

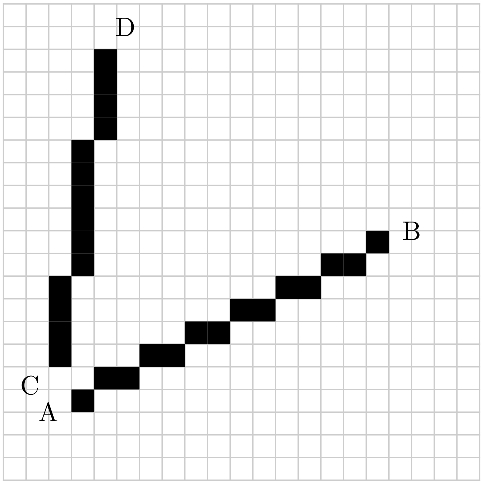


Figure II.24: The line segment AB has slope $\Delta j/\Delta i \leq 1$. The line segment CD has slope ≥ 1. The former segment is drawn with one pixel per column; the latter segment is drawn with one pixel per row.

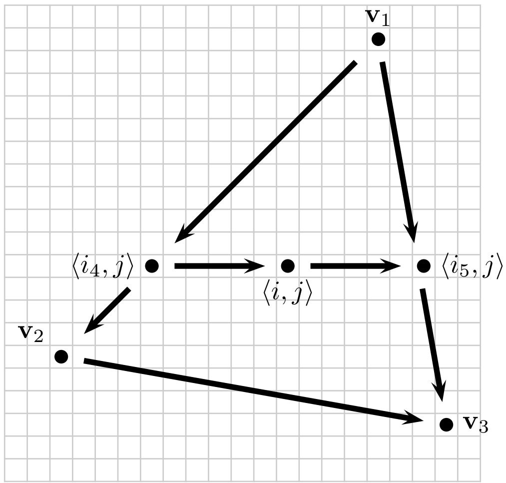


Figure II.25: The scan line interpolation method first interpolates along the edges of the triangle, then interpolates along the horizontal rows of pixels in the interior of the triangle. The interpolation directions are shown with arrows. If you look closely, you will note that the rightmost pixel, $\langle i_5, j\rangle$, on the horizontal scan line is not exactly on the line segment forming the right edge of the triangle — this is necessary since its position must be rounded to the nearest pixel.

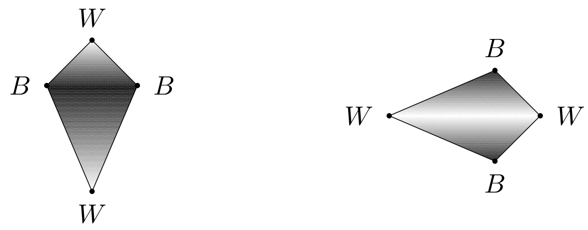


Figure II.26: Opposite vertices have the same black or white color. Scan line interpolation causes the appearance of the polygon to change radically when it is rotated. The two polygons are identical except for their orientation.

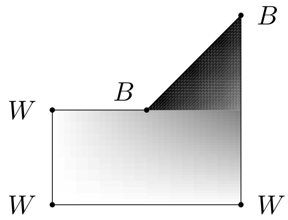


Figure II.27: Vertices are colored black or white as labeled. Scan line interpolation causes the non-convex polygon to be shaded discontinuously.

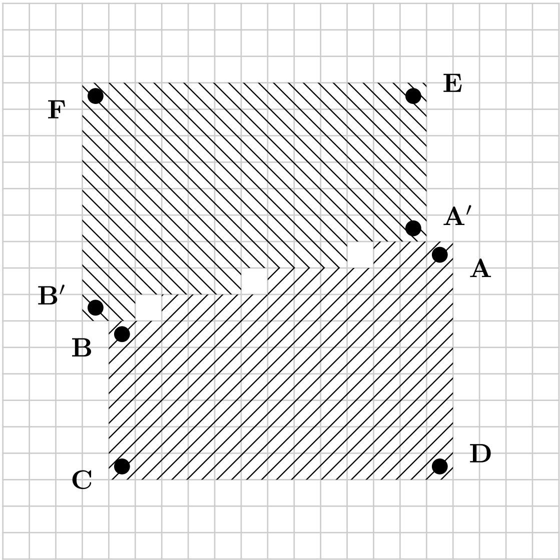


Figure II.28: The polygons $ABCD$ and $B'A'EF$ are supposed to share an edge, but arbitrarily small roundoff errors can cause a small displacement of the edge. This can lead to pixel-sized holes appearing between the two polygons. In the figure, the pixelized polygons are shown with different crosshatching: the three white pixels between the polygons are errors introduced by roundoff errors and will cause unwanted visual artifacts. This same effect can occur even in cases where only one of the vertices is affected by roundoff errors.

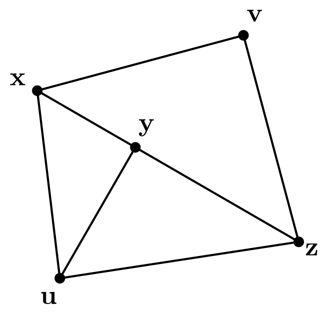
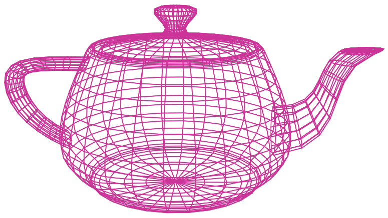


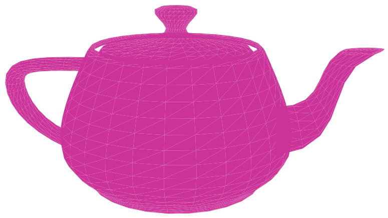
Figure II.29: Three triangles as placed by `glVertex*`. Even if no roundoff errors occur, the pixel-level discretization inherent in the Bresenham algorithm can leave pixel-sized gaps along the line $\overline{\mathbf{xz}}$.

Chapter III

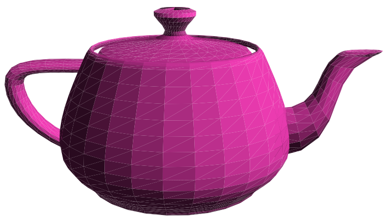
Lighting, Illumination, and Shading



(a)

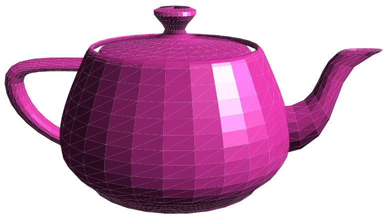


(b)



(c)

(d)



(e)

(f)

Figure III.1: Six teapots with various shading and lighting options. (a) Wireframe teapot. (b) Teapot drawn with solid color, but no lighting or shading. (c) Teapot with flat shading, with only ambient and diffuse lighting. (d) Teapot drawn with Gouraud interpolation, with only ambient and diffuse reflection. (e) Teapot drawn with flat shading, with ambient, diffuse, and specular lighting. (f) Teapot with Gouraud shading, with ambient, diffuse, and specular lighting. See color plate C.4.

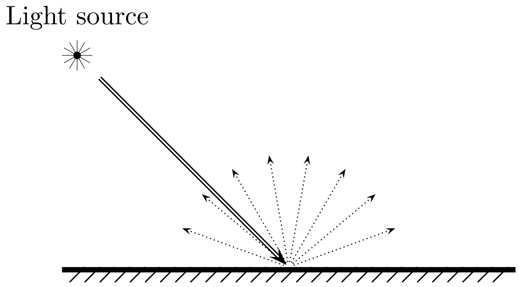


Figure III.2: Diffusely reflected light is reflected equally brightly in all directions. The double line is a beam of incoming light. The dotted arrows indicate outgoing light.

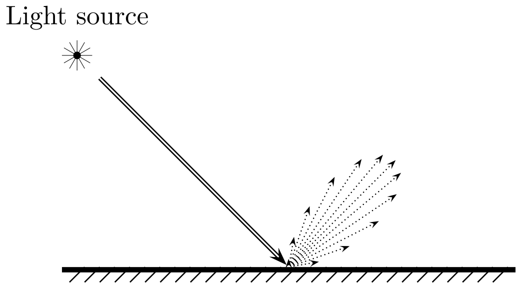


Figure III.3: Specularly reflected light is reflected primarily in the direction with the angle of incidence equal to the angle of reflection. The double line is a beam of incoming light. The dotted arrows indicate outgoing light; the longer the arrow, the more intense the reflection in that direction.

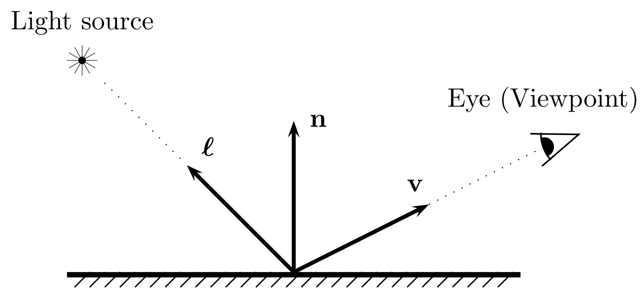


Figure III.4: The fundamental vectors of the Phong lighting model. The surface normal is the unit vector $\mathbf{n}$. The point light source is in the direction of the unit vector $\boldsymbol{\ell}$. The viewpoint (eye) is in the direction of the unit vector $\mathbf{v}$. The vectors $\boldsymbol{\ell}$, $\mathbf{n}$ and $\mathbf{v}$ are not necessarily coplanar.

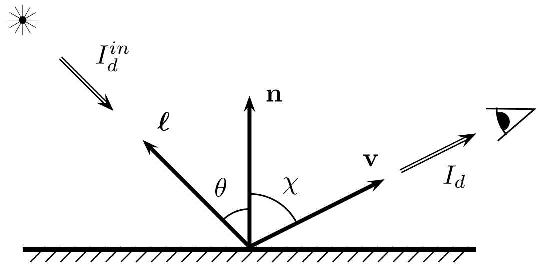


Figure III.5: The setup for diffuse reflection in the Phong model. The angle of incidence is θ. I_d^{in} and I_d are the incoming and outgoing light intensities in the indicated directions.

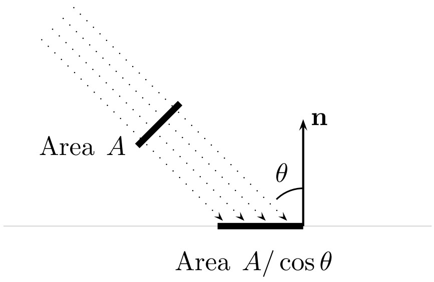


Figure III.6: The perpendicular cross sectional area of a beam of light is A. The area of the surface tilted at an angle θ is larger by a factor of $1/\cos\theta$.

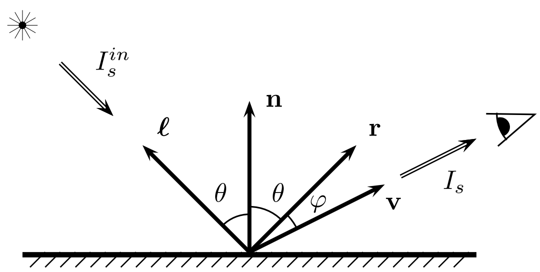


Figure III.7: The setup for specular reflection in the Phong model. The angle of incidence is θ. The vector $\mathbf{r}$ points in the direction of perfect mirror-like reflection. I_s^{in} and I_s are the incoming and outgoing specular light intensities in the indicated directions.

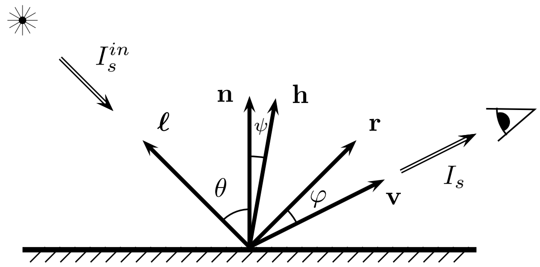
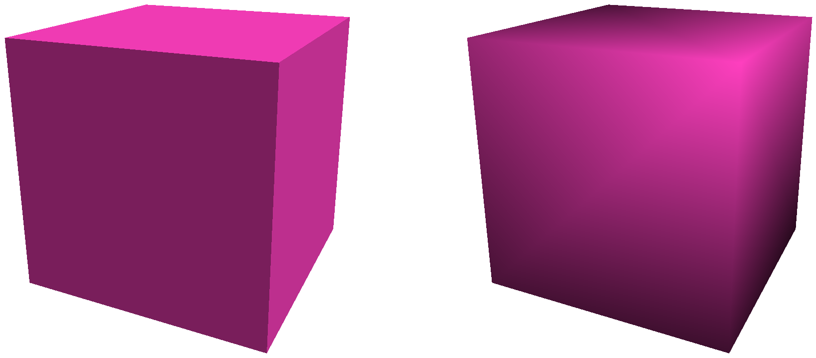


Figure III.8: The setup for calculating the specular reflection using the halfway vector $\mathbf{h}$, the unit vector halfway between $\boldsymbol{\ell}$ and $\mathbf{v}$.



(a) (b)

Figure III.9: Two cubes with (a) normals at vertices perpendicular to each face, and (b) normals outward from the center of the cube. Note that (a) is rendered with Gouraud shading, not flat shading. See color plate C.5.

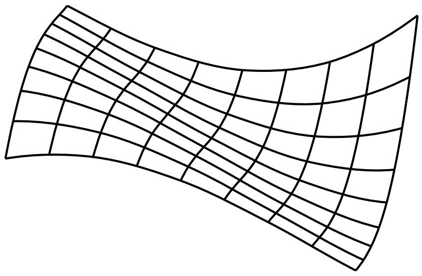


Figure III.10: A polygonal mesh defined by a parametric function. The horizontal and vertical curves are lines of constant y values and constant x values, respectively.

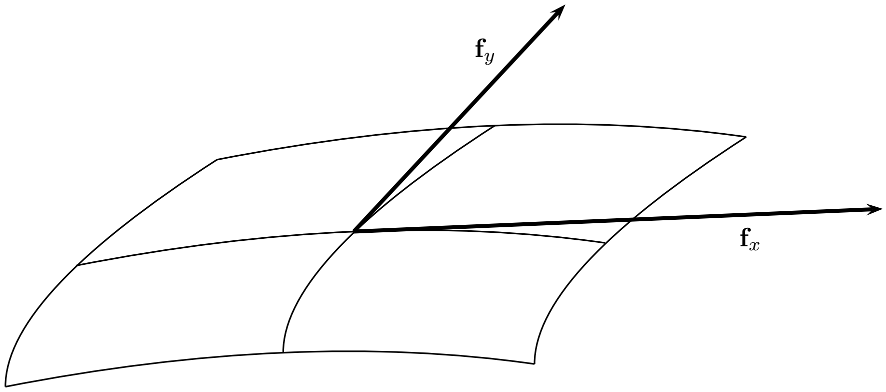


Figure III.11: A close-up view of a polygonal mesh. The partial derivatives are tangent to the horizontal and vertical cross section curves.

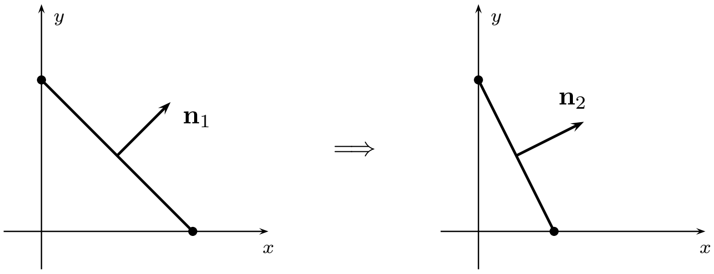


Figure III.12: An example of how a non-uniform scaling transformation affects a normal. The transformation maps $\langle x, y\rangle$ to $\langle \frac{1}{2}x, y\rangle$. The line with unit normal $\mathbf{n}_1 = \langle \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\rangle$ is transformed to a line with unit normal $\mathbf{n}_2 = \langle \frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}}\rangle$.

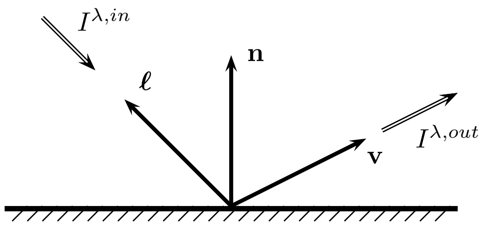


Figure III.13: The *BRIDF* function relates the outgoing light intensity and the incoming light intensity according to $BRIDF(\boldsymbol{\ell}, \mathbf{v}, \lambda) \ = \ I^{\lambda,out}/I^{\lambda,in}$.

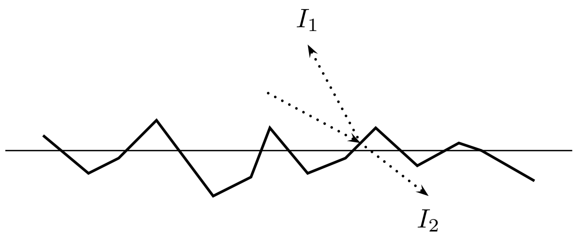


Figure III.14: A microfacet surface consists of small flat pieces. The horizontal line shows the average level of a flat surface, and the microfacets show the microscopic shape of the surface. Dotted lines show the direction of light rays. The incoming light can either be reflected in the direction of perfect mirror-like reflection (I_1), or can enter the surface (I_2). In the second case, the light is modeled as eventually exiting the material as diffusely reflected light.

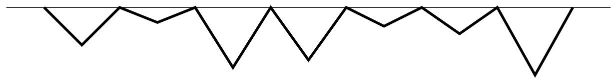
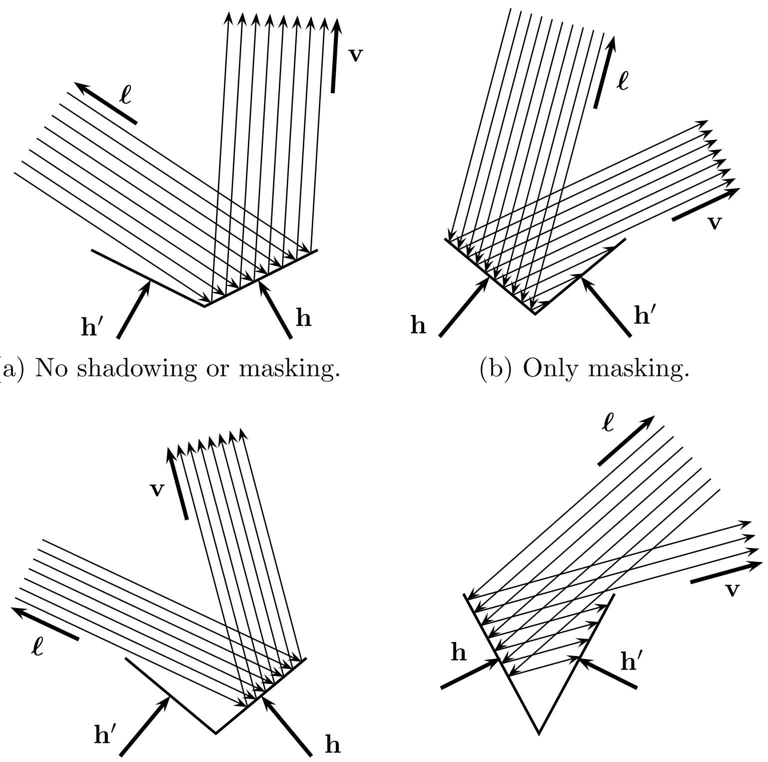


Figure III.15: For the derivation of the geometric term, G, the microfacets are modeled as symmetric, “V”-shaped grooves with the tops of the grooves all at the same height. The horizontal line shows the overall plane of the surface.



(a) No shadowing or masking.

(b) Only masking.

(c) Only shadowing.

(d) Both shadowing and masking.

Figure III.16: Shadowing and masking inside a single groove. The "V"-shape represents a groove; the unit vector $\mathbf{h}$ is normal to the facet where specular reflection occurs. Light from the direction of $\boldsymbol{\ell}$ is specularly reflected in the direction $\mathbf{v}$.

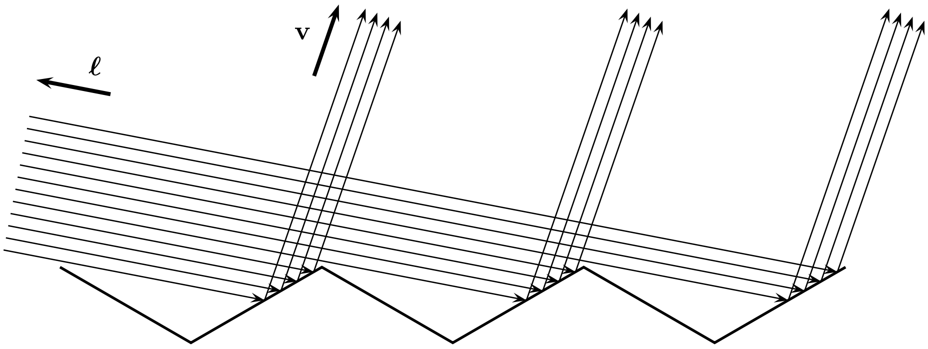


Figure III.17: Shadowing without masking does not reduce the intensity of the reflected light.

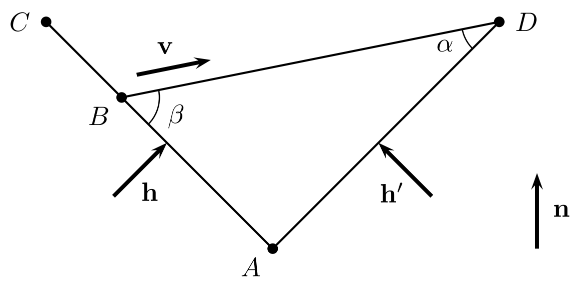


Figure III.18: The situation for Lemma III.4. The edges AC and AD form a symmetric groove, with AC and AD being of equal length. The vector $\mathbf{n}$ points upward, the vector $\mathbf{v}$ is in the direction from B to D. The vectors $\mathbf{h}$ and $\mathbf{h}'$ are normal to the sides of the groove. All four vectors are unit vectors. The ratio of $||BC||$ to $||AC||$ measures the fraction of the groove that is not masked.

	Red	Green	Blue
Gold:	0.93	0.88	0.38
Iridium:	0.26	0.28	0.26
Iron:	0.44	0.435	0.43
Nickel:	0.50	0.47	0.36
Copper:	0.93	0.80	0.46
Platinum:	0.63	0.62	0.57
Silver:	0.97	0.97	0.96

Figure III.19: Experimentally measured reflectances for perpendicularly incident light. Values based on [113].

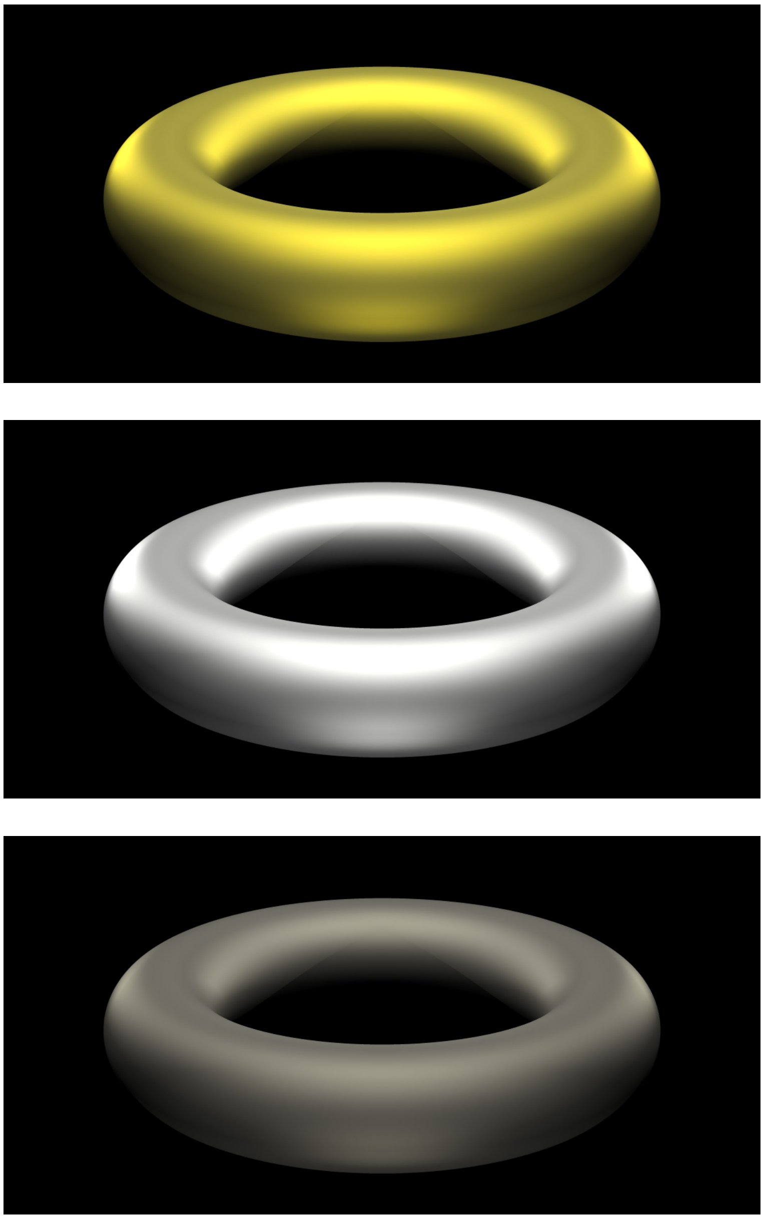


Figure III.20: Metallic tori, with specular component computed using the Cook-Torrance model. The materials are, from top to bottom, gold, silver and platinum. The roughness is $m = 0.4$ for all three materials. The tori are each illuminated by five positional white lights. See color plate C.16.

Chapter IV

Averaging and Interpolation

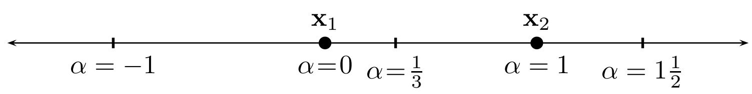


Figure IV.1: Interpolated and extrapolated points for various values of α. For $\alpha < 0$, $\mathbf{x}(\alpha)$ is to the left of $\mathbf{x}_1$. For $\alpha > 1$, $\mathbf{x}(\alpha)$ is to the right of $\mathbf{x}_2$. For $0 < \alpha < 1$, $\mathbf{x}(\alpha)$ is between $\mathbf{x}_1$ and $\mathbf{x}_2$.

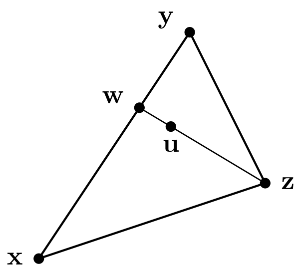


Figure IV.2: The point $\mathbf{u}$ in the interior of the triangle is on the line segment from $\mathbf{w}$ to $\mathbf{z}$. The point $\mathbf{w}$ is a weighted average of $\mathbf{x}$ and $\mathbf{y}$. The point $\mathbf{u}$ is a weighted average of $\mathbf{w}$ and $\mathbf{z}$.

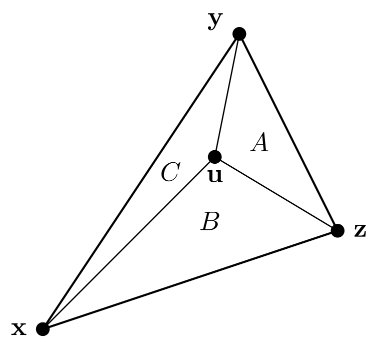


Figure IV.3: The barycentric coordinates α, β and γ for the point $\mathbf{u}$ are proportional to the areas A, B and C.

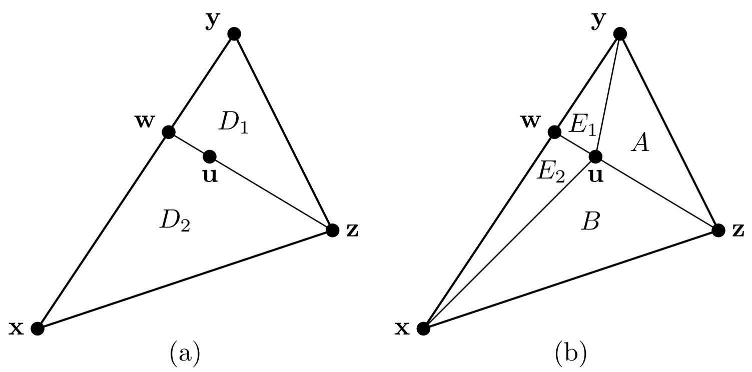


Figure IV.4: The areas used in the proof of Theorem IV.4.

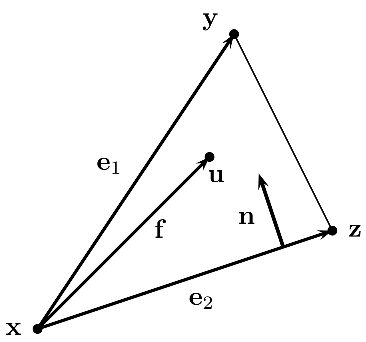


Figure IV.5: Calculating barycentric coordinates in $\mathbb{R}^3$.

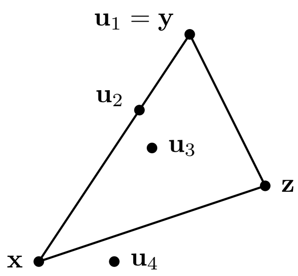


Figure IV.6: The points from exercise IV.5.

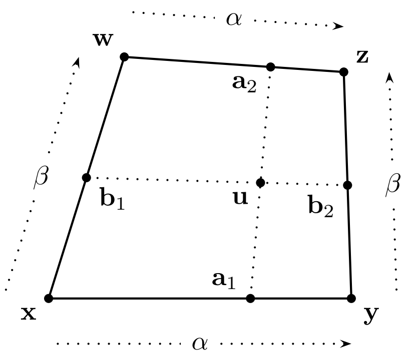


Figure IV.7: The point $\mathbf{u} = \mathbf{u}(\alpha, \beta)$ is formed by bilinear interpolation with the scalar coordinates α and β. The points $\mathbf{a}_1$ and $\mathbf{a}_2$ are obtained by interpolating with α, and $\mathbf{b}_1$ and $\mathbf{b}_2$ are obtained by interpolating with β.

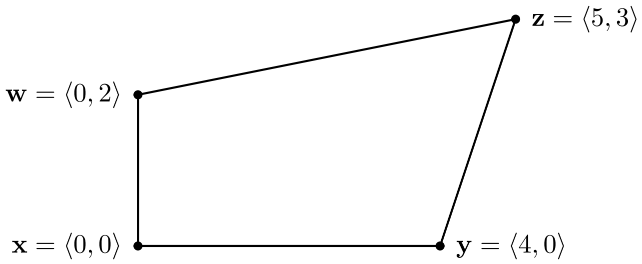


Figure IV.8: Figure for exercise IV.7.

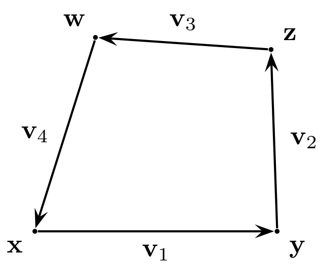


Figure IV.9: The vectors $\mathbf{v}_i$ are the directed edges around the quadrilateral.

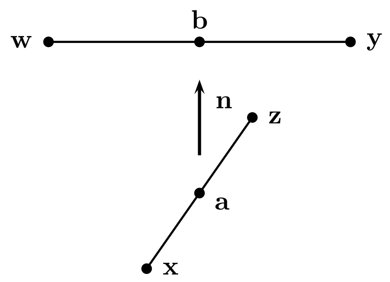


Figure IV.10: The line segments $\overline{\mathbf{xz}}$ and $\overline{\mathbf{yw}}$ have midpoints $\mathbf{a}$ and $\mathbf{b}$. The vector $\mathbf{n}$ is the unit vector in the direction from $\mathbf{a}$ to $\mathbf{b}$.

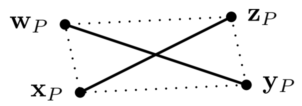


Figure IV.11: The projections of the two diagonals onto the plane P are non-collinear and intersect at their midpoints, at the common projection of $\mathbf{a}$ and $\mathbf{b}$. The four projected vertices form a convex quadrilateral.

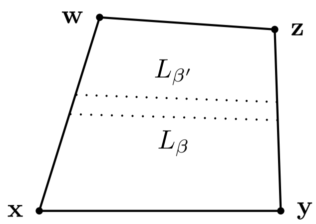


Figure IV.12: Since the polygon is convex, distinct values β and β' give non-intersecting "horizontal" line segments.

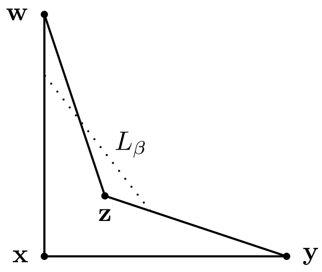


Figure IV.13: An example of the failure of Theorem IV.6 for non-convex, planar quadrilaterals.

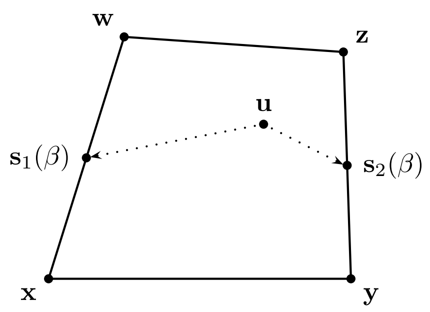


Figure IV.14: The three points $\mathbf{s}_1(\beta)$, $\mathbf{u}$, and $\mathbf{s}_2(\beta)$ will be collinear for the correct value of β. The value of β shown in the figure is smaller than the correct β coordinate of $\mathbf{u}$.

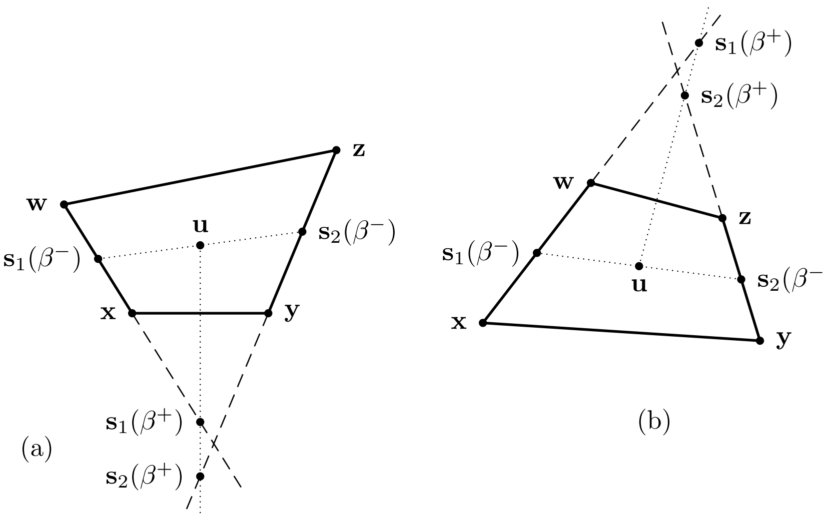


Figure IV.15: The two possibilities for the sign of $\mathbf{s}_1 \times \mathbf{s}_2$. In (a), $\mathbf{s}_1 \times \mathbf{s}_2 < 0$; in (b), $\mathbf{s}_1 \times \mathbf{s}_2 > 0$. In each case, there are two values for β where the points $\mathbf{s}_1(\beta)$, $\mathbf{s}_2(\beta)$, and $\mathbf{u}$ are collinear. The values β^+ and β^- are the solutions to equation (IV.22) obtained with the indicated choice of plus/minus sign. For both (a) and (b), $\beta = \beta^-$ is between 0 and 1 and is the desired root.

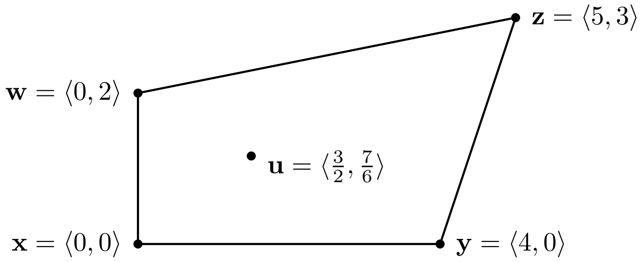


Figure IV.16: Figure for exercise IV.11.

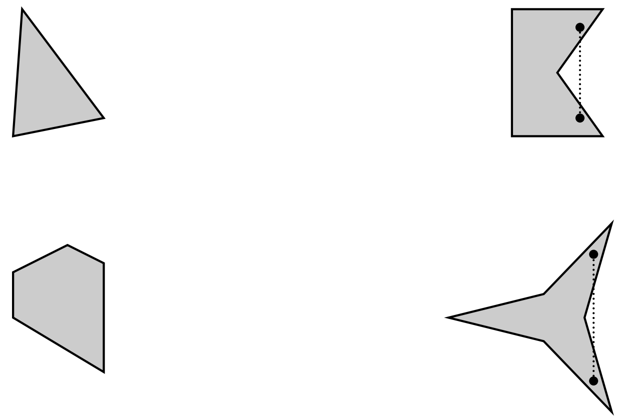


Figure IV.17: The shaded regions represent sets. The two sets on the left are convex, and the two sets on the right are not convex. The dotted lines show line segments with endpoints in the set which are not entirely contained in the set.

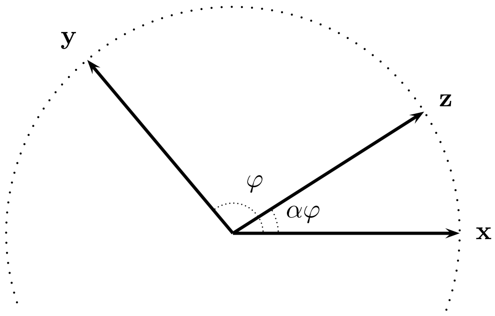


Figure IV.18: The angle between $\mathbf{x}$ and $\mathbf{y}$ is φ. $slerp(\mathbf{x}, \mathbf{y}, \alpha)$ is the vector $\mathbf{z}$ which is obtained by rotating $\mathbf{x}$ a fraction α of the way towards $\mathbf{y}$. All vectors are unit vectors, as $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ lie on the unit sphere.

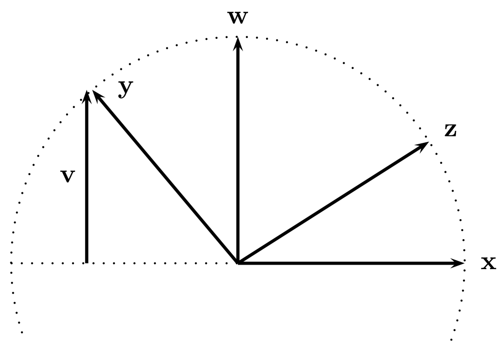


Figure IV.19: $\mathbf{v}$ and $\mathbf{w}$ are used to derive the formula for spherical linear interpolation. The vector $\mathbf{v}$ is the component of $\mathbf{y}$ perpendicular to $\mathbf{x}$, and $\mathbf{w}$ is the unit vector in the same direction. The magnitude of $\mathbf{v}$ is $\sin\varphi$.

Chapter V

Texture Mapping

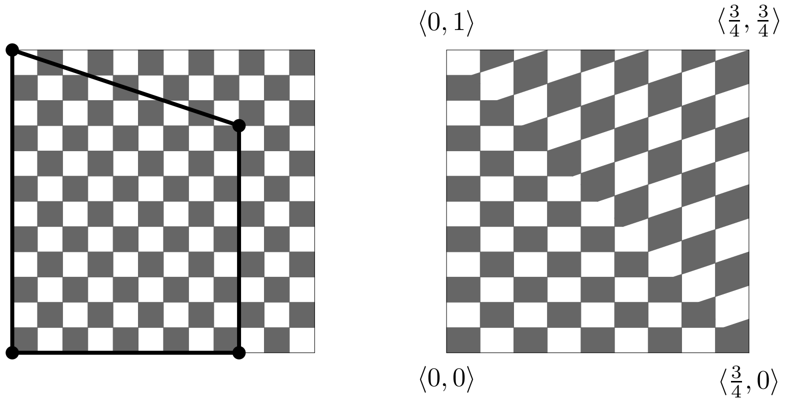
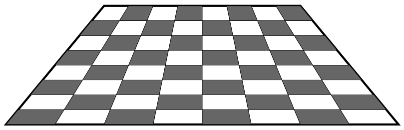


Figure V.1: The square on the left is a texture map. The square on the right is filled with a quadrilateral region of this texture map. The coordinates labeling the corners of the square are s,t values indexing into the texture map. The subregion of the checkerboard texture map selected by the s and t coordinates is shown in the left square. This subregion of the texture map was converted to two triangles first, and each triangle was mapped by linear interpolation into the corresponding triangle in the square on the right: this caused the visible diagonal boundary between the triangles.


Without hyperbolic interpolation

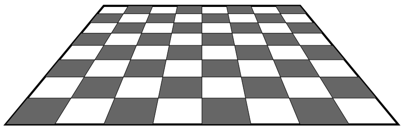

With hyperbolic interpolation

Figure V.2: The figure on the right uses hyperbolic interpolation to render the correct perspective foreshortening. The figure on the left does not.

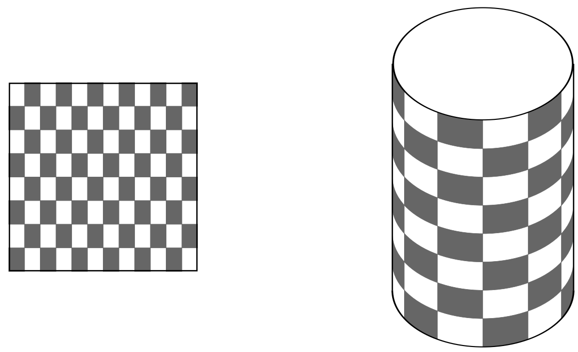


Figure V.3: A texture map and its application to a cylinder.

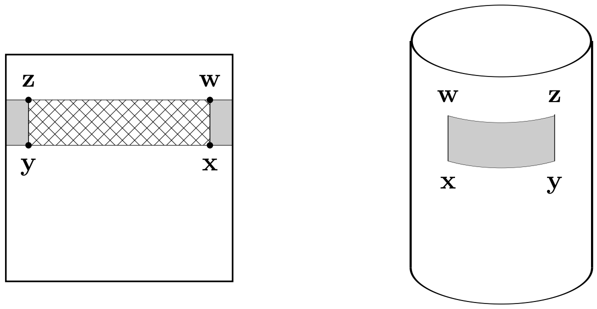


Figure V.4: The quadrilateral $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, $\mathbf{w}$ selects a region of the texture map. The crosshatched region of the texture map is *not* the intended region of the texture map. The shaded area is the intended region.

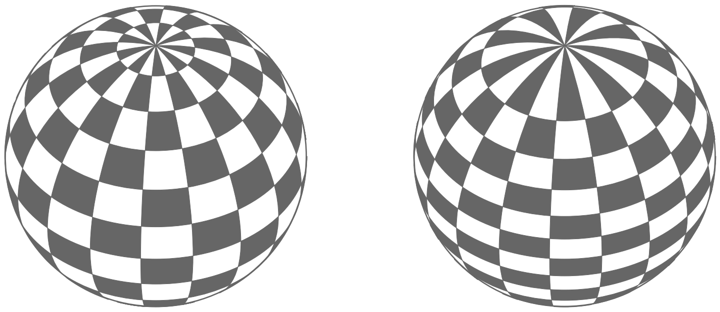


Figure V.5: Two applications of a texture map to a sphere. The sphere on the left has a checkerboard texture applied with texture coordinates given by the spherical map of equation (V.2). The sphere on the right uses texture coordinates given by the cylindrical projection of equation (V.3). The spheres are drawn with a tilt and a small rotation.

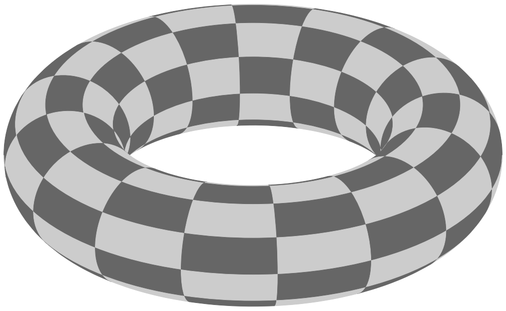


Figure V.6: A checkerboard texture map applied to a torus.

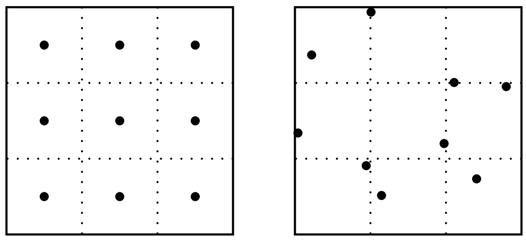


Figure V.7: In the first figure, the nine supersample points are placed at the centers of the nine subpixels. In the second figure, the supersample points are jittered, but are constrained to stay inside their subpixel.

Figure V.8: A bump mapped torus. Note the lack of bumps on the silhouette. There are four white lights shining on the scene, plus a low level of ambient illumination. This picture was generated with the ray tracing software described in appendix B. See color plate C.6.

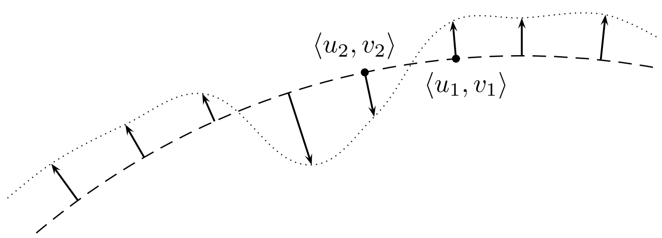


Figure V.9: The dashed curve represents a cross section of a two dimensional surface. The surface is imagined to be displaced perpendicularly a distance $d(u, v)$ to form the dotted curve. The outward direction of the surface is upward, and thus the value $d(u_1, v_1)$ is positive and the value $d(u_2, v_2)$ is negative.

Figure V.10: An environment map mapped into a sphere projection. This is the kind of environment map supported by OpenGL. See color plate C.7. The scene is the same as is shown in figure V.11. Note that the front wall has the most fidelity, and the back wall the least. For this reason, spherical environment maps are best used when the view direction is close to the direction used to create the environment map.

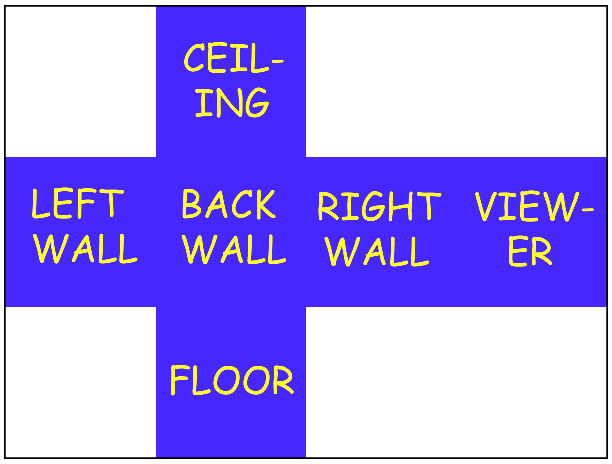


Figure V.11: An environment map mapped into a box projection consists of the six views from a point, mapped to the faces of a cube and then unfolded to make a flat image. This scene shows the reflection map from the point at the center of a room. The room is solid blue, except for yellow writing on the walls, ceiling and floor. The rectangular white regions of the environment map are not used. See color plate C.8.

Chapter VI

Color

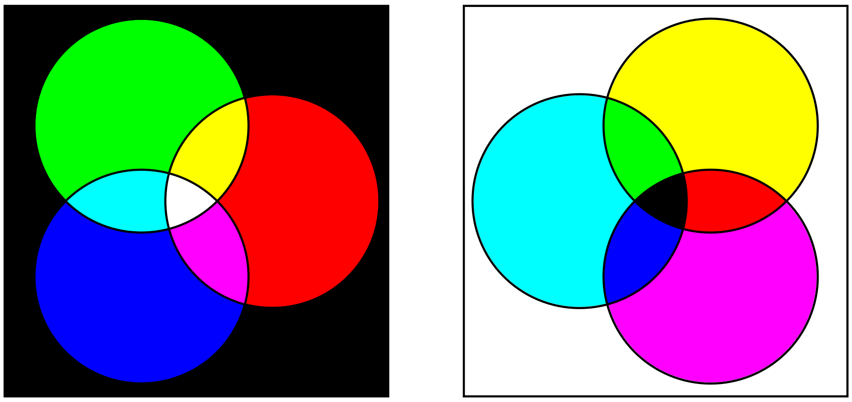


Figure VI.1: (a) The additive colors are red, green, and blue. (b) The subtractive colors are cyan, magenta, and yellow. See color plate C.2.

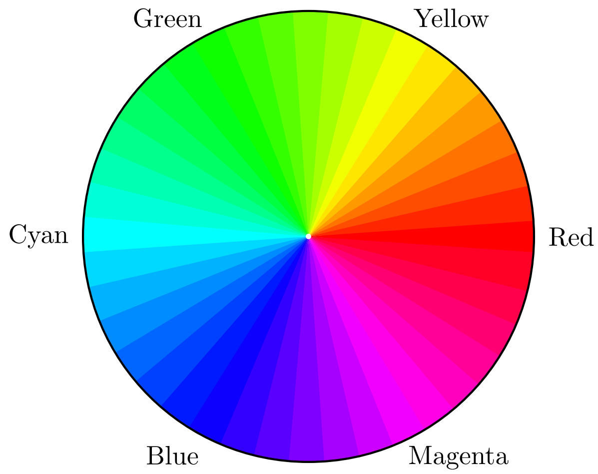


Figure VI.2: Hue is measured in degrees, representing an angle around the color wheel. Pure red has hue equal to 0, pure green has hue equal to 120, and pure blue has hue equal to 240. See color plate C.3.

Chapter VII

Bźier Curves

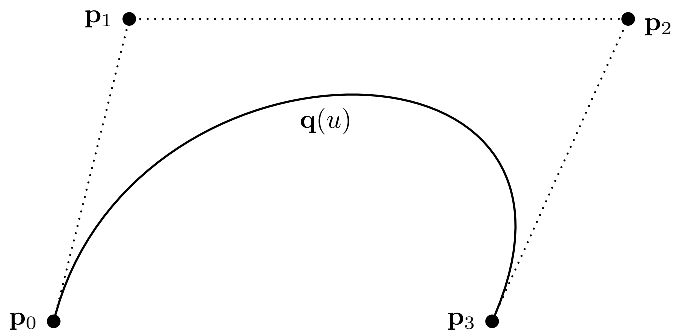


Figure VII.1: A degree three Bézier curve $\mathbf{q}(u)$. The curve is parametrically defined with $0 \le u \le 1$, and it interpolates the first and last control points with $\mathbf{q}(0) = \mathbf{p}_0$ and $\mathbf{q}(1) = \mathbf{p}_3$. The curve is "pulled towards" the middle control points $\mathbf{p}_1$ and $\mathbf{p}_2$. At $\mathbf{p}_0$, the curve is tangent to the line segment joining $\mathbf{p}_0$ and $\mathbf{p}_1$. At $\mathbf{p}_3$, it is tangent to the line segment joining $\mathbf{p}_2$ and $\mathbf{p}_3$.

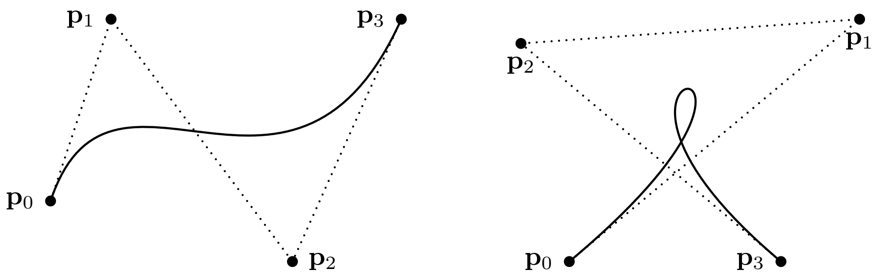


Figure VII.2: Two degree three Bézier curves, each defined by four control points. The curves interpolate only their first and last control points, $\mathbf{p}_0$ and $\mathbf{p}_3$. Note that, just as in figure VII.1, the curves start off, and end up, tangent to line segments joining control points.

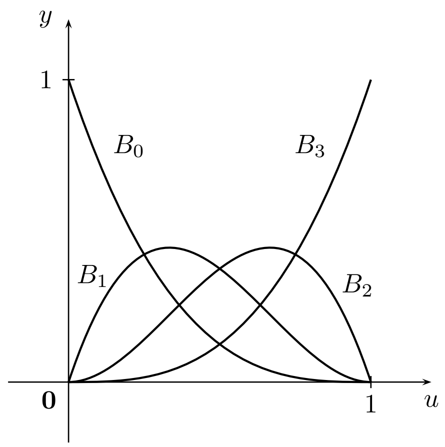


Figure VII.3: The four blending functions for degree three Bézier curves. We are only interested in their values in the interval $[0, 1]$. Each $B_i(u)$ is a degree three polynomial.

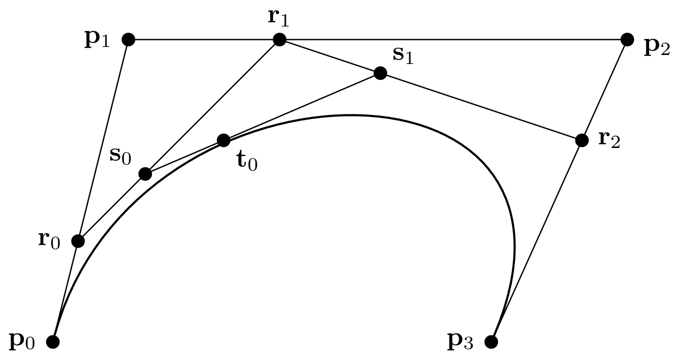


Figure VII.4: The de Casteljau method for computing $\mathbf{q}(u)$ for $\mathbf{q}$ a degree three Bézier curve. This illustrates the $u = 1/3$ case.

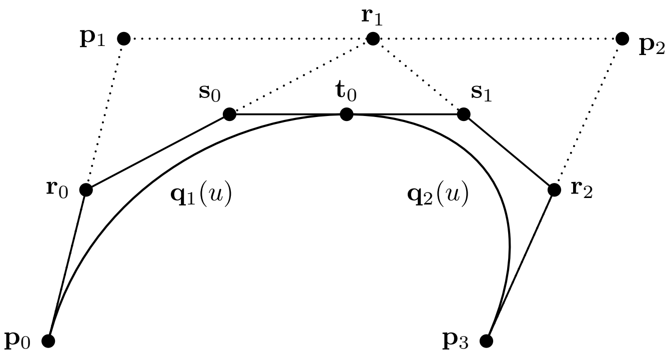


Figure VII.5: The de Casteljau method for computing $\mathbf{q}(u)$ for $\mathbf{q}$ a degree three Bézier curve is the basis for finding the new points needed for recursive subdivision. Shown here is the $u = 1/2$ case. The points $\mathbf{p}_0, \mathbf{r}_0, \mathbf{s}_0, \mathbf{t}_0$ are the control points for the Bézier curve $\mathbf{q}_1(u)$ which is equal to the first half of the curve $\mathbf{q}(u)$, i.e., starting at $\mathbf{p}_0$ and ending at $\mathbf{t}_0$. The points $\mathbf{t}_0, \mathbf{s}_1, \mathbf{r}_2, \mathbf{p}_3$ are the control points for the curve $\mathbf{q}_2(u)$ equal to the second half of $\mathbf{q}(u)$, i.e., starting at $\mathbf{t}_0$ and ending at $\mathbf{p}_3$.

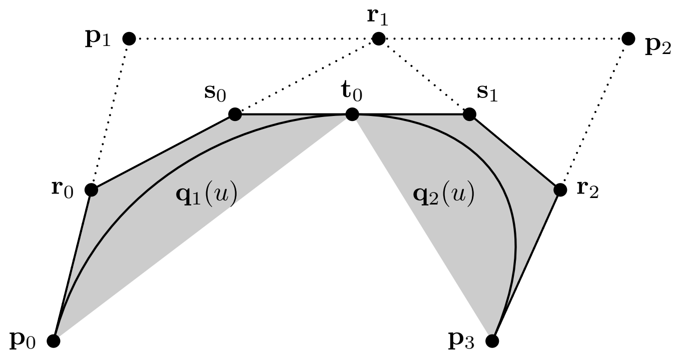


Figure VII.6: The convex hull of the control points of the Bézier curves shrinks rapidly during the process of recursive subdivision. The whole curve is inside its convex hull, i.e., inside the quadrilateral $\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$. After one round of subdivision, the two subcurves are known to be constrained in the two convex shaded regions.

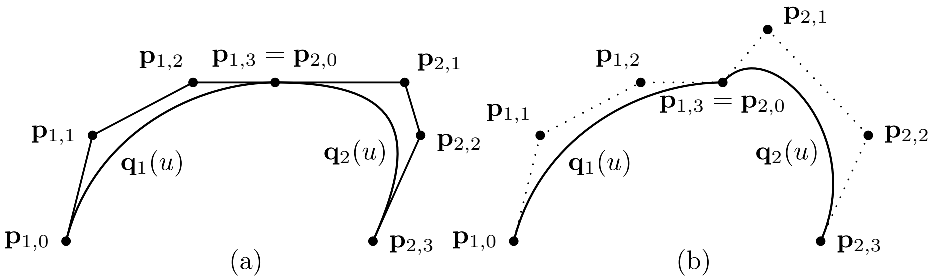


Figure VII.7: Two curves, each formed from two Bézier curves, with control points as shown. The curve in part (a) is G^1-continuous, but not C^1-continuous. The curve in part (b) is neither C^1-continuous nor G^1-continuous. Compare these curves to the curves of figures VII.5 and VII.6 which are both C^1-continuous and G^1-continuous.

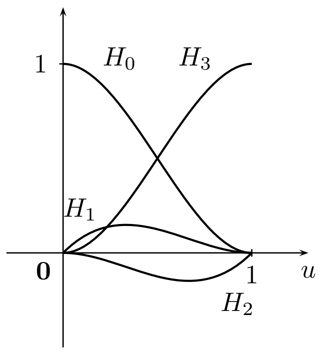
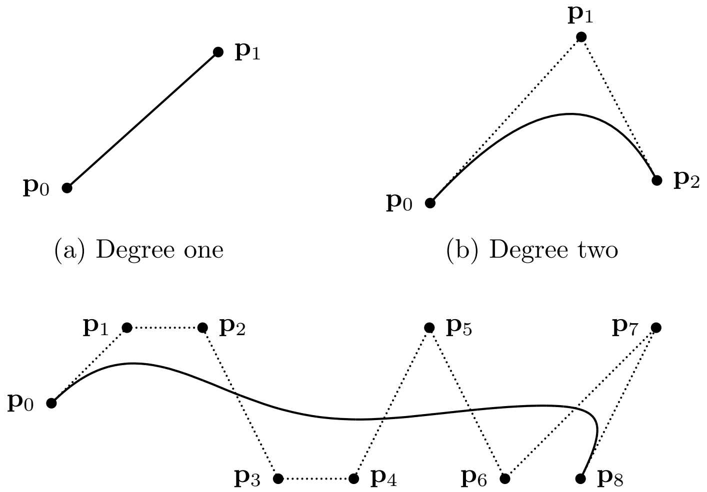


Figure VII.8: The degree three Hermite polynomials.



(a) Degree one

(b) Degree two

(c) Degree eight

Figure VII.9: (a) A degree one Bézier curve is just a straight line interpolating the two control points. (b) A degree two Bézier curve has three control points. (c) A degree eight Bézier curve has nine control points. The dotted straight line segments are called the *control polygon* of the Bézier curve.

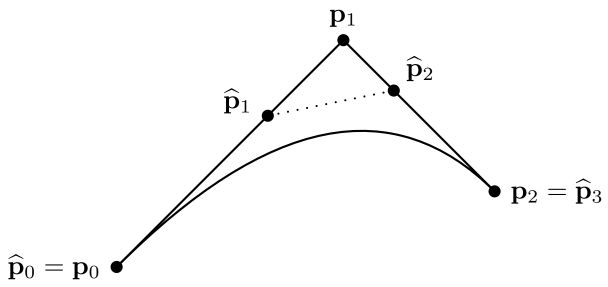


Figure VII.10: The curve $\mathbf{q}(u) = \widehat{\mathbf{q}}(u)$ is both a degree two Bézier curve with control points $\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$, and a degree three Bézier curve with control points $\widehat{\mathbf{p}}_0$, $\widehat{\mathbf{p}}_1$, $\widehat{\mathbf{p}}_2$, and $\widehat{\mathbf{p}}_3$.

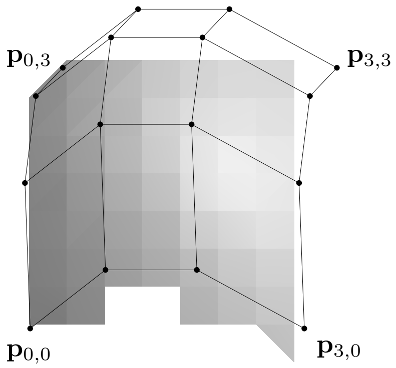


Figure VII.11: A degree three Bézier patch and its control points. The control points are shown joined by straight line segments.

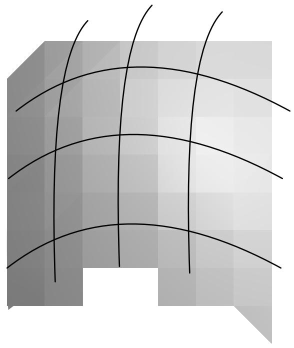


Figure VII.12: A degree three Bézier patch and some cross sections. The cross sections are Bézier curves.

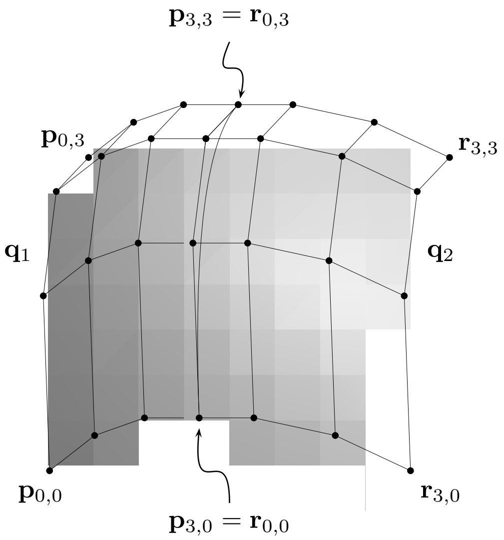


Figure VII.13: Two Bézier patches join to form a single smooth surface. The two patches $\mathbf{q}_1$ and $\mathbf{q}_2$ each have sixteen control points. The four rightmost control points of $\mathbf{q}_1$ are the same as the four leftmost control points of $\mathbf{q}_2$. The patches are shown forming a C^1-continuous surface.

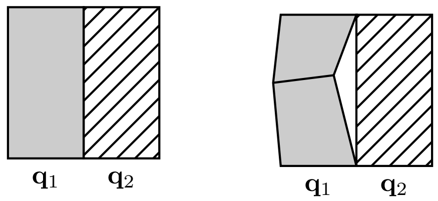


Figure VII.14: Non-uniform subdivision can cause cracking. On the left, two Bézier patches share a common boundary. On the right, after subdivision of the left patch $\mathbf{q}_1$, the boundaries no longer match up.

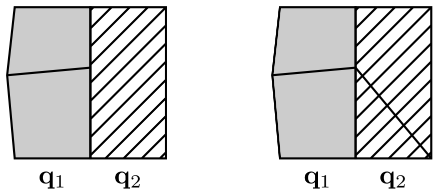


Figure VII.15: Two solutions to the cracking problem. On the left, the subdivided $\mathbf{q}_1$ and the original $\mathbf{q}_2$ share a common, straight boundary. However, the lighting and shading calculations may cause the surface to be rendered discontinuously at the boundary. On the right, the patch $\mathbf{q}_2$ has been subdivided in an ad hoc way to allow the common boundary to have the same points and normals with respect to both patches.

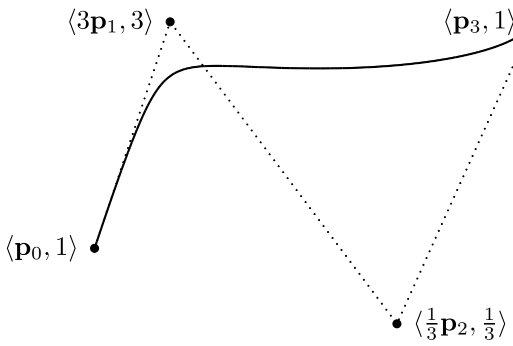


Figure VII.16: A degree three, rational Bézier curve. The control points are the same as in the left-hand side of figure VII.2 on page 156, but now the control point $\mathbf{p}_1$ is weighted 3, and the control point $\mathbf{p}_2$ is weighted only $1/3$. The other two control points have weight 1. In comparison with the curve of figure VII.2, this curve more closely approaches $\mathbf{p}_1$, but does not approach $\mathbf{p}_2$ nearly as closely.

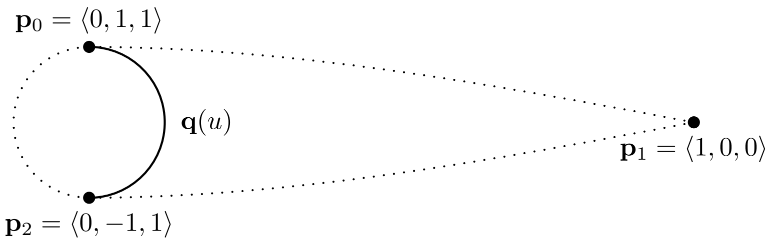


Figure VII.17: The situation of Theorem VII.9. The middle control point is actually a point at infinity, and the dotted lines joining it to the other control points are actually straight and are tangent to the circle at $\mathbf{p}_0$ and $\mathbf{p}_2$.

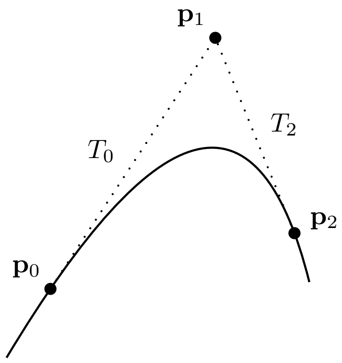


Figure VII.18: A portion of a branch of a conic section C is equal to a rational quadratic Bézier curve. Control points $\mathbf{p}_0$ and $\mathbf{p}_2$ have weight 1 and $\mathbf{p}_1$ gets weight $w_1 \geq 0$.

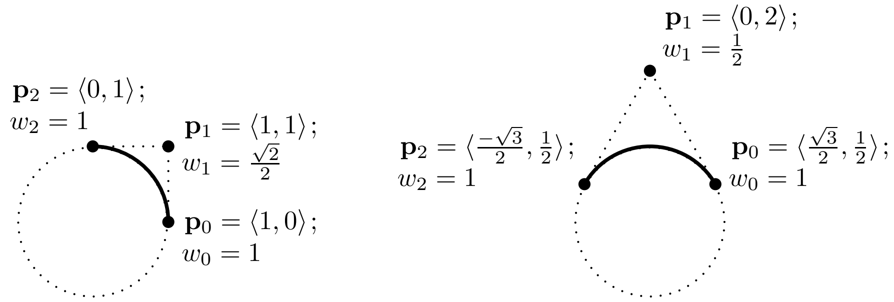


Figure VII.19: Two ways to define circular arcs with rational Bézier curves without control points at infinity.

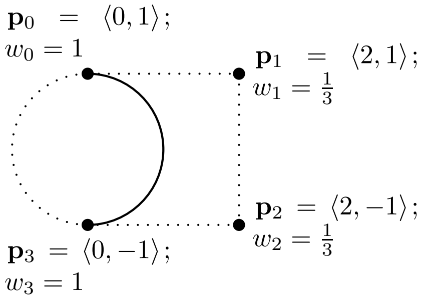


Figure VII.20: A semicircle as a degree three Bézier curve. See exercise VII.17.

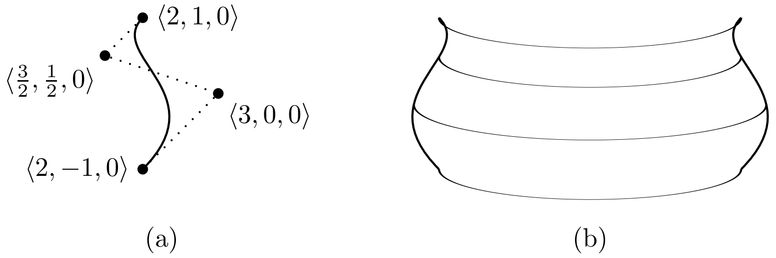


Figure VII.21: (a) A silhouette of a surface of revolution (the control points are in x, y, z-coordinates). (b) The front half of the surface of revolution. This example is implemented in the `SimpleNurbs` progam.

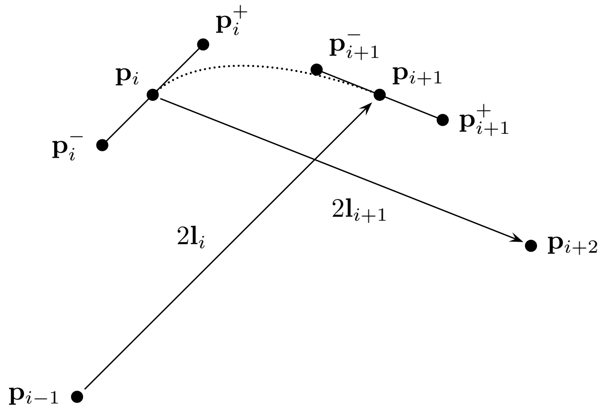
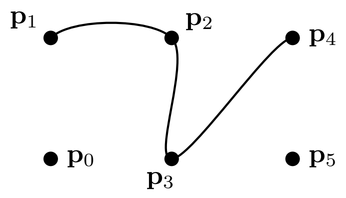


Figure VII.22: Defining the Catmull-Rom spline segment from the point $\mathbf{p}_i$ to the point $\mathbf{p}_{i+1}$. The points $\mathbf{p}_i^-$, $\mathbf{p}_i$, and $\mathbf{p}_i^+$ are collinear and parallel to $\mathbf{p}_{i+1} - \mathbf{p}_{i-1}$. The points $\mathbf{p}_i$, $\mathbf{p}_i^+$, $\mathbf{p}_{i+1}^-$, and $\mathbf{p}_{i+1}$ form the control points of a degree three Bézier curve, which is shown as a dotted curve.



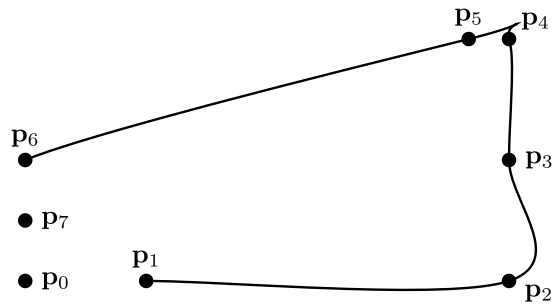
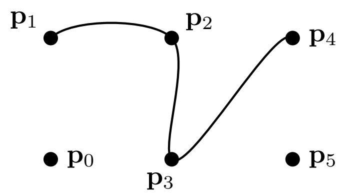


Figure VII.23: Two examples of Catmull-Rom splines with uniformly spaced knots.



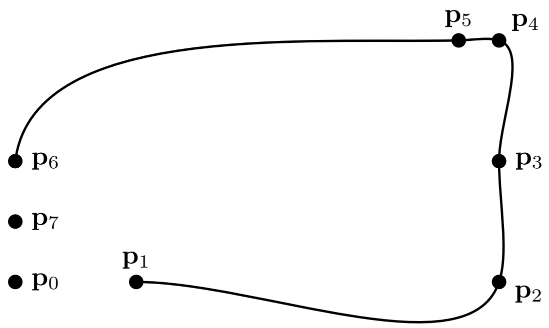


Figure VII.24: Two examples of Overhauser spline curves. The knot positions were set by chord-length parameterization. These are defined from exactly the same control points as the Catmull-Rom curves in figure VII.23.

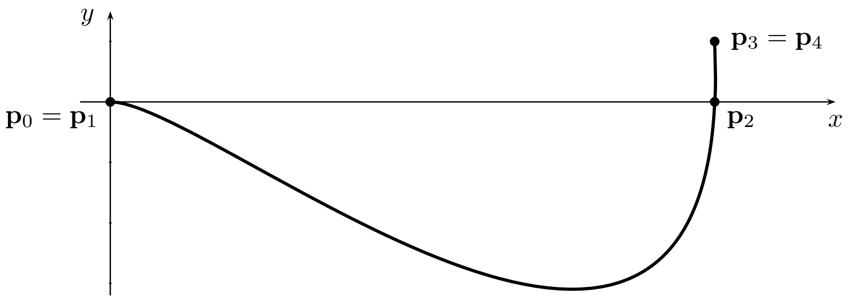


Figure VII.25: The Overhauser spline which is the solution to exercise VII.24.

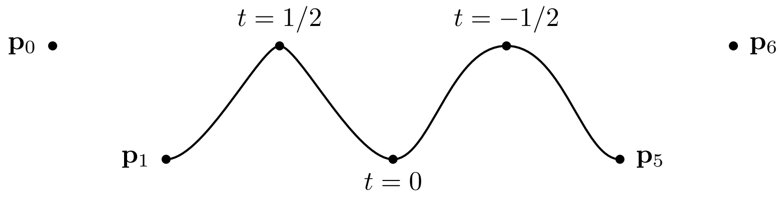


Figure VII.26: The effects of the tension parameter.

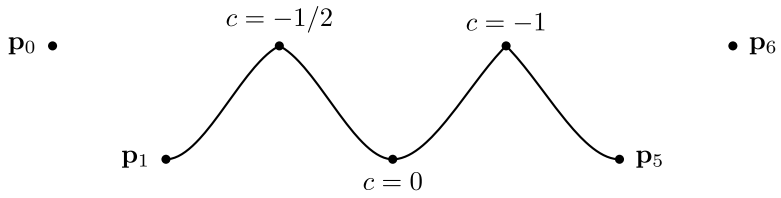


Figure VII.27: The effects of the continuity parameter.

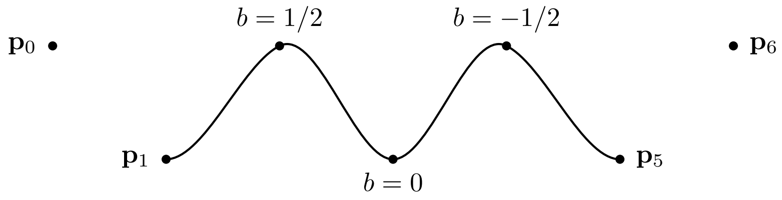
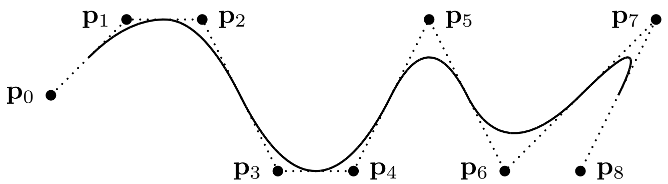


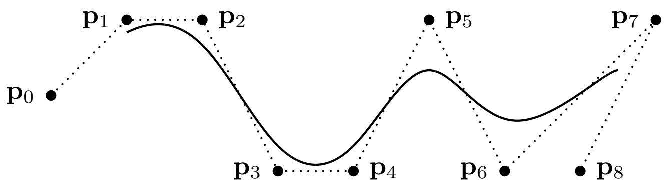
Figure VII.28: The effects of the bias parameter.

Chapter VIII

B-Splines



(a) Degree two B-spline curve.



(b) Degree three B-spline curve.

Figure VIII.1: Degree two and degree three B-spline curves with uniformly spaced knots and nine control points. The degree three curve is smoother than the degree two curve, whereas, the degree two curve approaches the control points a little more closely. Compare with the degree eight Bézier curve of figure VII.9(c) on page 167.

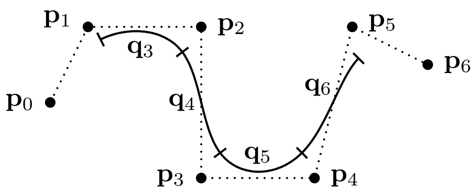


Figure VIII.2: A degree three uniform B-spline curve with seven control points.

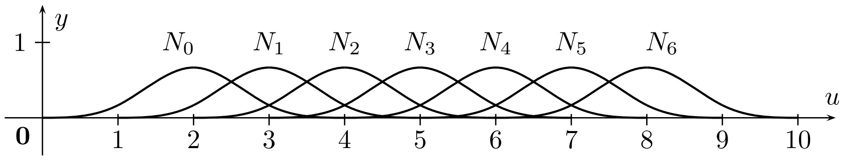


Figure VIII.3: The blending functions for a uniform, degree three B-spline. Each function N_i has support $(i, i+4)$.

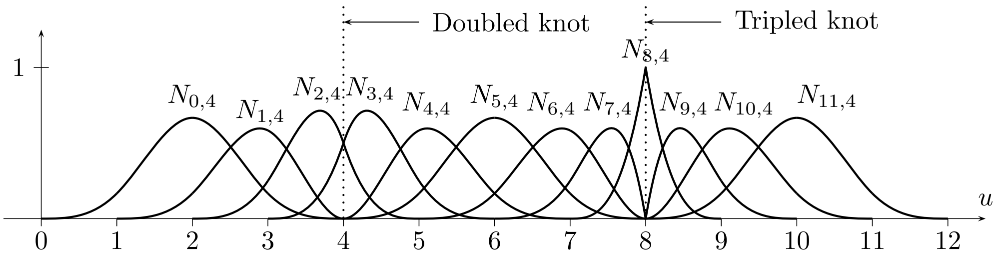


Figure VIII.4: Example of order four (degree three) blending functions with repeated knots. The knot vector is $[0, 1, 2, 3, 4, 4, 5, 6, 7, 8, 8, 8, 9, 10, 11, 12]$ so that the knot 4 has multiplicity two and the knot 8 has multiplicity three.

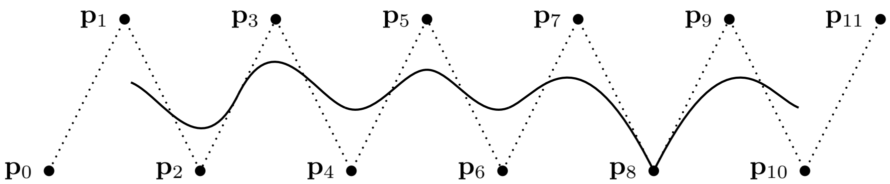


Figure VIII.5: Example of an order four B-spline created with repeated knots. This curve is created with the knot vector and blending functions shown in figure VIII.4. It has domain $[3, 9]$.

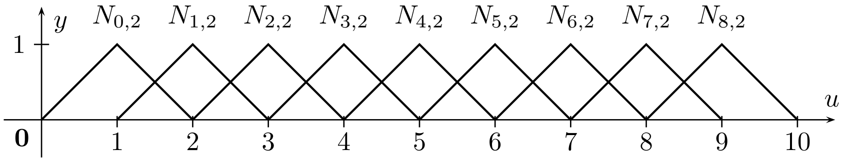


Figure VIII.6: The order two (piecewise degree one) blending functions with uniformly spaced knots, $u_i = i$. Here $\ell = 10$, and there are $\ell + 1$ knots and $\ell - 1$ blending functions. The associated B-spline curve of equation (VIII.2) is defined for $1 \leq u \leq \ell - 1$.

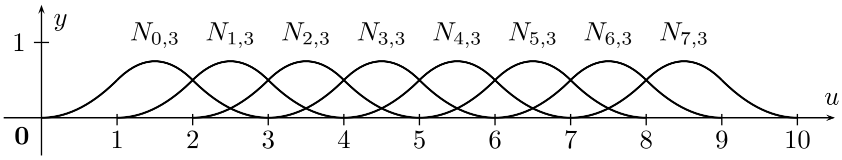
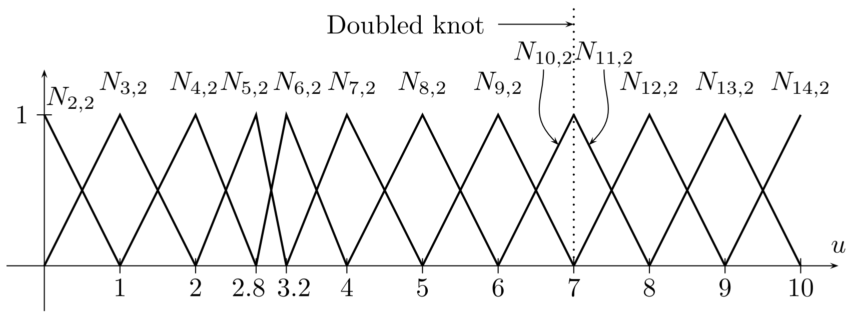
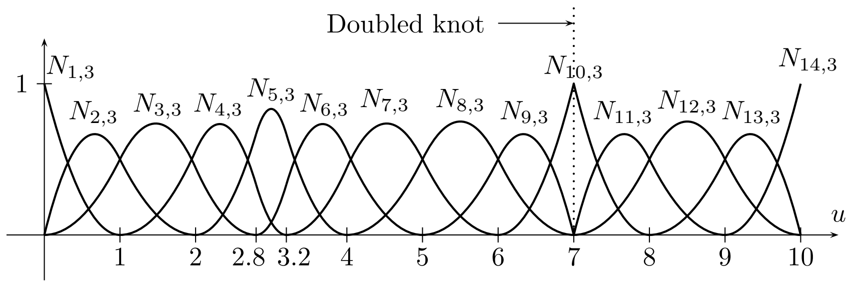


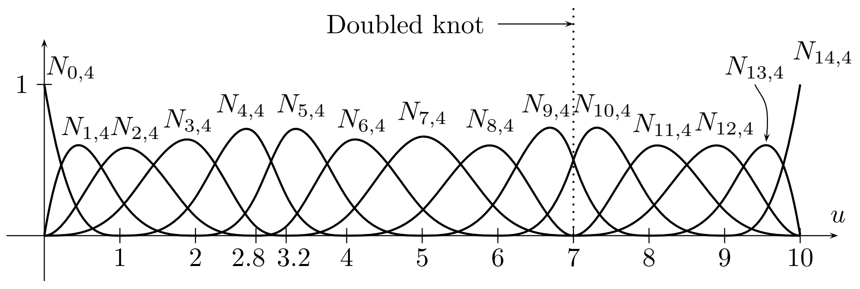
Figure VIII.7: The order three (piecewise degree two) blending functions with uniform knot positions $u_i = i$. We still have $\ell = 10$; there are $\ell + 1$ knots and $\ell - 2$ blending functions. The associated B-spline curve of equation (VIII.3) is defined for $2 \leq u \leq \ell - 2$.



(a) Degree one blending functions.



(b) Degree two blending functions.



(c) Degree three blending functions.

Figure VIII.8: Degree one, two, and three blending functions for a non-uniform knot sequence. The knot 7 has multiplicity two, and the knots 0 and 10 have multiplicity 4.

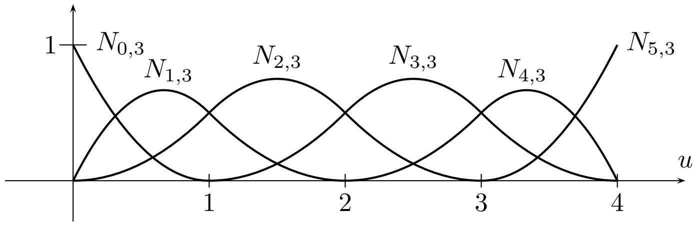


Figure VIII.9: The degree two blending functions, $N_{i,3}(u)$, for the knot vector of exercise VIII.6.

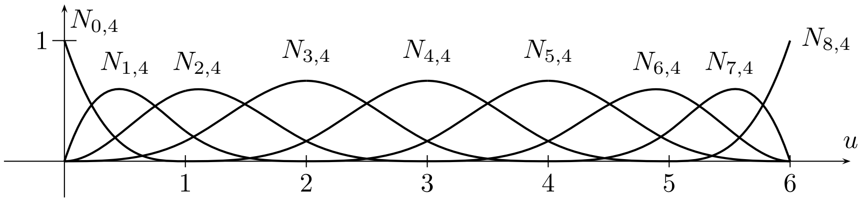


Figure VIII.10: The degree three blending functions, $N_{i,4}(u)$, for the knot vector $[0,0,0,0,1,2,3,4,5,6,6,6,6]$ of exercise VIII.7.

$$
\begin{array}{cccccccccccc}
 & & & & & & & & & \mathbf{p}_{s-k}^{(0)} & = & \mathbf{p}_{s-k} \\
 & & & & & & & & \swarrow & & & \\
 & & & & & & & \mathbf{p}_{s-k+1}^{(1)} & \leftarrow & & & \\
 & & & & & & \swarrow & & & & & \\
 & & & & & \mathbf{p}_{s-k+2}^{(2)} & \leftarrow & & & \vdots & & \vdots \\
 & & & & & & & \vdots & & & & \\
 & & & & \cdot^{\cdot^{\cdot}} & \vdots & & & \swarrow & & & \\
 & & & & & & & \mathbf{p}_{s-2}^{(1)} & \leftarrow & \mathbf{p}_{s-2}^{(0)} & = & \mathbf{p}_{s-2} \\
 & & & \swarrow & & & \swarrow & & \swarrow & & & \\
 & & \mathbf{p}_{s-1}^{(k-1)} & \leftarrow & & \mathbf{p}_{s-1}^{(2)} & \leftarrow & \mathbf{p}_{s-1}^{(1)} & \leftarrow & \mathbf{p}_{s-1}^{(0)} & = & \mathbf{p}_{s-1} \\
 & \swarrow & & \swarrow & & & \swarrow & & \swarrow & & & \\
\mathbf{p}_{s}^{(k)} & \leftarrow & \mathbf{p}_{s}^{(k-1)} & \leftarrow & \cdots & \mathbf{p}_{s}^{(2)} & \leftarrow & \mathbf{p}_{s}^{(1)} & \leftarrow & \mathbf{p}_{s}^{(0)} & = & \mathbf{p}_{s}
\end{array}
$$

Figure VIII.11: The control points obtained as $\mathbf{q}(u)$ is expressed as B-spline curves of lower degrees. For $j > 0$, the values $\mathbf{p}_i^{(j)}$ depend on u.

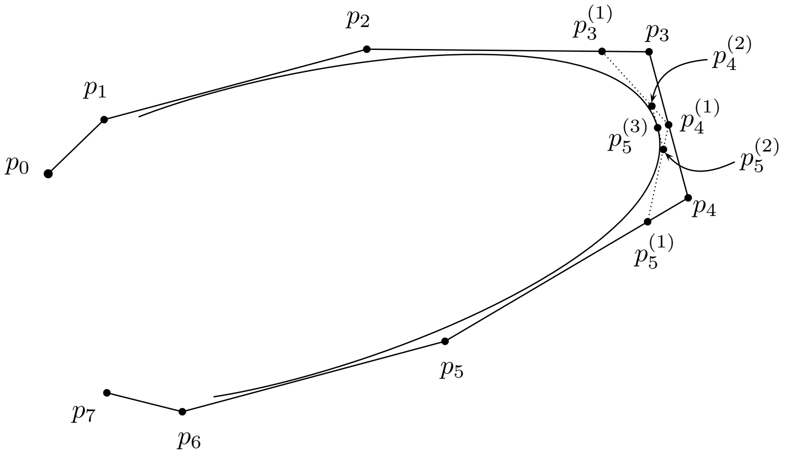
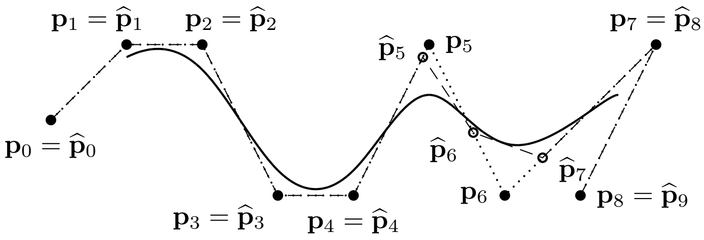


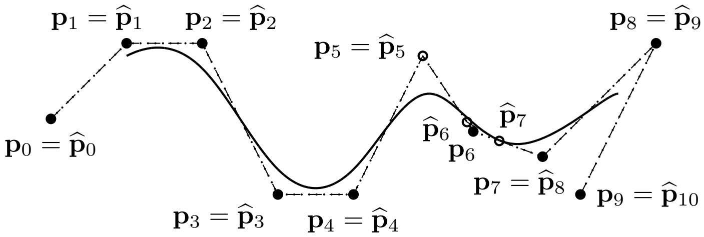
Figure VIII.12: The use of the de Boor algorithm to compute $\mathbf{q}(u)$. The degree three spline has the uniform knot vector $u_i = i$ for $0 \le i \le 11$ and control points $\mathbf{p}_i$. The points $\mathbf{p}_i^{(j)}$ are computed by the de Boor algorithm with $u = 5\frac{1}{2}$, and $\mathbf{p}_5^{(3)} = \mathbf{q}(5\frac{1}{2})$.

$$
\begin{array}{ccccccccc}
 & & & & & & & & \mathbf{b}(u_{[s-k+1,s]}) \\
 & & & & & & & \swarrow & \\
 & & & & & & \mathbf{b}(u_{[s-k+2,s]}, u) & \longleftarrow & \\
 & & & & & \swarrow & & & \\
 & & & & \mathbf{b}(u_{[s-k+3,s]}, u^{\langle 2\rangle}) & \longleftarrow & & & \vdots \\
 & & & & & & \vdots & & \\
 & & & & & & & \swarrow & \\
 & & \iddots & & \vdots & & \mathbf{b}(u_{[s-1,s+k-3]}, u) & \longleftarrow & \mathbf{b}(u_{[s-1,s+k-2]}) \\
 & & & & & \swarrow & & \swarrow & \\
 & & & & \mathbf{b}(u_{[s,s+k-3]}, u^{\langle 2\rangle}) & \longleftarrow & \mathbf{b}(u_{[s,s+k-2]}, u) & \longleftarrow & \mathbf{b}(u_{[s,s+k-1]}) \\
 & \swarrow & & & & \swarrow & & \swarrow & \\
\mathbf{b}(u^{\langle k\rangle}) & \longleftarrow & \cdots & & \mathbf{b}(u_{[s+1,s+k-2]}, u^{\langle 2\rangle}) & \longleftarrow & \mathbf{b}(u_{[s+1,s+k-1]}, u) & \longleftarrow & \mathbf{b}(u_{[s+1,s+k]})
\end{array}
$$

Figure VIII.13: A table of blossom values. The value $\mathbf{b}(u^{\langle k\rangle})$ on the left is equal to $\mathbf{q}_s(u)$. The blossom values in the right column are equal to the control points of the B-spline curve. The symmetry and multiaffine properties of the blossom function mean that each blossom value is a weighted average of the two blossom values that point to it, as expressed in Equation (VIII.17).



(a) Knot vector becomes $[0, 1, 2, 3, 4, 5, 6, 7, 7\frac{3}{4}, 8, 9, 10, 11]$.



(b) Knot vector becomes $[0, 1, 2, 3, 4, 5, 6, 7, 7\frac{3}{4}, 7\frac{3}{4}, 8, 9, 10, 11]$.

Figure VIII.14: Showing the insertion of knots into a degree three curve. The original knot vector is the uniform knot vector $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$. We insert the value $7\frac{3}{4}$ into the curve twice, each time adding a new control point, and making the control polygon more closely approximate the curve near $7\frac{3}{4}$. The dotted straight lines show the control polygon before the insertion of the new knot position. The dashed straight lines are the control polygon after the insertion. (In (b), the dashed line from $\widehat{\mathbf{p}}_6$ to $\widehat{\mathbf{p}}_7$ is so close to the curve that it cannot be seen in the graph.) The filled circles are the original control point positions. The open circles are the changed control point positions. The control points $\widehat{\mathbf{p}}_i$ of (a) are renamed $\mathbf{p}_i$ in (b). In both figures, one new knot has been inserted and some of the control points have been moved, but the B-spline curve itself is unchanged. If we inserted $7\frac{3}{4}$ a third time, then the new control point $\widehat{\mathbf{p}}_7$ would be equal to the point on the curve at $u = 7\frac{3}{4}$.

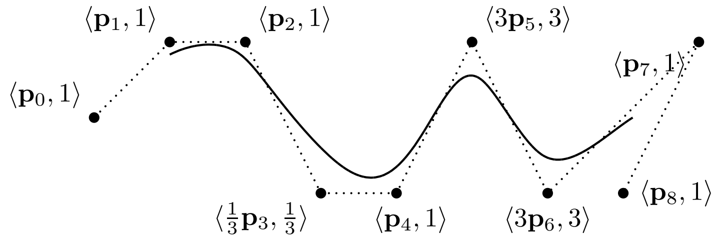


Figure VIII.15: A degree three, rational B-spline curve. The control points are the same as in figure VIII.1 on page 201, but now the control point $\mathbf{p}_3$ is weighted only $1/3$, and the two control points $\mathbf{p}_5$ and $\mathbf{p}_6$ are weighted 3. All other control points have weight 1. In comparison with the curve of figure VIII.1(b), this curve more closely approaches $\mathbf{p}_5$ and $\mathbf{p}_6$, but does not approach $\mathbf{p}_3$ as closely.

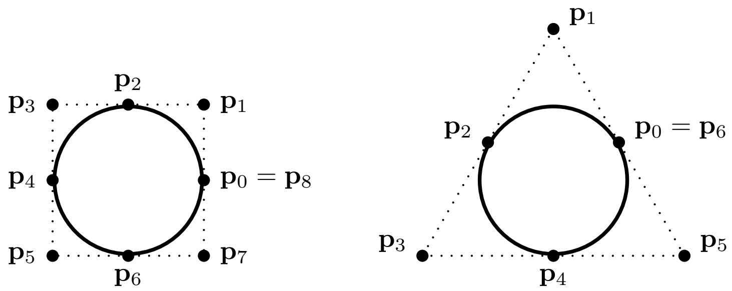


Figure VIII.16: Two ways to form a complete circle with a quadratic B-spline curve. The first curve has knot vector $[0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4]$ and the control points $\mathbf{p}_i$ have weight 1 when i is even and weight $\frac{\sqrt{2}}{2}$ when i is odd. The second curve has knot vector $[0, 0, 0, 1, 1, 2, 2, 3, 3, 3]$ and the control points $\mathbf{p}_i$ have weight 1 when i is even and weight $\frac{1}{2}$ when i is odd.

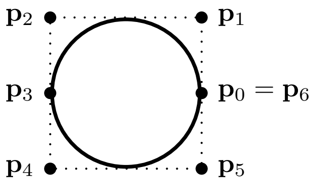


Figure VIII.17: Another way to form a complete circle with a quadratic B-spline curve. The curve has knot vector $[0, 0, 0, 1, 2, 2, 3, 4, 4, 4]$ and the control points $\mathbf{p}_0$, $\mathbf{p}_3$, and $\mathbf{p}_6$ have weight 1, and the other control points $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_4$, and $\mathbf{p}_5$ have weight $\frac{1}{2}$. Exercise VIII.11 on page 225 shows a way to prove the correctness of this B-spline curve.

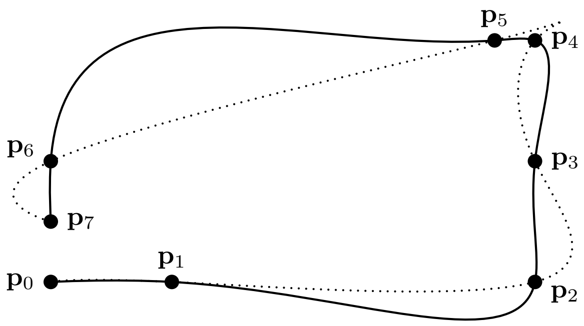


Figure VIII.18: Degree three interpolating spline. The dotted curve uses uniform knot spacing. The solid curve uses chord-length parameterization. It is clear that chord-length parameterization gives much better results. The interpolation points are the same as used for the interpolating Catmull-Rom and Overhauser splines shown in figures VII.23 and VII.24 on pages 190 and 192.

Chapter IX

Ray Tracing

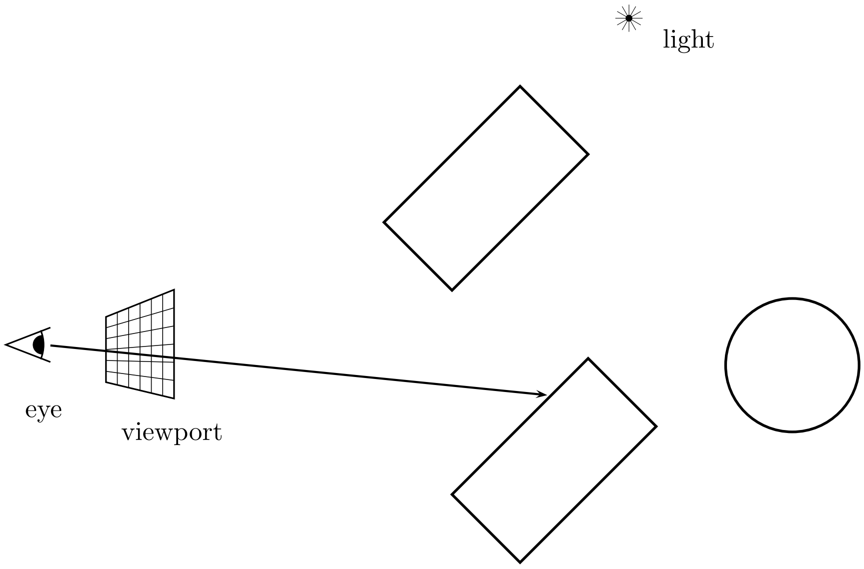


Figure IX.1: The simplest kind of ray tracing, non-recursive ray tracing, involves casting rays of light from the view position through pixel positions. A local lighting model is used to calculate the illumination of the surface intersected by the ray.

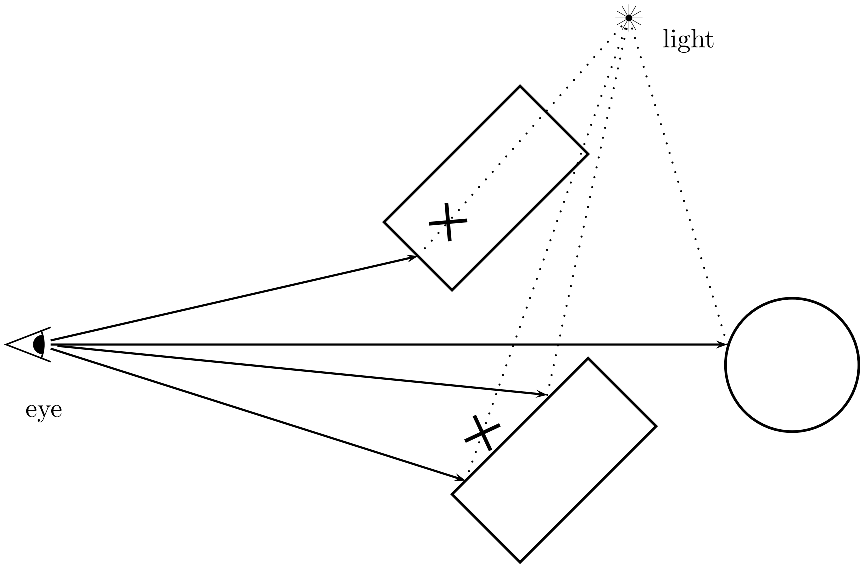


Figure IX.2: Shadow feelers: Rays from the eye are traced to their intersections with objects in the scene. Shadow feeler rays, shown as dotted lines, are sent from the points in the scene to each light in order to determine whether the point is directly illuminated by the point light source or whether it is in a shadow. The two shadow feelers marked with an "**X**" show that the light is not directly visible from the point.

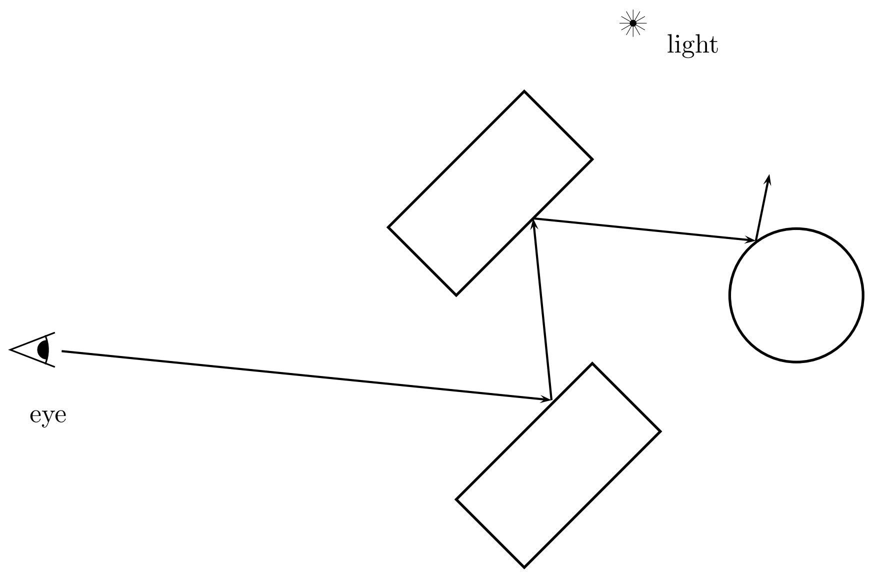


Figure IX.3: Reflection rays: The path of the ray from the eye is traced through multiple reflections. This calculates approximations to the lighting effects of multiple reflections.

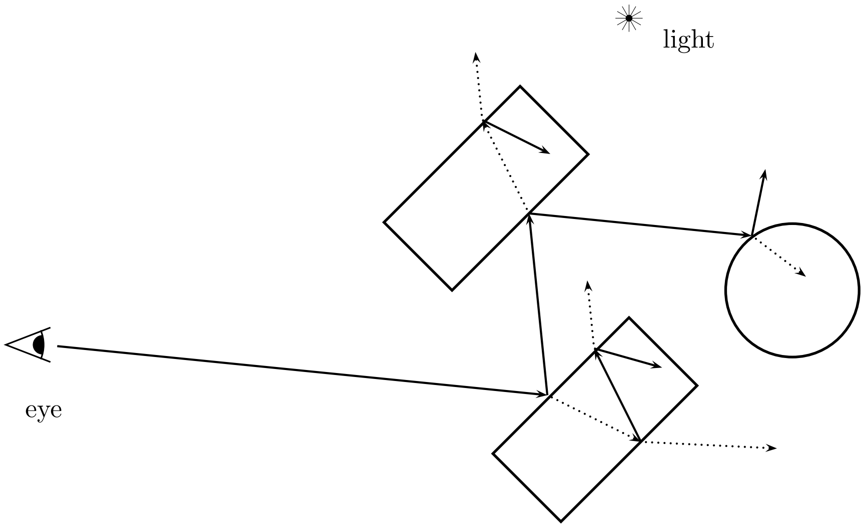


Figure IX.4: Transmission and reflection rays: The path of the ray from the eye is traced through multiple reflections and transmissions. Reflection rays are shown as solid lines, transmission rays as dotted lines. The shadow feeler rays would still be used, but are not shown.

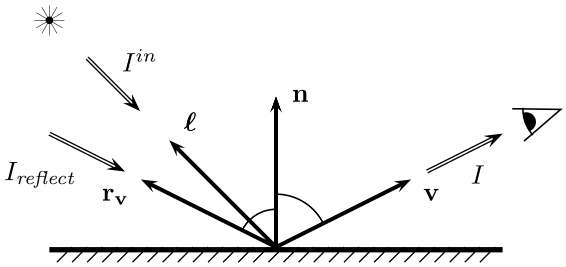


Figure IX.5: The usual setup for reflection rays in basic recursive ray tracing. The vector $\mathbf{v}$ points in the direction opposite to the incoming ray. The direction of perfect reflection is shown by the vector $\mathbf{r_v}$. The vector $\boldsymbol{\ell}$ points to a point light source. I is the outgoing light intensity as seen from the direction given by $\mathbf{v}$. $I_{reflect}$ is the incoming light from the reflection direction $\mathbf{r_v}$. I^{in} is the intensity of the light from the light source. (Compare this to figure III.7 on page 72.)

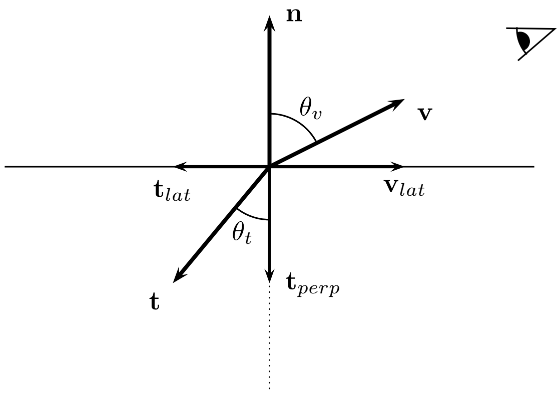


Figure IX.6: Computing the transmission ray direction $\mathbf{t}$. The horizontal line represents the surface of a transmissive material; $\mathbf{n}$ is the unit vector normal to the surface. The vector $\mathbf{v}$ points in the direction opposite to the incoming ray. The direction of perfect transmission is shown by the vector $\mathbf{t}$. The vectors $\mathbf{v}_{lat}$ and $\mathbf{t}_{lat}$ are the projections of these vectors onto the plane tangent to the surface. And, $\mathbf{t}_{perp}$ is the projection of $\mathbf{t}$ onto the normal vector.

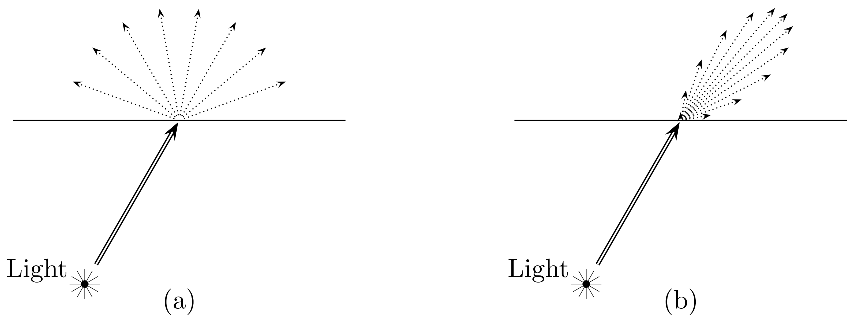


Figure IX.7: (a) Diffusely transmitted light. (b) Specularly transmitted light. The specularly transmitted light is centered around the transmission direction from Snell's law.

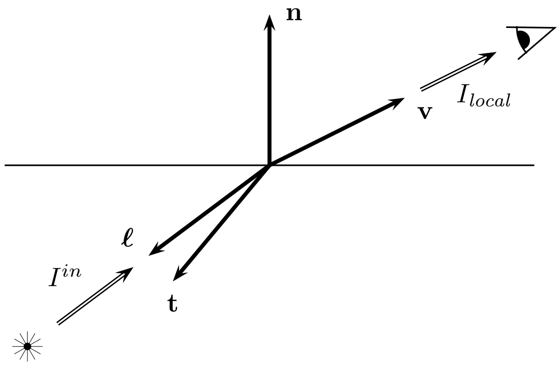
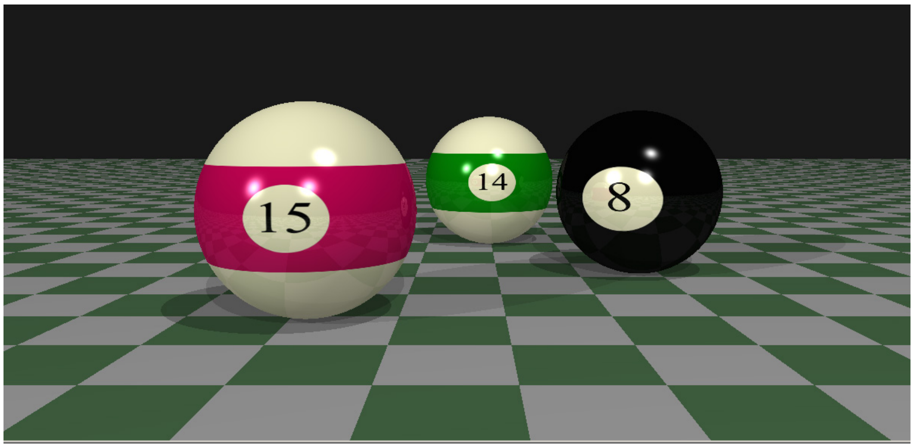
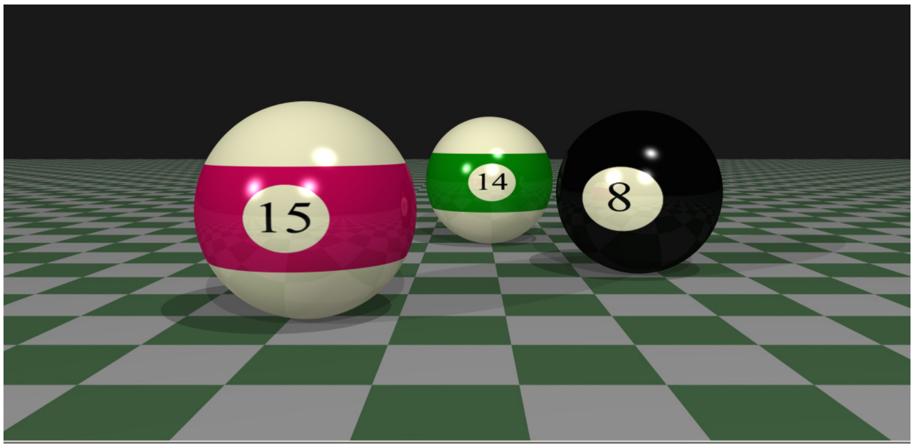


Figure IX.8: The vectors used in the computation of transmitted light are $\mathbf{v}$, $\boldsymbol{\ell}$, $\mathbf{t}$, and $\mathbf{n}$. The vector $\mathbf{v}$ points in the direction opposite to the incoming ray. The direction of perfect transmission is shown by the vector $\mathbf{t}$. The direction opposite to the incoming light is given by $\boldsymbol{\ell}$.

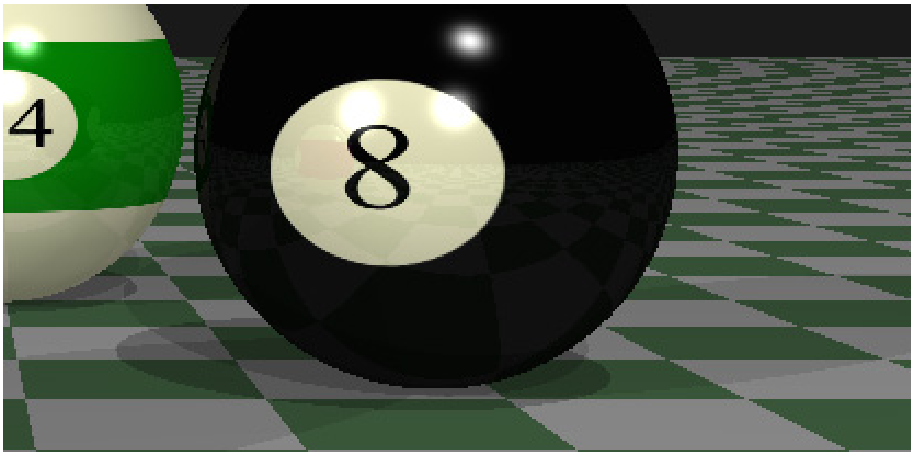


(a) No supersampling.

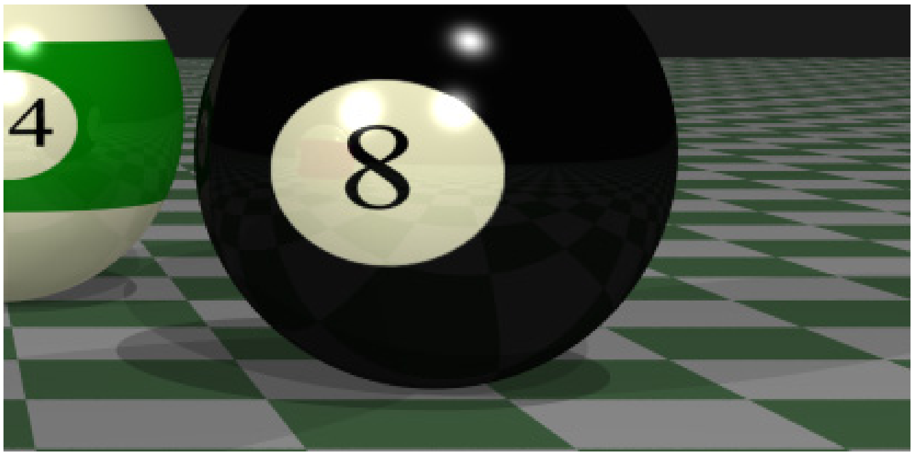


(b) Supersampling with jittered subpixel centers.

Figure IX.9: An example of anti-aliasing using jittered subpixel centers. (a) shows the scene rendered without supersampling; note the "jaggies" on the silhouettes of the balls, for instance. (b) is the scene with pixels selectively supersampled up to a maximum of 40 times. See color plate C.9.



(a) No supersampling.



(b) Supersampling with jittered subpixel centers.

Figure IX.10: Close up views of the images in figure IX.9. See color plate C.10.

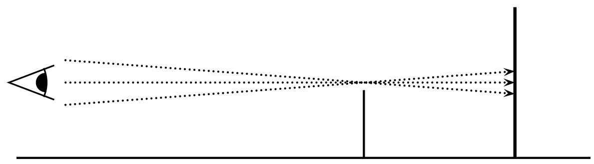


Figure IX.11: The rays from the jittered viewpoints converge at the focal plane, but not at the back plane.

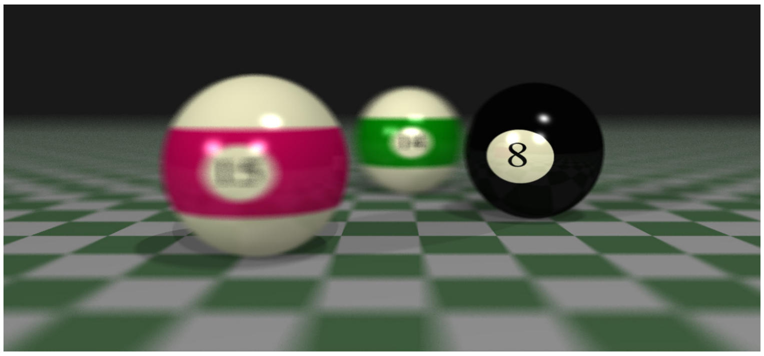


Figure IX.12: An example of depth of field. The front of the eight ball is the focal plane. Note also the blurring of the checkerboard plane. In this image, each pixel is selectively supersampled up to 40 times. The eye positions and the subpixel positions were independently jittered as described on page 315. See color plate C.11.

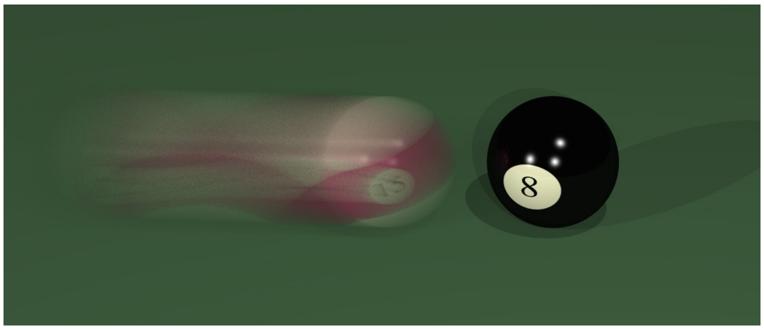


Figure IX.13: An example of motion blur. Pixels were selectively supersampled up to 40 times. Both motion supersampling and subpixel supersampling were used. See color plate C.12.

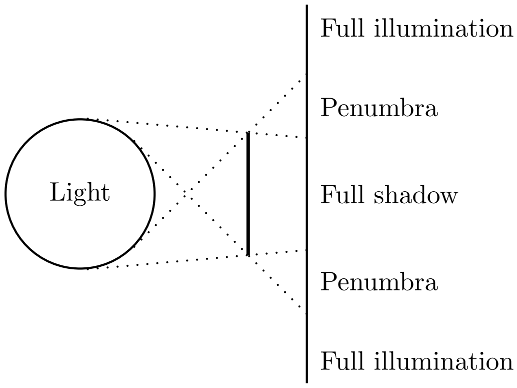


Figure IX.14: The penumbra is the area where the light is only partly blocked.

Chapter X

Intersection Testing

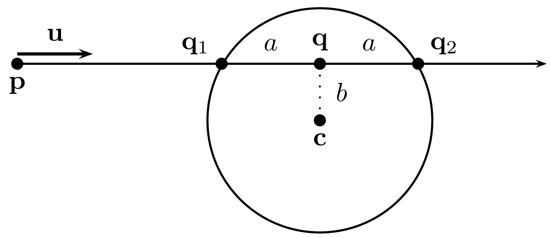


Figure X.1: The ray from $\mathbf{p}$ in direction $\mathbf{u}$ reaches its closest point to the center $\mathbf{c}$ of the sphere at $\mathbf{q}$. The points $\mathbf{q}_1$ and $\mathbf{q}_2$ are the two places where the line intersects the sphere.

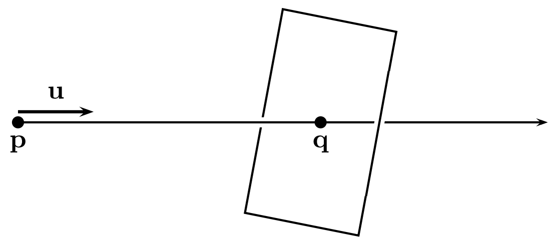


Figure X.2: The ray specified by its start position $\mathbf{p}$ and direction $\mathbf{u}$ intersects the plane at the point $\mathbf{q}$.

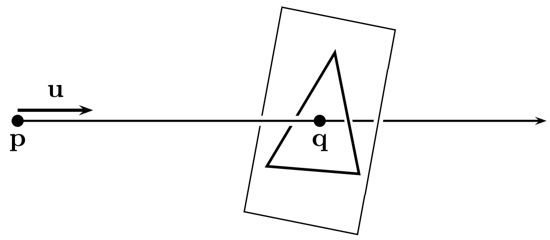


Figure X.3: Intersecting a ray with a triangle. The first step is to find the intersection **q** with the plane containing the triangle.

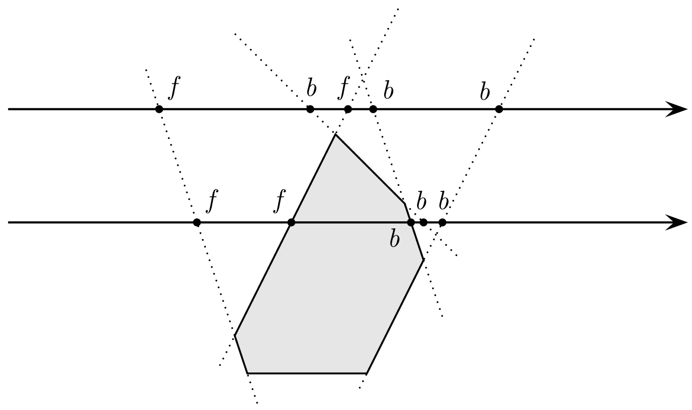


Figure X.4: The two rays are shown intersecting the bounding planes of the shaded polytope. Front intersections are labeled "f" and back intersections are labeled "b". The upper ray hits some back intersections before some front intersections and thus does not intersect the polytope. The lower ray hits all the front intersections before all the back intersections and thus does intersect the polytope.

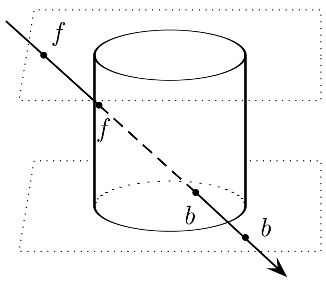


Figure X.5: A cylinder is the intersection of an infinite cylinder and the area between two parallel planes. The ray is shown hitting the top plane, then entering the (infinite) cylinder, then exiting the cylinder, and finally hitting the bottom plane. All the front intersections come before all the back intersections, so the ray does intersect the cylinder.

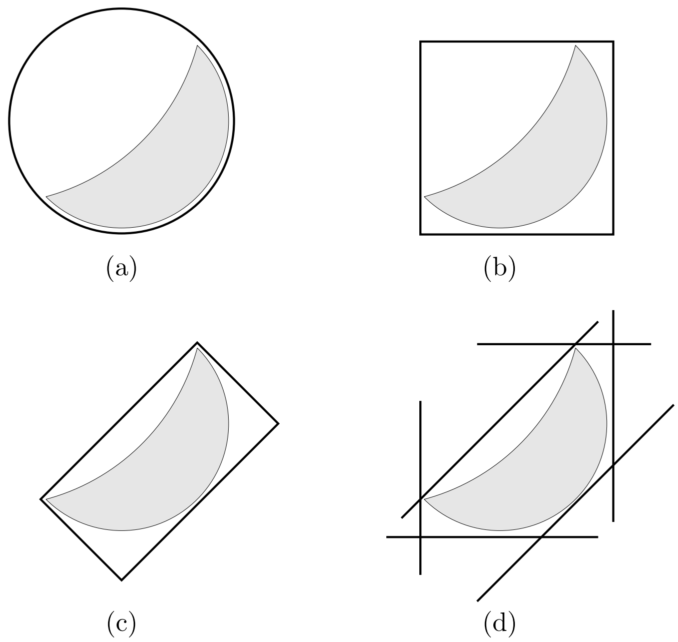


Figure X.6: Various kinds of two dimensional bounding volumes enclosing a crescent shape. (a) is a bounding sphere. (b) is an axis aligned bounded box (AABB). (c) is an oriented bounding box (OBB). (d) is a discrete oriented polygon (k-DOP) with $k = 3$.

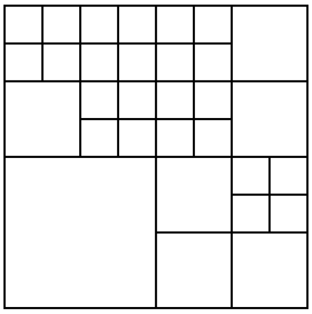


Figure X.7: A quadtree.

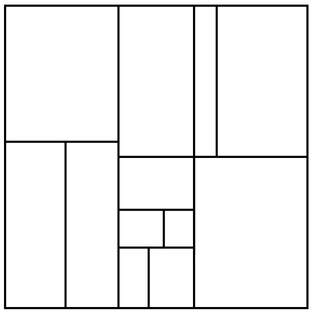


Figure X.8: A k-d tree in two dimensions. A region in a k-d tree can be subdivided by either a vertical or a horizontal line.

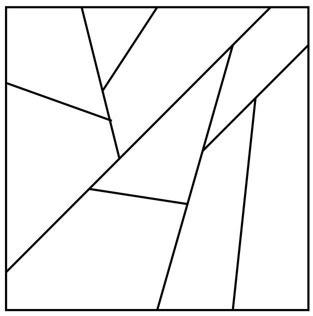


Figure X.9: A BSP tree in two dimensions.

Chapter XI

Radiosity

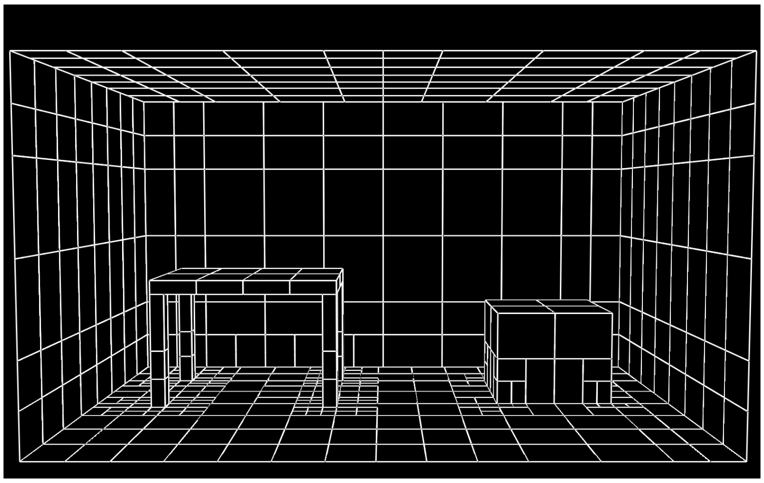


Figure XI.1: The patches used to render the radiosity scene of figures XI.2 and XI.3. See color plate C.13.

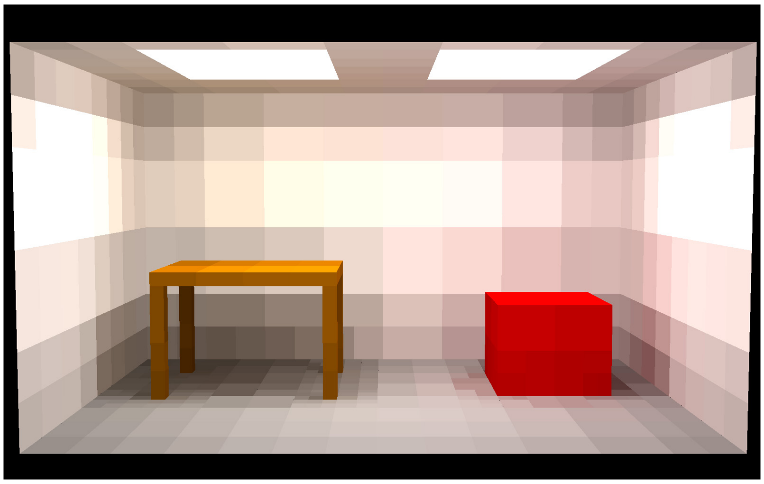


Figure XI.2: A radiosity rendered figure, with flat shading. It is evident that this image is based on the patches shown in figure XI.1. See color plate C.14.

Figure XI.3: A radiosity rendered figure, with smooth shading of illumination. The red color of the box is reflected onto the nearby walls, giving them a slight reddish hue. This is based on the patches shown in figure XI.1. See color plate C.15.

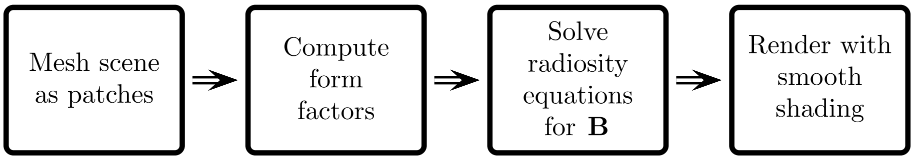


Figure XI.4: The four stages of the radiosity algorithm

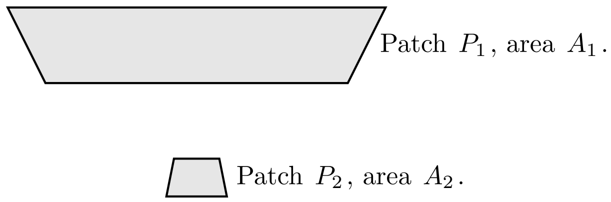


Figure XI.5: A large patch and small patch.

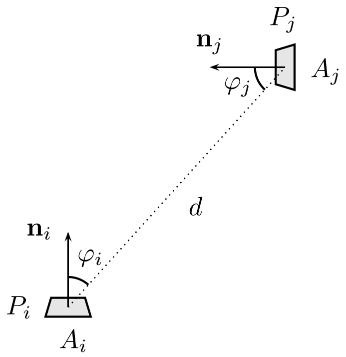


Figure XI.6: Two infinitesimally small patches P_i and P_j with areas A_i and A_j. The normals to the patches are $\mathbf{n}_i$ and $\mathbf{n}_j$.

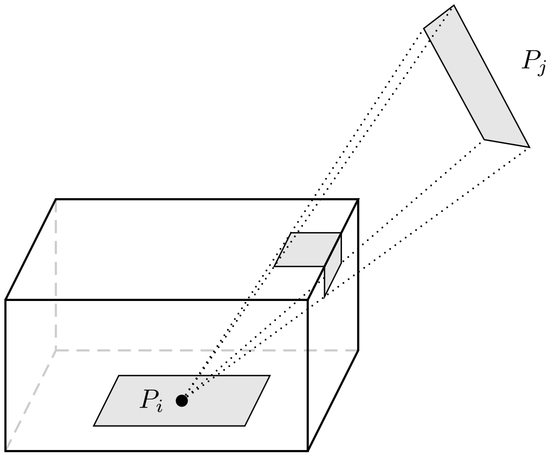


Figure XI.7: Projection onto a hemicube.

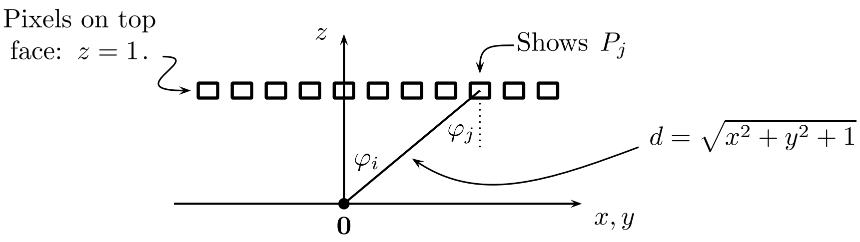


Figure XI.8: A row of pixels along the top of the hemicube. One pixel shows patch P_j. The origin is placed at the center of patch P_i. The top of the cube is the $z = 1$ plane.

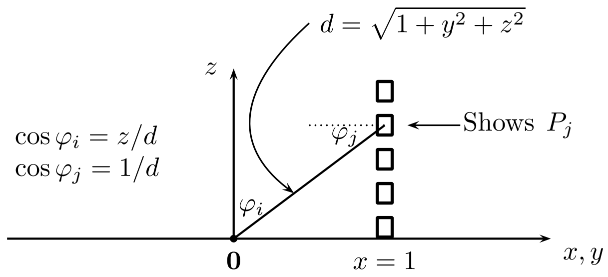


Figure XI.9: A row of pixels along the $x = 1$ side of the hemicube.

Chapter XII

Animation and Kinematics

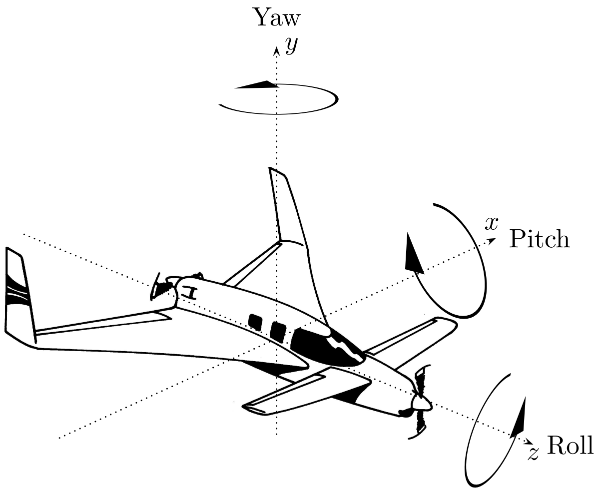


Figure XII.1: Yaw, pitch, and roll represent rotations around the y-axis, the x-axis and the z-axis. If the axes move with the object, then the rotations are performed in the order yaw, then pitch, and finally roll. If the axes are taken as fixed, then the rotations are performed in the opposite order: roll, then pitch, then yaw. Rotation directions are determined by the righthand rule. The reader is warned that the rotation directions for pitch and yaw that are shown in the figure are opposite to customary usage in aviation. For us, a positive pitch means the nose dips down and a positive yaw steers to the left. However, aviation conventions are that a positive pitch means the nose moves up, and a positive yaw means turning to the right. It is customary for positive roll to mean that the right wing dips, which agrees with the our convention. In aviation conventions, the directions of the x and y axes are reversed, with the x-axis pointing rightward and the y-axis pointing downward.

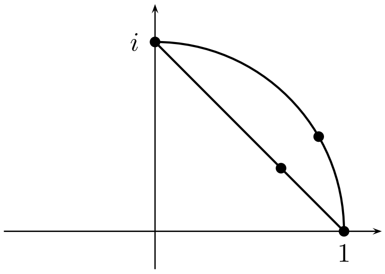
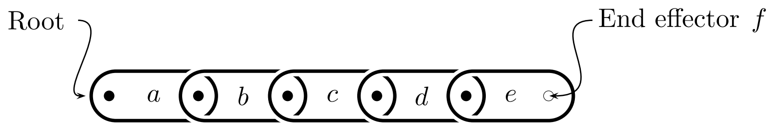
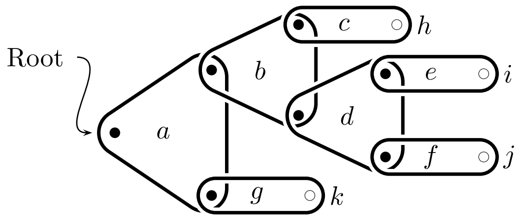


Figure XII.2: Lerping moves from 1 to i at a constant rate along the secant line. Slerping moves from 1 to i at a constant rate along the great circle. The points drawn on the secant line and on the great circle are obtained by lerping and slerping with $\alpha = \frac{1}{3}$. They do not correspond to the same rotation.



(a) A linear chain of links with 1-DOF rotational joints.



(b) A tree-like arrangement of links and 1-DOF joints. The points h-k are end effector points.

Figure XII.3: Two examples of tree-like arrangements of links. The black dots represent 1-DOF rotational joints. The small circles are end effector points.

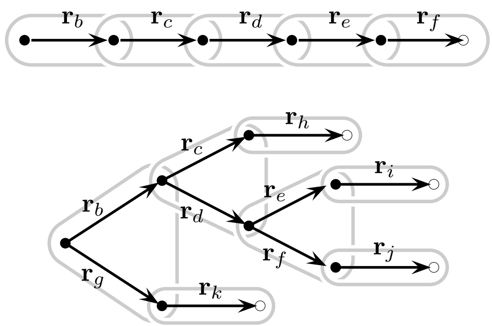


Figure XII.4: The relative position vectors $\mathbf{r}_x$ measure the relative rest positions of the links. $\mathbf{r}_a = \mathbf{0}$ and the other relative position vectors are as shown.

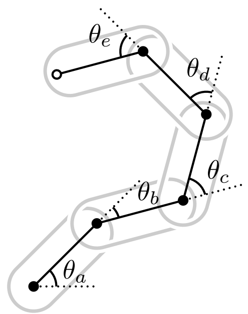


Figure XII.5: The angles of joints are measured relative to the two links it joins.

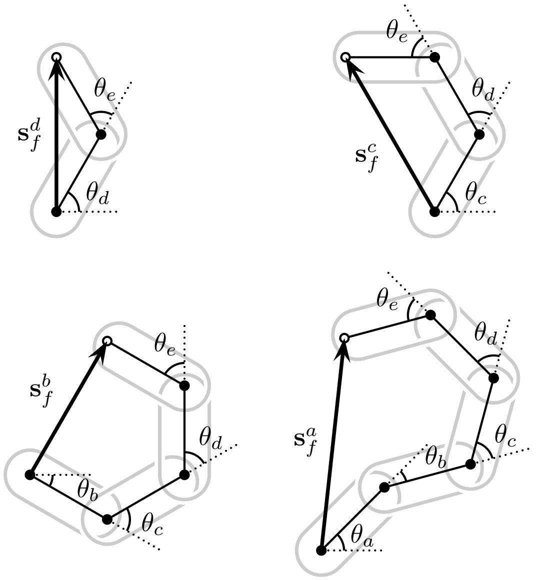


Figure XII.6: The definitions of $\mathbf{s}_f^y$ for the end effector f and the links $y = a, b, c, d$ of the linear chain shown in figure XII.3.

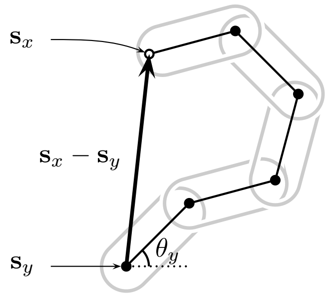


Figure XII.7: The rate of change of the position $\mathbf{s}_x$ of point x with respect to the joint angle θ_y is calculated in terms of rotation around the axis $\mathbf{w}_y$ of the joint y as $\mathbf{w}_y \times (\mathbf{s}_x - \mathbf{s}_y)$. The axis $\mathbf{w}_y$ is pointing out of the page, so $\mathbf{w}_y \times (\mathbf{s}_x - \mathbf{s}_y)$ has the right direction for the partial derivative. Since $\mathbf{w}_y$ is a unit vector, $\mathbf{w}_y \times (\mathbf{s}_x - \mathbf{s}_y)$ also has the right magnitude. This calculation works for any x and y, not for just end effectors and the root link.

Appendix A

Mathematics Background

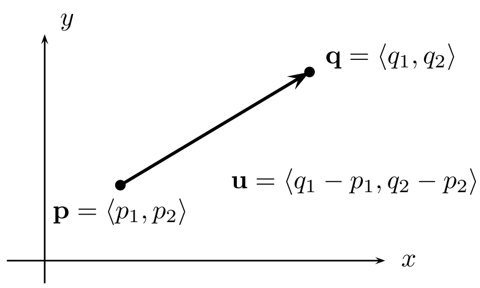


Figure A.1: Two points $\mathbf{p}$ and $\mathbf{q}$ and the vector $\mathbf{u} = \mathbf{q} - \mathbf{p}$.

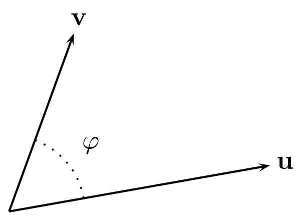


Figure A.2: The angle between $\mathbf{u}$ and $\mathbf{v}$ equals φ, and is used for computing the dot product. Since $\cos\varphi = \cos(-\varphi) = \cos(360^\circ - \varphi)$ it makes no difference which way the angle is measured for the purposes of computing the dot product.

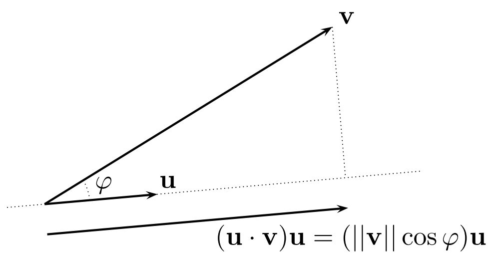


Figure A.3: The projection of $\mathbf{v}$ onto the line parallel to a unit vector $\mathbf{u}$. φ is the angle between $\mathbf{u}$ and $\mathbf{v}$. How would the picture look if φ were between $90°$ and $180°$?

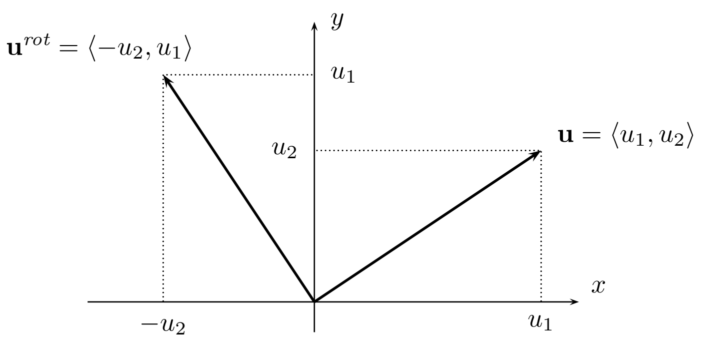


Figure A.4: The effect of rotating a vector 90 degrees counter-clockwise.

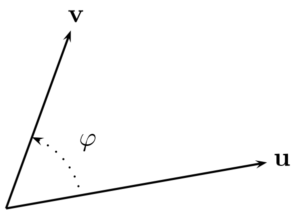


Figure A.5: The angle between $\mathbf{u}$ and $\mathbf{v}$ equals φ, and is used for computing the cross product. The direction (sign) of the angle is important for the cross product.

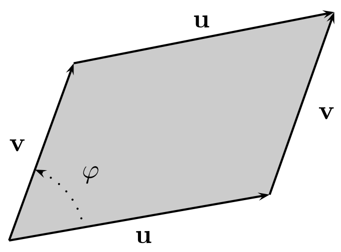


Figure A.6: In $\mathbb{R}^2$, the (signed) area of the parallelogram is equal to the cross product $\mathbf{u} \times \mathbf{v}$. In $\mathbb{R}^3$, the area of the parallelogram is equal to the magnitude of $\mathbf{u} \times \mathbf{v}$.

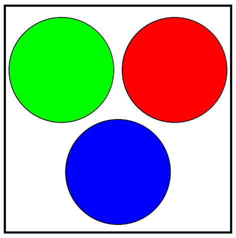
Appendix B

Ray Trace Software Package

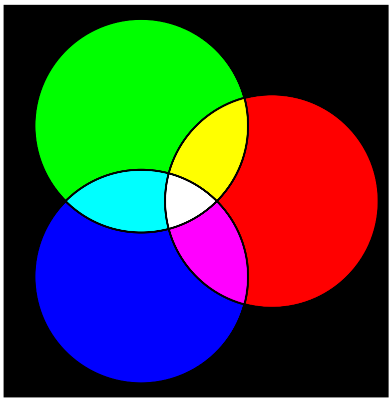
No figures in this appendix.

Appendix C

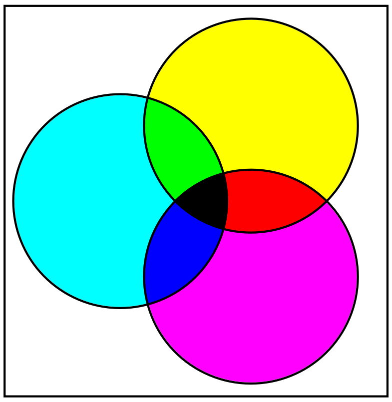
Color Plates



Color Plate C.1: (Figure I.1.) A pixel is formed from subregions or subpixels, each of which displays one of three colors.

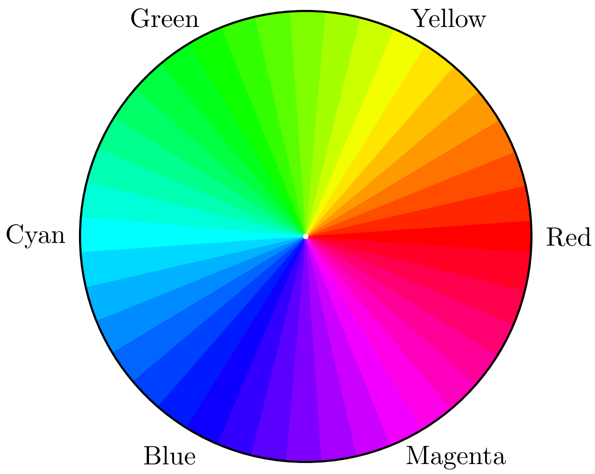


(a)

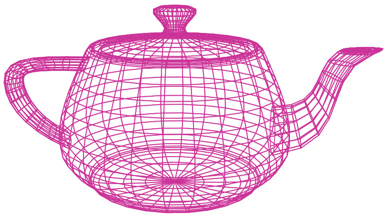


(b)

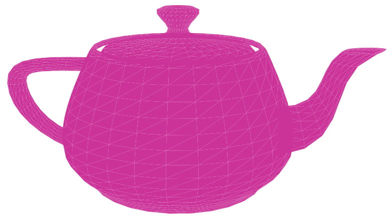
Color Plate C.2: (Figure VI.1.) (a) The additive colors are red, green, and blue. (b) The subtractive colors are cyan, magenta, and yellow.



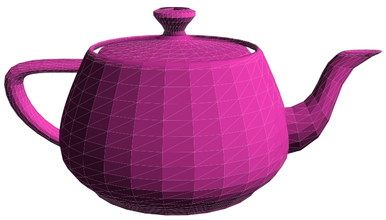
Color Plate C.3: (Figure VI.2.) Hue is measured in degrees, representing an angle around the color wheel. Pure red has hue equal to 0, pure green has hue equal to 120, and pure blue has hue equal to 240.



(a)

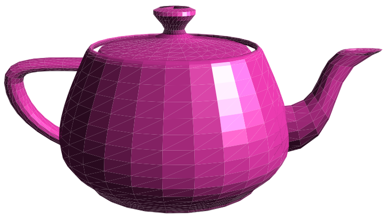


(b)



(c)

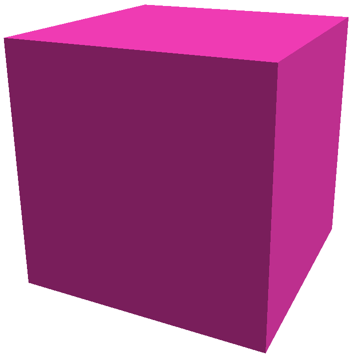
(d)



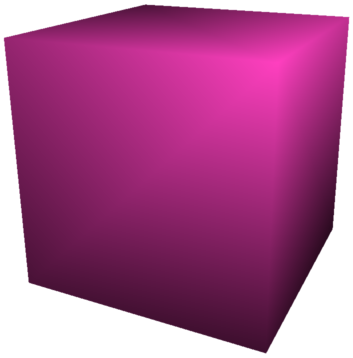
(e)

(f)

Color Plate C.4: (Figure III.1.) Six teapots with various shading and lighting options. (a) Wireframe teapot. (b) Teapot drawn with solid color, but no lighting or shading. (c) Teapot with flat shading, with only ambient and diffuse lighting. (d) Teapot drawn with Gouraud interpolation, with only ambient and diffuse reflection. (e) Teapot drawn with flat shading, with ambient, diffuse, and specular lighting. (f) Teapot with Gouraud shading, with ambient, diffuse, and specular lighting.



(a)

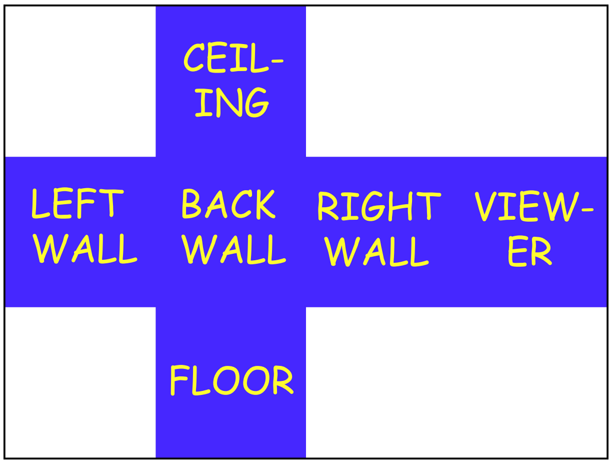


(b)

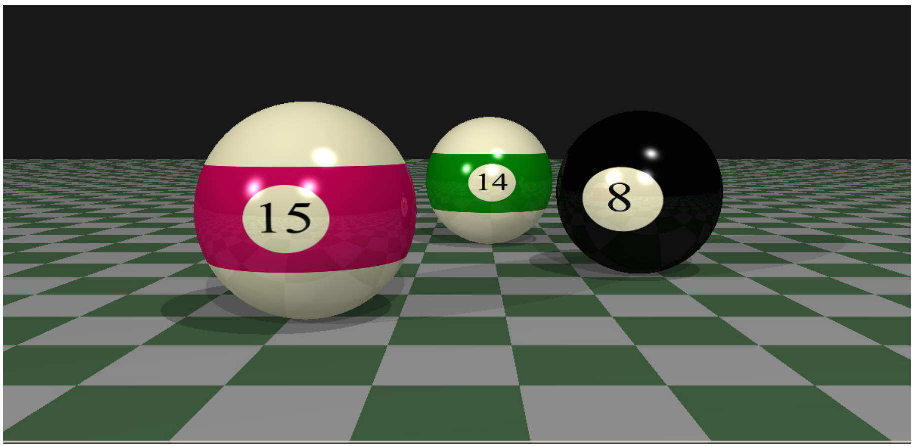
Color Plate C.5: (Figure III.9.) Two cubes with (a) normals at vertices perpendicular to each face, and (b) normals outward from the center of the cube. Note that (a) is rendered with Gouraud shading, not flat shading.

Color Plate C.6: (Figure V.8.) A bump mapped torus. Note the lack of bumps on the silhouette. There are four white lights shining on the scene, plus a low level of ambient illumination. This picture was generated with the ray tracing software described in appendix B.

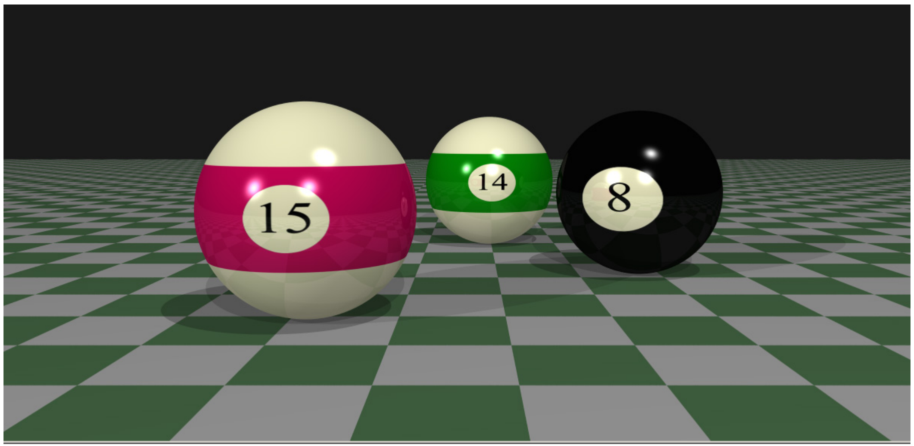
Color Plate C.7: (Figure V.10.) An environment map mapped into a sphere projection. This is the kind of environment map supported by OpenGL. The scene is the same as is shown in figure V.11. Note that the front wall has the most fidelity, and the back wall the least. For this reason, spherical environment maps are best used when the view direction is close to the direction used to create the environment map.



Color Plate C.8: (Figure V.11.) An environment map mapped into a box projection consists of the six views from a point, mapped to the faces of a cube and then unfolded to make a flat image. This scene shows the reflection map from the point at the center of a room. The room is solid blue, except for yellow writing on the walls, ceiling and floor. The rectangular white regions of the environment map are not used.

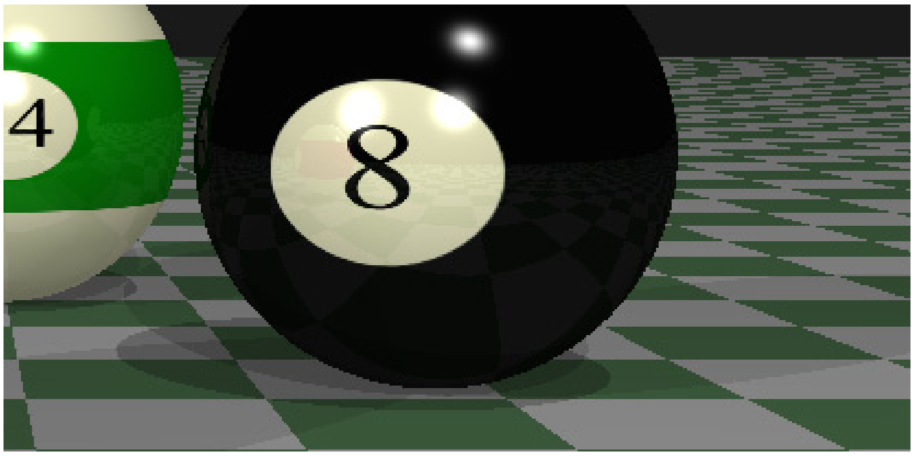


(a) No supersampling.

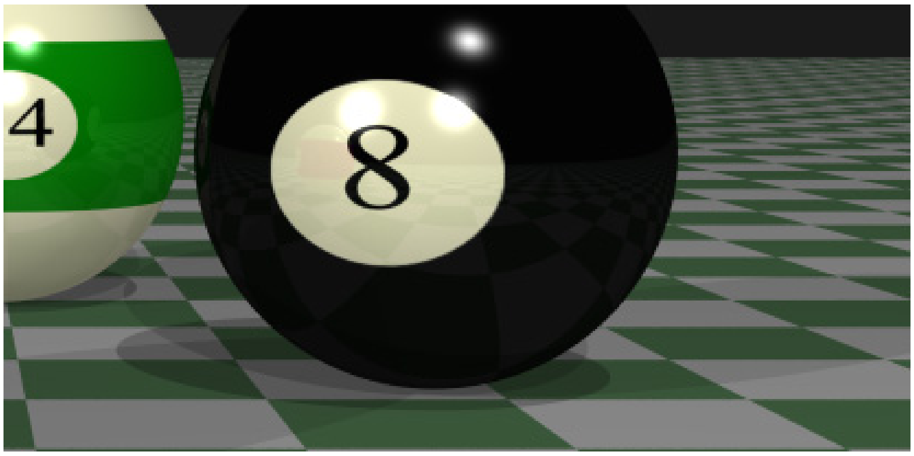


(b) Supersampling with jittered subpixel centers.

Color Plate C.9: (Figure IX.9.) An example of anti-aliasing using jittered subpixel centers. (a) shows the scene rendered without supersampling; note the "jaggies" on the silhouettes of the balls, for instance. (b) is the scene with pixels selectively supersampled up to a maximum of 40 times.

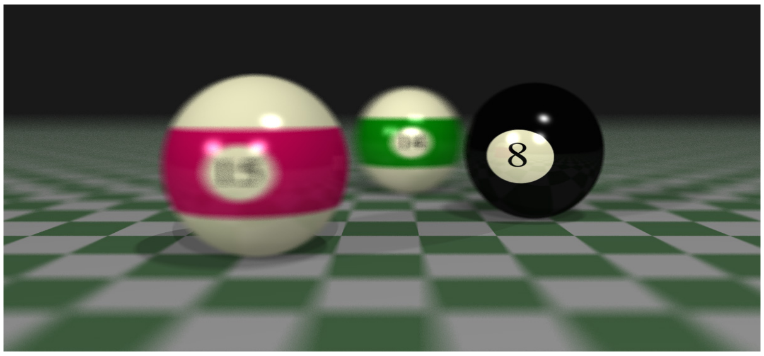


(a) No supersampling.

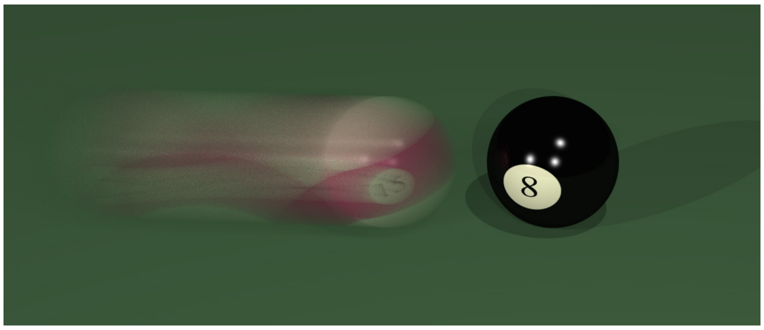


(b) Supersampling with jittered subpixel centers.

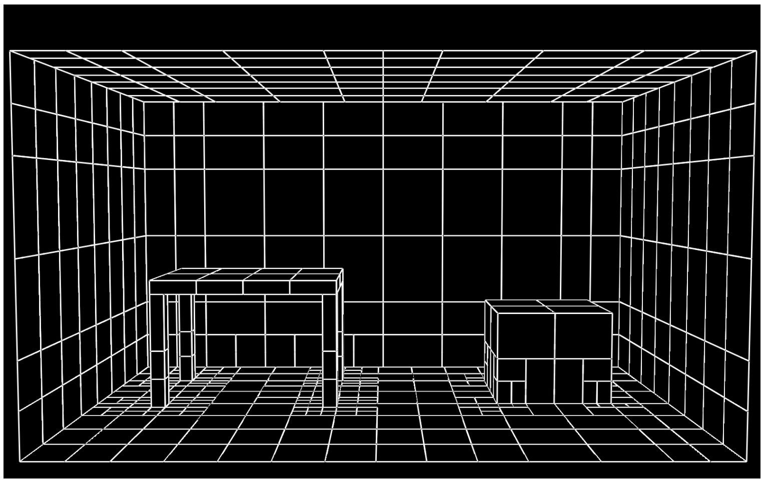
Color Plate C.10: (Figure IX.10.) Close up views of the images in figure IX.9.



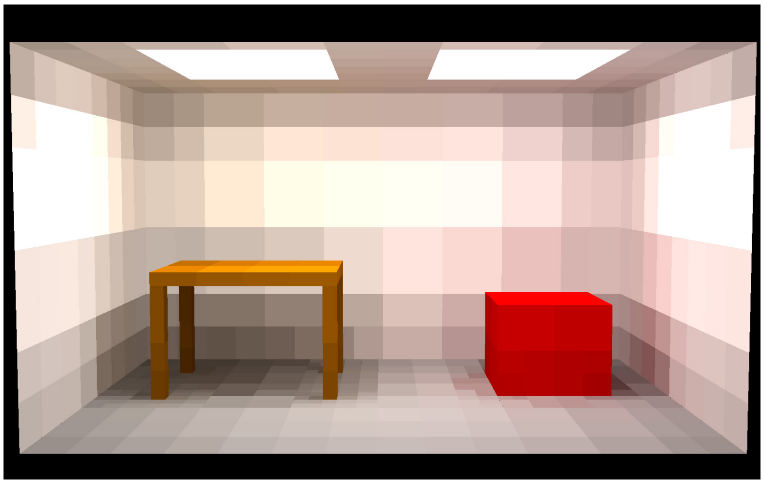
Color Plate C.11: (Figure IX.12.) An example of depth of field. The front of the eight ball is the focal plane. Note also the blurring of the checkerboard plane. In this image, each pixel is selectively supersampled up to 40 times. The eye positions and the subpixel positions were independently jittered as described on page 249.



Color Plate C.12: (Figure IX.13.) An example of motion blur. Pixels were selectively supersampled up to 40 times. Both motion supersampling and subpixel supersampling were used.

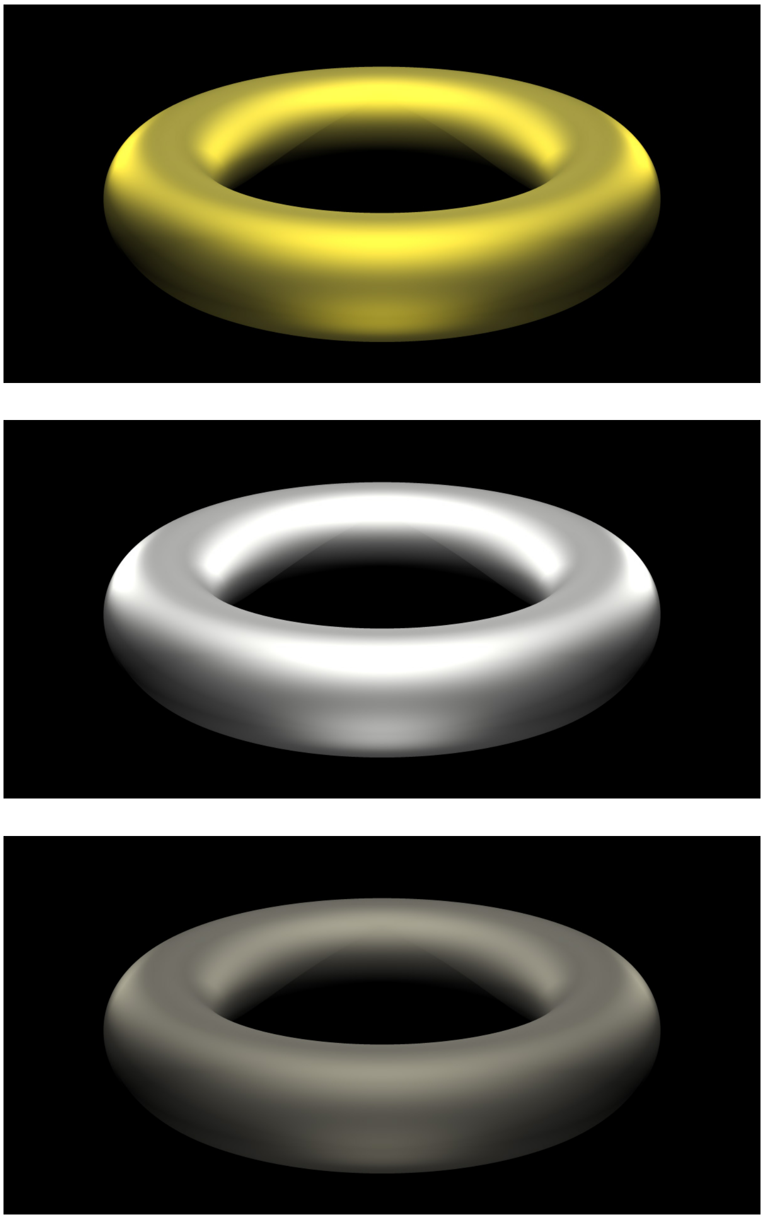


Color Plate C.13: (Figure XI.1.) The patches used to render the radiosity scene of figures XI.2 and XI.3.



Color Plate C.14: (Figure XI.2.) A radiosity rendered figure, with flat shading. It is evident that this image is based on the patches shown in figure XI.1.

Color Plate C.15: (Figure XI.3.) A radiosity rendered figure, with smooth shading of illumination. The red color of the box is reflected onto the nearby walls, giving them a slight reddish hue. This is based on the patches shown in figure XI.1.



Color Plate C.16: (Figure III.20.) Metallic tori, with specular component computed using the Cook-Torrance model. The materials are, from top to bottom, gold, silver and platinum. The roughness is $m = 0.4$ for all three materials. The tori are each illuminated by five positional white lights.