CSC418 Assignment #2

Filip Tomin
1001329984

1. **Transformations**

   (a) To find the transformation matrix T that maps the triangle $a = (a_1, a_2, a_3)$ to the triangle $b = (b_1, b_2, b_3)$, we can simply change the transformation equation to solve for T.

   The transformation equation is $b = Ta$, which can be rearranged to $b/a = T$ or $T = ba^{-1}$. To be more precise, a and b are 3x3 matrices of their points, with each z value equal to 1. So,

   $$T = \begin{bmatrix} b_{1x} & b_{2x} & b_{3x} \\ b_{1y} & b_{2y} & b_{3y} \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} a_{1x} & a_{2x} & a_{3x} \\ a_{1y} & a_{2y} & a_{3y} \\ 1 & 1 & 1 \end{bmatrix}^{-1}$$

   For a transformation matrix to be fully determined, there must be only one possible transformation matrix which maps a to b. In general, the points need to be unique, as in if $a_1 = a_2 = a_3$ and $b_1 = b_2 = b_3$ then there are infinitely many scalar + transform combinations which could map one point to the other. The points also must not be collinear as this also induces the possibility for multiple transform matrices. For example, a vertical line at $x = 4$ can be shifted left 2 or scaled by a half to get the same line, (or even shifted left 3 then scaled by 2). In general, the points should form triangles with an area greater than 0.

   (b) For a 2D similarity transform, there are 6 D.O.F so we need 3 point pairs or 6 points in order to compute. A 2D homography is 8 D.O.F so we would need 4 point pairs or 8 points in order to compute.

   (c) The centroid of a triangle is affine invariant:

   An affine transformation is of the form $b = Sa + T$, where point a is mapped onto point b by a scalar vector S and a transformation vector T. Then centroid c could be defined as $c = (\frac{x_1 + x_2 + x_3}{3}, \frac{y_1 + y_2 + y_3}{3})$ where the 3 points of the triangle are declared by (x_i, y_i) for the ith point. Applying the affine transformation, each point $x_i' = S_x x_i + T_x$. Then:

   $c' = (\frac{x_1' + x_2' + x_3'}{3}, \frac{y_1' + y_2' + y_3'}{3})$
   $= (\frac{S_x x_1 + T_x + S_x x_2 + T_x + S_x x_3 + T_x}{3}, \frac{S_y y_1 + T_y + S_y y_2 + T_y + S_y y_3 + T_y}{3})$
   $= (S_x(\frac{x_1 + x_2 + x_3}{3}) + T_x, S_y(\frac{y_1 + y_2 + y_3}{3}) + T_y)$
   $= Sc + T$

Thus, the centroid of the transformed triangle is just the transformed centroid.
The circumcenter of a triangle is not affine invariant:
We can prove this with a counter-example. Let's define the triangle a formed
by (a_1, a_2, a_3), where $a_1 = (0,0), a_2 = (4,0), a_3 = (2,2)$. For a, the circumcenter
$c = (2,0)$.
Now, let's apply a simple affine tranformation that only scales the y value. So,
$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Applying T to our triangle, only a_3 is changed since the other
y values are 0. $a_1' = a_1, a_2' = a_2, a_3' = (2,4)$. With these new points, the new
circumcenter $c' = (2, 1.5)$. Applying T to c would have gotten us c again, since it
has a y value that is 0, but the $c' \neq c$. Thus, by counter-example, the circumcenter
is not affine invariant.

2. **Viewing and Projection**

   (a) Knowing light travels in a straight line, if we position an object in front of the
   pinhole camera, light from the top of the object goes diagonally through the
   pinhole to the bottom on the other side and vice versa. The light ray from the
   top of the object reaches the bottom of the pinhole and the light ray from the
   bottom of the object reaches the top of the pinhole, thus creating an inverted
   image.

   (b) Since the line $\vec{u}$ is parallel to the vertical line of the screen, we can consider this
   as a simple 2D perspective transform. If p is in the center of the screen plane,
   then $p = (0, 0, d)$. Let any point on $\vec{u}$ be defined as (x, y, z) and the transformed
   point on $\vec{u}'$ as (x', y', z').
   Since this is a simple 2D perspective transform, $z' = d$ and the other values are
   multiplied by the ratio d/z. Thus, $x' = x\frac{d}{z}$ and $y' = y\frac{d}{z}$.

   The transformation matrix would then be $T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix}$.

   (c) As long as the family of parallel lines is also parallel to either the vertical or
   horizontal line of the screen, they will remain parallel.

   (d) When the condition is not met, the family of parallel lines converge towards the
   vanishing point. You can create another member of this family of parallel lines
   which shares their slope m but begins at the camera c. This line will eventually
   intersect with the screen plane at some point (x, y). This POI is where every line
   in the family will converge to.

3. **Surfaces**

(a) The surface normal would be the gradient of the function:
$$\nabla f(p = (x, y, z)) = (\tfrac{\delta f}{\delta x}, \tfrac{\delta f}{\delta y}, \tfrac{\delta f}{\delta z}) = (\frac{-2x(R-\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}, \frac{-2y(R-\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}, 2z)$$

(b) The implicit function for a plane is $(p - p_0) \cdot \vec{n} = 0$, where p_0 is a point on the plane. Since we're given the point p, let u represent p in the plane function and $p_0 = p$. We have $\vec{n}$ from the gradient function, so:
Tangent Plane $P = (u - p) \cdot \nabla f(p) = 0$.

(c) We can plug in $q(\lambda)$ into the implicit surface function to determine whether the curve lies on the surface. So:
$$
\begin{aligned}
f(Rcos\lambda, Rsin\lambda, r) \quad &= (R - \sqrt{R^2cos^2\lambda + R^2sin^2\lambda})^2 + r^2 - r^2 \quad &= 0 \\
&= (R - \sqrt{R^2(cos^2\lambda + sin^2\lambda)})^2 \quad &= 0 \\
&= R - \sqrt{R^2} \quad &= 0 \\
&= R - R \quad &= 0
\end{aligned}
$$
Thus $q(\lambda)$ lies on the surface.

(d) The tangent vector of $q(\lambda)$ is simply the derivative with respect to λ, so $q'(\lambda) = (-Rsin\lambda, Rcos\lambda, 0)$.

(e) We can show the tangent vector lies on the implicit equation by using $q'(\lambda)$ as $(u - p)$, finding the dot product and plugging in $q(\lambda)$. So:
$q'(\lambda) \cdot \nabla f = 0$
$$\frac{2x(R-\sqrt{x^2+y^2})(Rsin\lambda)}{\sqrt{x^2+y^2}} - \frac{2y(R-\sqrt{x^2+y^2})(Rcos\lambda)}{\sqrt{x^2+y^2}} + 0 = 0, \text{ re-arranged to:}$$
$$\frac{2x(R-\sqrt{x^2+y^2})(Rsin\lambda)}{\sqrt{x^2+y^2}} = \frac{2y(R-\sqrt{x^2+y^2})(Rcos\lambda)}{\sqrt{x^2+y^2}}, \text{ then we cancel anything on both sides:}$$
$xsin\lambda = ycos\lambda$, now substituting from $q(\lambda)$:
$Rcos\lambda sin\lambda = Rsin\lambda cos\lambda$. Thus,
$q'(\lambda) \cdot \nabla f = 0$ and the tangent vector lies on the tangent plane.

4. **Curves**

(a) Using the bezier quadratic equation:
$B_1(t) = (1 - t)^3 p_1 + 3t(1 - t)^2 p_2 + 3t^2(1 - t)p_3 + t^3 p_4$, then
$B_1'(t) = -3(1 - t)^2 p_1 + (9t^2 - 12t + 3)p_2 + (6t - 9t^2)p_3 + 3t^2 p_4$, at $t = 1$:
$B_1'(1) = 0 + 0 - 3p_3 + 3p_4 = -3p_3 + 3p_4$.
Similarly for B_2:
$B_2'(t - 1) = (-3t^2 + 12t - 12)p_4 + (9t^2 - 30t + 24)p_5 + (-9t^2 + 24t - 15)p_6 + (3t^2 - 6t + 3)p_7$, plugging in $t = 1$:
$B_2'(1) = -3p_4 + 3p_5 + 0 + 0 = -3p_4 + 3p_5$.

$$B_1'(1) = -3p_3 + 3p_4$$
$$B_2'(1) = -3p_4 + 3p_5$$

(b) $B_1''(t) = (6 - 6t)p_1 + (18t - 12)p_2 + (6 - 18t)p_3 + 6tp_4$, at $t = 1$:
$B_1''(1) = 0 + 6p_2 - 12p_3 + 6p_4 = 6p_2 - 12p_3 + 6p_4$.

Similarly for B_2:
$B_2''(t-1) = (12-6t)p_4 + (18t-30)p_5 + (24-18t)p_6 + (6t-6)p_7$, at $t = 1$:
$B_2''(1) = 6p_4 - 12p_5 + 6p_6 + 0 = 6p_4 - 12p_5 + 6p_6$.

$$B_1''(1) = 6p_2 - 12p_3 + 6p_4$$
$$B_2''(1) = 6p_4 - 12p_5 + 6p_6$$

(c) For C^2 continuity, the second derivatives $B_1''(1)$ and $B_2''(1)$ must be equal. Then:
$6p_2 - 12p_3 + 6p_4 = 6p_4 - 12p_5 + 6p_6$
$6p_2 - 12p_3 = 6p_6 - 12p_5$
$p_6 - 2p_5 = p_2 - 2p_3$
The values of p_5 and p_6 are fixed such that the above equality must be satisfied. The value of p_7 is not fixed.

(d) 4 reasons why Bezier curves are dope af:
1. Cubic Bezier curves are easy to compute and can form any shape whether on its own or combined with more Bezier curves. Cubics hold the same precision as higher order curves but with the least amount of inputs.
2. While they can be used to draw curves in vector graphics, they can also be used in the domain of time to perfectly smooth animations or moving parts.
3. Since the curve is in the convex hull of its control points, the control points can be displayed on screen and easily manipulated.
4. Bezier curves are affine invariant, so transforming a Bezier curve can be done by transforming only the control points rather than all the points on the curve.