CSC418S/2504S: Computer Graphics
Department of Computer Science
University of Toronto

Eugene Fiume | Assignment 1a | Due: 11:59pm, 29 Jan. 2010

Plagiarism Policy. The work you hand in is expected to be a consequence of your intellectual effort. You are welcome to discuss issues in your assignment with others, but please do not copy their work. Similarly, feel free to consult references, but do not copy the results. If your work is discovered to be from another source, you won't like the consequences. Express your proofs/programs in your own words/code.

Your assignment will consist of a practical and a theoretical component. The practical component will be distributed separately and will account for about one-half of your grade on this assignment. Assignment 1a is the theoretical component; work on this first. There is a lot of notation in this assignment. Don't panic! Work through it slowly, draw pictures, rewrite the question, and try to understand what the point of the question is.

(1) A transformation $T : \mathbf{R}^n \to \mathbf{R}^n$ is *linear* if for vectors $\mathbf{x}, \mathbf{y} \in \mathbf{R}^2$ and scalar $a \in \mathbf{R}^2$, $T(a\mathbf{x} + \mathbf{y}) = aT(\mathbf{x}) + T(\mathbf{y})$. Consider any real line segment $\mathbf{L}(P_1, P_2)$ with $P_1, P_2 \in \mathbf{R}^2$ in the real plane (not only in the first octant). Show that for any linear transformation T,

$$T\Big(\mathbf{L}(P_1, P_2)\Big) = \mathbf{L}\Big(TP_1, TP_2\Big).$$

Hint: choosing the right representation for line segments makes this much easier! Note that points can be seen as position vectors, so applying a linear transformation to points is valid. In words, what does this theorem mean? Why is this important in practice? Everything in this question actually generalises to $\mathbf{R}^n$.

(2) Consider two parametric quadratic curves $\mathbf{C}_1(t)$ and $\mathbf{C}_2(t)$ in $\mathbf{R}^2$. We define the parametric curve $\mathbf{C}_1(t) = \Big(x_1(t), y_1(t)\Big)$, where both $x_1(t)$ and $y_1(t)$ are quadratic curves, and we will let t vary over the interval [0,1] for each curve. Likewise for $\mathbf{C}_2$. Although it's notationally awkward, define

$$\begin{aligned} x_1(t) &= a_1^x t^2 + b_1^x t + c_1^x, \\ y_1(t) &= a_1^y t^2 + b_1^y t + c_1^y, \end{aligned}$$

and likewise once again for $x_2(t)$ and $y_2(t)$ of $\mathbf{C}_2$. Now that you have a complete mess of subscripts and superscripts, suppose you wish to create a compound or *piecewise* curve $\mathbf{C}(t)$ consisting of $\mathbf{C}_1(t)$ traced out over $t \in [0, 1]$, followed by $\mathbf{C}_2(t)$, again for $t \in [0, 1]$. If we were to write a procedure to do this, we would get a piece of a quadratic space curve in one place on the screen corresponding to $\mathbf{C}_1$ and another piece appearing in some random location corresponding to $\mathbf{C}_2$.

Let's impose some constraints on these curves so that they in fact appear continuous. The overall curve $\mathbf{C}$ might, for example, represent a trajectory of a particle moving in space, or it could be the outline of a character to be rasterised. In this case, rather than jumping from $\mathbf{C}_1$ to $\mathbf{C}_2$, we would remove the gap between them, and thus the particle would instead travel continuously along the piecewise path given by $\mathbf{C}$. This can be done by imposing (boundary) conditions on $\mathbf{C}_1$ and $\mathbf{C}_2$ for appropriate values of t. In particular, we would first like $\mathbf{C}_1(1)$ to coincide with $\mathbf{C}_2(0)$. What constraints must be imposed on the coefficients a_2^x, a_2^y, etc., of x_2 and y_2 in terms of those in x_1 and y_1, respectively, to ensure that this boundary condition is met? Note that because all the components are quadratic, you will only need to work out the constraints for the coefficients in x_2 from x_1. The identical constraints will apply to the case for y_2 from y_1.

Now suppose that in addition to maintaining positional continuity, we also want their first derivatives to match. In other words, now we wish the instantaneous velocity (and thus the kinematics) of the particle not to change abruptly when going from $\mathbf{C}_1$ to $\mathbf{C}_2$. Write down the additional constraints on the coefficients of $\mathbf{C}_2$, and lastly the overall constraints on $\mathbf{C}_2$ when preserving *both* position and velocity constraints.

(3) In this question, assume: that values referred to by x, y, i, j below are in $\mathbf{N}$ (the non-negative integers), that your raster image/device contains unit-square pixels with integral centres, that pixel (0,0) refers to the lower-left corner, and that pixel $(n-1, n-1)$ refers to the top-right corner (on a device of resolution $n \times n$). We will look at the error characteristics of the *rasterisation* of real line segments $\mathbf{L}$ that lie in the first octant with one endpoint at the origin. The "optimal" line rasterisation that Bresenham's algorithm implements is defined as

$$R_o[\,\mathbf{L}\big((0,0),(x,y)\big)\,] \;=\; \Big\{\,(i,j) : 0 \le i \le x,\ j = round\Big(i\,\frac{\Delta y}{\Delta x}\Big)\Big\},\ \text{with } 0 \le y \le x.$$

This states that the rasterisation R_o transforms any real line segment $\mathbf{L}\big((0,0),(x,y)\big)$ in the first octant of $\mathbf{R}^2$ to a set of pixels (i, j) that are turned "on" by the rasterisation. Assume that $(x, y) \ne (0,0)$, so in our case $\Delta x = x$, $\Delta y = y$, and the slope of the line segment is well defined. For an arbitrary line rasterisation R, we define the error it makes on the ideal line segment $L = \mathbf{L}\big((0,0),(x,y)\big)$, $0 \le y \le x$ as

$$\varepsilon(R[L], L) \;=\; \sum_{(i,j)\,\in R[L]} \left|\,\frac{\Delta y}{\Delta x}\,i \;-\; j\,\right|.$$

This just measures the vertical distance between the ideal line segment and the centre of the pixel(s) chosen by R, for each column of pixels. To get some idea of the worst-case performance of R, we define a worst-case error metric ε_{max} for line segments in the first octant as follows:

$$\varepsilon_{max}(R, n) \;=\; \max\Big\{\,\varepsilon(R[L], L) : L = \mathbf{L}\big((0,0),(n,y)\big), 0 \le y \le n\Big\}.$$

Prove that $\varepsilon_{max}(R_o, n)$ is $O(n)$. That is, show that for all line segments in the first octant, R_o induces $O(n)$ (i.e., at *most* $k_0 n$, $k_0 > 0$, for some constant k_0) error. Derive a specific value for k_0.

Bonus: prove that $\varepsilon_{max}(R_o, n)$ is also $\Omega(n)$: that there exists a line segment (indeed a family of line segments) in the first octant that causes R_o to make $\Omega(n)$ (i.e., at *least* $k_1 n$, $k_1 > 0$) error, for a constant k_1. Derive a specific value for k_1.