COMP27112 Introduction to Visual Computing (70)

Vector-based Display draw lines, texts, and points. Think of SVG.
Pixel-based Display draw pixels, which is effectively estimations of shapes and everything, like PNG.

Fixed Graphics Pipeline includes *transformations and viewing*, *lighting*, *primitive assembly (clipping)* that outputs primitives, *rasterisation (scan conversion)* that outputs fragments, and fragment operations like *blending* and *depth test*.

Primitive in a graphics pipeline is bit of geometry that has been clipped to generate fragments.
Fragment in a graphics pipeline is a candidate pixel, which may or may not end up being displayed.
Depth Test in a graphics pipeline is used to detect hidden surfaces that should or would not be rendered in the final display.

Programmable Graphics Pipeline includes *vertex shader program* (programmable, runs per-vertex), primitive assembly (clipping), rasterization, fragment shader program (programmable, runs per-fragment), and fragment operations (hidden surface removal).

Normalisation of Vector is to take a vector of any length and, keeping it pointing in the same direction, change its length to 1, turning it into a unit vector.

Cross Product of Vector of two linearly independent vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, is a vector that is perpendicular to both a and b, and thus normal to the plane containing them. $\boldsymbol{a} \times \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}| \sin(\boldsymbol{\theta})\, \boldsymbol{n}$
Dot Product of Vector of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, is a value calculated as $\boldsymbol{a} \cdot \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}| \cos(\boldsymbol{\theta})$.

Tessellation or tiling is the covering of a surface, often a plane, using one or more geometric shapes, called tiles, with no overlaps and no gaps. In graphics rendering, the polygons are convex (opposite of concave), and are usually triangles.

Z-buffer is a type of data buffer to represent depth information of objects in 3D space from a particular perspective (usually, the invisible camera).

Viewing in computer graphics is to simulate the real world in a way that realism is maximised.

Duality of Modelling and Viewing is the property that moving the model by some transformation is equivalent to applying the inverse transformation to the camera viewing it. Basics for viewing.

Parallel Projection is the projection in which the rays of light from the scene are parallel to each other as they strike the projection plane (light source infinitely far). Very accurate, used in CAD.
Perspective Projection is the projection in which the rays of light from the scene converge at a single point, called the vanishing point. This type of projection is often used for creating realistic images.

Projection Normalisation is to convert 3D points into 2D points that can be displayed on a monitor or other display device while preserving the Z-plane, which is used to create overlays. Otherwise, all will be the same as the projection plane.

Local Model of light-matter interaction treat each object in a scene separately from any other object.
Global Model of light-matter interaction treat all objects together, and model the interactions between objects.

Diffuse Reflection makes materials look dull and matte, as light reflected goes to all kinds of directions.
Specular Reflection makes materials look shiny, as all light reflected goes to the same direction.

Ambient Light approximates the overall light level in an environment.

$I = ambient + distance(diffuse + specular) = k_a I_a + \frac{I_p}{d'}\left[k_d(\hat{\mathbf{N}} \cdot \hat{\mathbf{L}}) + k_s(\hat{\mathbf{R}} \cdot \hat{\mathbf{V}})^n\right]$ where,

$d' = k_c + k_l d + k_q d^2$ with k_c, k_l, and k_q tuned for best ***(not accurate)*** results; k_a is the *ambient reflection coefficient*; $\hat{\mathbf{N}}$ is *normalized surface normal*; $\hat{\mathbf{L}}$ *is normalized direction of light souce*; I_p is the *intensity of light source*; k_d is the *diffuse reflection coefficient*; $(\hat{\mathbf{N}} \cdot \hat{\mathbf{L}})$ is also $\cos(\theta)$, *cosine of the angle of light and surface normal*; $(\hat{\mathbf{R}} \cdot \hat{\mathbf{V}})$ is also $cos(\phi)$, *cosine of the angle of specular light and viewing angle*; k_s is the *specular reflection coefficient*; n is a value tuned for best results, usually between 0 and 200.

Gouraud Shading interpolate pixel colour along a scan line, takes 2 flops per pixel.
Phong Shading interpolate the normal vector along a scan line, takes 20 flops per pixel.

Mach Banding is the effect that humans perceive changes of intensity greater than what it should be.

Mip-mapping is a technique that would choose the two closest textures, and do bilinear interpolation for extra smoothness, to deal with mismatches between texel and pixel resolutions.

Bump Mapping is a technique that would change the surface normal during rendering to make the objects make bumpier.

Perceptual Colour Spaces are colour spaces that are designed to be more perceptually uniform. A change of the same amount in a colour value will produce a change of about the same visual importance.

Colour Standards are necessary so that a colour can be described exactly to others.

Point-processing is manipulating an image where an output pixel value depends only on its corresponding pixel in the input image.

Field of View (FOV) of a camera influences the smallest object that can be seen.
Number of Pixels influences the size of the smallest object that can be seen.

Thresholding an image will turn a picture into a binary form based on the pre-defined threshold value. The best threshold value is the one that minimised the *misclassification rate*.

Geometric Transform will rearrange the location of pixels. This includes cropping, rotating, resizing, and wrapping. Technically, options are: *translation*, *rotation*, *scaling*, and *shear*.

Affine transform is a linear transformation that preserves points, straight lines, and planes. This means that it will not change the shape or size of objects, but it can move them around.

Kernel is a small, square array of numbers that is used to filter the input function.

Convolution Operator takes two functions and produces a third function. The third function is a weighted sum of the two original functions, where the weights are determined by the *kernel*. Its purpose is to assess the similarity between the two inputs. Time complexity for an image with r rows and c columns and a x by y template is $O(rcxy)$.

Noise in an image is any factor in the imaging system that gives rise to perturbations. They are assumed to be *zero mean* and *Gaussian distributed*.

Median Smoothing sets the colour of the pixel to the medium value of the kernel. It is less sensitive to the outliers (usually noises) while have a good preservation of the edges and significant image structures.
Gaussian Smoothing uses a kernel generated by Gaussian distribution. Its results resemble a translucent screen / out of focus lens. It preserves the pattern and introduces fewer artefacts at the edges of objects.

Edge is a vector, it has magnitude of $sqrt(x^2 + y^2)$ *[sum or x and y derivatives in quadrate]* and direction $atan2\left(\frac{y'}{x'}\right)$ *[inverse tangent of the ratio of the y and x derivatives].*

Template Matching find known objects in an image using a template abstracts the shape of the object, and hence it is bad to find patterns that have a lot variations.
Template is a model of the object to be found.

Blob is a set of pixels that a) Share some property and b) Are connected.

Connectivity of Blob is how to determine 'connected'. *4-Connected* checks for pixels on four directions, but joining at corners can be dropped. *8-Connected* solves the corner problem but can pierce thin objects.

Connected Component Analysis / Region Labelling is a process to identify groups of contiguous pixels and to label separate blobs. Background may be labelled as 0 and blobs from 1 to N. Issues may be dealt with using different techniques of image thresholding. It needs 2 passes, first one give labels and mark equivalence, the second re-label all equivalents.

Boundary Detection works by first finding the first pixel LRTB and the first background as the one on the left, and then go find next by checking clockwise from first background, with the last background checked as next background. Stops when returning to first pixel and next object pixel found is the one next (preventing the case where boundary continues off in another direction).

Chain Code is a way to describe the boundaries of a shape, by numbering 8 directions from 0 to 7.
Differential Chain Code is the difference between elements of *Chain Code* and is rotationally irrelevant.

Shoelace Algorithm Area = $\frac{1}{2}|\Sigma(D)| = \frac{1}{2}|\sum_{i=1}^{n}(x_i y_{i+1} - x_{i+1} y_i)|$ [1] It uses location of vertices to get area size.

Moments of Area is defined as $M_{\alpha\beta} = \sum_{image} x^a\, y^b I(x,y)$. Where, $(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}})$ gives region's centre of gravity, and for binary images, $\alpha = \beta = 0$ gives sum of pixels, $(\alpha,\beta) = (1,0)$ gives the sum of x values of pixels.

Central Moments of Area is defined as $M_{\alpha\beta} = \sum_{image}(x - \bar{x})^\alpha\, (y - \bar{y})^\beta I(x,y)$, where $(\bar{x},\bar{y})$ is the centre of gravity.

Run-Length Encoding (RLE) is a method of compression that stores value and length to compress long and duplicating data. It may result in negative compression rate in bad cases.

Hough Line in *Hough Parameter Space* represents every possible line through the point in (x, y) space. It is in $y = mx + c$ form, but would be altered into $c = -mx + y$ in HPS. It might be much better to place it using polar coordinates so that it would have no problem to obtain otherwise impossible (vertical) lines.

Hough Circle in Polar Coordinates: $r = x\cos(\theta) + y\sin(\theta)$

Erosion is process to 'erode' the outer structure of a blob: if the centre of *structuring element (SE)* is over an object pixel and any of the other elements of the SE are over a background, the centre becomes background.
Dilation is the process to 'dilate' the borders of a blob: if the centre of the *structuring element (SE)* is over a background pixel and any of the other elements of the SE are over an object, the centre becomes object.

Opening will get rid of 'sticky out bits' and also enlarge holes. $\text{open}(I) = \text{dilate}(\text{erode}(I))$
Closing will fill in holes and fill out 'sticky out bits': $\text{close}(I) = \text{erode}(\text{dilate}(I))$

---

[1] This is not a mathematically sound representation – when it comes to n+1st value, 1st value will be used.