Syllabus

Lectures in Graphics

Jarek Rossignac
Interactive Computing
Georgia Tech
Revised on July 12, 2012

This is the combined syllabus for several courses in Visual Computing offered at Georgia Tech.

The material posted here is organized into some 500 lecturettes (very short lectures), which are each assigned a category and a level of difficulty (which corresponds to a particular CS course).

The **categories** include:
ACC (acceleration): Culling, occlusion, levels-of-detail, panoramas, textures, and other acceleration techniques
ANI (animation): Represent and animate motions, morphs, and deformations of shapes
CUR (curved geometry): Represent, create, and process curves and surfaces (ellipse, Bezier, J-splines, NUBS…)
EXA (exams): Test samples, questions and past exams, quizzes, midterms
GEN (general): Course overview, policy, grading, software used, theory principles, project guidelines
GRA (graphics): Transforms, perspective, clipping, lighting, rasterization, and z-buffer in the graphics pipeline
GUI (graphic user interface): Design and implement user interaction (pick, drag, 3D and multi-touch input)
LIN (linear geometry): Create and process geometric primitives (points, vectors, frames, lines, planes, triangles).
MED (media): import, use, or export images, photos, textures, audio, and videos
MOD (modeling): Construct and process representations of shapes and scenes (polygons, triangle meshes, patterns).
PHO (photorealism): light, reflections, BRDF, shadows, highlights, ray tracing, radiosity, and other effects
PHY (physics): Equation of motion, collision prediction and reaction, flow simulation, dynamic systems
PRO (projects): Description of a project or project component, with deliverables, demos, reading material, code template
SOF (software): Tutorial of how to install and use the Processing language and its graphics API
VIS (visualization): Perception (color, resolution, illusions) and how to better visualize shapes, structures, and animations

Levels of difficulty are color coded as follows:
Clover green (apprentice): Every student should know this material and should be able to use it correctly.
Midnight blue (pro): Every graduate student and most developers in the field should understand this material and be able to reinvent it and explain it.
Cayenne red (guru): Advanced material for those interested in improving performance and scalability of graphic and animation applications, or interested in research and/or teaching.
Black (unassigned): Back-up material not included in any class, but of potential use in projects.

The short description of each lecturette summarizes what the students are expected to know

This syllabus and linked material are posted at http://www.cc.gatech.edu/~jarek/LecturesInGraphics/
The linked material includes:
- Videos explaining or demonstrating various concepts or projects: videoXXX.m4v or videoXXX.mov
- Processing sketches sketchXXX.pde or zipped folders (source code and data files): sketchXXX.zip
- Link to web pages with further details or different perspectives: pageXXX.html,
- My notes and assigned reading (mandatory, unless in parentheses): noteXXX.pdf

This material is primarily designed to support the following courses at Georgia Tech,
but may be of value for other Visual Computing courses in rendering, animation, virtual and augmented reality, game design, and video effects:
- Undergraduate introduction to Computer Graphics (CS 3451)
- Graduate course in Computer Graphics (CS 6491)
- Advanced graduate course on 3D Complexity Techniques for Graphics, Modeling, and Animation (CS 7491)

Each course uses a different subset of lecturettes. Furthermore, depending on the instructor, students' skills and interests, the knowledge based needed for the projects, the progress made by the class that semester, and the evolution of the field, the particular subset and the order of the lecturettes may be different each time the course is taught.

But, unless specified otherwise, the following color coding applies to the different courses:
- students of CS3451 are accountable for understanding and knowing the material in green,
- students of CS6491 are responsible for knowing the material in green and being able to explain and use the material in blue, and
- students of CS7491 are responsible for knowing the material in green and blue, and for understanding the research results and open questions in red.

Overview

GEN	**Scope of the course**: What is covered in the course, target audience, resources. video001,
GEN	**Grading**: tests, projects, presentations, attendance, late policy (discussed in class)
GEN	**Projects**: Source (comments, header), demo (picture, name, GUI), write-up (header, format), videos, upload, team projects
GEN	**Resources**: Text book, examples & demo sketches, class notes, videos, slides, papers, sites

Intro to Processing and 2D graphics

Getting started and basics components of a Processing sketch

SOF	**Install & run**: Download Processing, run, edit, save, export. Processing site, video002,
SOF	**Structure of a sketch**: Global variables, initialization, display loop, user actions. sketch001, video003, tutorials
SOF	**Software constructs**: variables, methods, arrays, conditionals, loops, functions, classes, objects. reference, online Tutorials
SOF	**Learning, writing, and debugging code**: Load/edit examples, comments, references, coding and debugging techniques. ProcessingBooks.pdf

Graphics primitives and user interaction in 2D

GRA	**Canvas**: Coordinate system, canvas size, center, clear, background, double buffering. sketch002,
GUI	**Mouse**: Current and previous position, actions/modifiers (press, drag, release), rubber-band, sketch002,
GRA	**Shapes**: line, rectangle, disk, triangle, quad, polygon, weight. sketch004,
SOF	**Colors**: stroke & fill states, HLS color systems, stroke/fill, color picker, color ramp, opacity. sketch003,
GUI	**Drag disk**: display disk, drag with mouse, snap vs relative drag. sketch001,
GRA	**Display disks**: Retained model**,** array of disk objects, initialize, display, color, label. sketch003,
GUI	**Pick & modify disk**: pick center or border and drag to modify. sketch005,
GUI	**Append disk**: click&drag to append disks of different radii. sketch006
GUI	**Insert or delete disk:** click to delete disk, click & drag to append after and update array.

Text, keys and files

SOF	**Manipulate text**: Char and string, operations and conversions, concatenation, formatting
SOF	**Text**: debugging print, write on canvas, load font, write coordinates next to mouse, label a point
GUI	**Help**: show and toggle help menu, assign actions to keys, state change vs while pressed
SOF	**File Read/write**: saving data to file, reading from file, managing file names
SOF	**File selection**: Define fixed, automatic, or user-specified path for loading or saving files
SOF	**Keyboard input**: read, parse

Images and videos

MED	**Images**: save snapshots of canvas as images, load and display images
MED	**Pixels**: read and change pixels in image and on screen, image processing
MED	**Videos**: make videos from saved snapshots, capture & process live video from camera
MED	**Audio**: load, play, record, save, analyze audio
MED	**SVG**: output vector graphics files
MED	**PDF**: output PDF files

Transformations

MOD	**Transformations**: translate, rotate, scale, order and non-commutativity.
MOD	**Fixed point**: Rotate or scale around a given fixed point.
MOD	**Matrix stack**: push/pop operations, sketch003,
MOD	**Patterns**: Translating, rotating, spiraling, patterns of patterns
MOD	**Scene graph**: Articulated shape, 2D human stick figure
MOD	**Recursive patterns**: Spirals, snowflakes, tree, P010.jpg, sketch010,

Animation

ANI	**In-betweening**: position and radius of disk controlled by mouse
ANI	**Time**: update time at each frame, loop / round trip, ease-in / ease-out, frame rate, speed control
VIS	**Ghosts**: elastic chain of previous positions while dragging or animating a disk
PHY	**Free fall in box:** Equation of free fall motion, collision with cube boundaries, reflected velocity

Projects on 2D graphics

PRO	**Drag your picture and your name**: Demo, file header, submission process
PRO	**Rotate and scale**: Relative to center of picture, use transforms and interrupts, nice GUI
PRO	**Single click&drag**: Specify translation, rotation, and scaling in a single gesture
PRO	**Edit disks**: Interactive editor for a set of disks (append, move, delete), display with IDs, archival
PRO	**Disk chain creation**: Attach tangentially next disk to previous
PRO	**Prevent disk overlap**: Limit disk creation and size to the empty space, combine with disk chain.
PRO	**Disk chain editing**: Grow disk as much as possible, Apollonius problem. Computational complexity.
PRO	**Insert maximal disk**: Grow (and adjust) disk as much as possible, Apollonius problem. Computational complexity.
PRO	**Fill ring**: Fill largest disk with all the other ones. Principles, strategy, complexity, heuristics.
PRO	**Follow**: Explain Examples>Topics>Interaction>Follow3
PRO	**Reach**: Fix Examples>Topics>Interaction>Reach3

Geometric primitives in 2D

Overview

GEN	**Points & Vectors**: Semantics, importance in graphics, representation, notation, usage, difference
GEN	**Implicit, parametric, procedural**: Definition, notation, representation, examples, advantages, conversion
GEN	algebraic sets, analytic sets, procedural sets, fractals

Points

LIN	**Implementation:** Class, methods, display, label, creation, assignment.
LIN	**Distance**: distance between points, pick closest point, two closest neighbors and tracing
LIN	**Affine combinations**: LERP, Different formulations, more points, constraint, implementation
LIN	**Bi-linear interpolation**: (s,t) mapping, mouse control, parametric expression, inverse map

Vectors

LIN	**Implementation**: Class, methods, display as line/arrow at point, creation from points,
LIN	**Comparing:** testing vector equality, number of constraints
LIN	**Special alignment**: testing whether two vectors are parallel or orthogonal
LIN	**Linear combinations**: scaling a vector, add/subtract, weighted combination
LIN	**Magnitude and direction:** formula, norm vs normalization, angle vs direction, unit vectors
LIN	**Angles**: computing the angle between two vectors, normalizing angle between -180 and +180
LIN	**Dot-product**: definition, properties, computation, use as test, use as constraint
MOD	**Reflected vector**: from a line
LIN	**Cross-product (2D version)**: definition, properties, computation, use as test, use as constraint
LIN	**Other vector products**: geometric, outer
MOD	**Rotated vector**: 90 degree rotation formula and justification, arbitrary angle rotation
MOD	**Polar coordinates:** cartesian vs polar, conversions
LIN	**Dot-product proof:** representing vectors as complex numbers, Euler's formula, proof of dot product formula
MOD	**In-betweening**: Linear versus steady interpolation between vectors, de Moivre's formula
EXA	**Review:** Linear algebra for graphics, PDF,

Lines

MOD	**Implicit line and half-space**: defined by point and normal, defined by 2 points.
MOD	**Parametric line and ray**: defined by point and tangent, by two points
LIN	**Special configuration of two lines:** orthogonal, parallel, coincident
LIN	**Point projection**: compute normal projection of point on line
LIN	**Point distance**: compute point/line distance
MOD	**Numeric rounding**: floating point round-off, representable points and lines

Linear halfspaces

MOD	**Representation of a half-space**: point&normal, point&tangent, two-points, implicit equation
GUI	**Display a half-space**: line through screen, sidewalk, tangent/normal arrow, label inside
MOD	**Point-in-halfspace inclusion test**: Implicit, parametric, round-off errors, cost
MOD	**Numeric rounding**: point inclusion tolerance

Rays

LIN	**Applications:** photon, collisions, visibility/shadows
LIN	**Representation**: point and (unit) tangent direction
LIN	**Reflection:** principle, formula of reflected ray, its derivation

Edges

MOD	**Motivation:** borders, trajectories, skeletons
MOD	**Parametric representation**: representation by end-points, map from [0,1], evaluation
MOD	**Implicit formulation**: Semi-algebraic representation, point-inclusion test
MOD	**Point-in-slab test**: test whether a point projects inside the edge
MOD	**Point-in-edge test**: test whether a point lies in an edge, distinguish inside and endpoint
MOD	**Point-edge distance**: formulation, justification, computation
MOD	**Closest point projection**: on edge
MOD	**Edge/edge distance**: Definition, computation, applications
MOD	**Singular cases**: Vertex contacts, parallel, overlaps

Triangle

GRA	**Rendering**: Closed shape, smooth fill, linear border
MOD	**Construction**: from 3 sides
MOD	**Centroid**: definition, intersection or medians, average of points
LIN	**Barycentric coordinates**: formulation, computation
LIN	**Local vs barycentric coordinates**: conversion
MOD	**Measures and properties**: angles, distances, perimeter, signed area, circumradius
MOD	**Point-in-triangle inclusion test**: areas signs, barycentric coordinates
MOD	**Triangle centers**: barycenter, orthocenter, circumcenter, Euler line, other centers
MOD	**Triangle subdivision**: in 2, 3, 4

Circle and arc

CUR	**Implicit circle/disk**: equation, inclusion test.
CUR	**Parametric circle**: equations, inclusion test
CUR	**Arc**: display arc, fit arc through 3 ordered points.
CUR	**Fit circle to 2 points and a tangent**: compute center and radius of circumcircle
CUR	**Fit circle to 3 points**: compute center and radius of circumcircle
CUR	**Closest point projection**: on circle, on arc
CUR	**Fast display of circle:** avoid trig at each step, use GPU
CUR	**Circle inversion**: Formula, properties, and applications
CUR	**Conformal maps**: Definition, applications, case of circle inversion
CUR	**Appolonian gasket**: Definition, generation
CUR	**Appolonian circles**: Definition, generation, applications

Intersections

LIN	**Line/line intersection computation**: special cases, solving using implicit forms, parametric forms, mixed forms
MOD	**Edge/edge intersection test**: using parameter, direct test, overlap, contact
MOD	**Edge/edge intersection test -- proof**: prove that the direct intersection test is necessary and sufficient
MOD	**Edge/edge intersection acceleration**: bounding boxes, cells, trees
LIN	**Ray/edge intersection**: mixed form, proper solution, and application to visible edge in 2D
CUR	**Ray/circle bounce**: intersection, reflection

MOD	**Edge/triangle intersection test**: formulation, computation
MOD	**Triangle/triangle intersection test in 2D**: half-space intersection
MOD	**Acceleration of triangle/triangle intersection tests in 2D**: disks, boxes,
CUR	**Circle/circle intersection**: Tes, contact plane
CUR	**Ray/circle intersection**: Mixed form, proper solutions
CUR	**Ray/arc intersection**: Mixed form, proper solutions
CUR	**Arc/arc intersection**: Mixed form, proper solutions

Frames and local coordinate systems

LIN	**Local frame**: basis, origin, local coordinates interpretation
LIN	**Computing local coordinates**: relative vector, coordinates
LIN	**Change of coordinate systems**: relative vector, coordinates
LIN	**Affine transforms**: {I,J,O} representation, 2x3 matrix form
LIN	**Homogeneous coordinates**: matrix, points, vectors, multiplication
LIN	**Affine transforms**: {I,J,O} representation, 2x3 matrix form
LIN	**Cascades of affine transforms**: associativity, non-commutativity
LIN	**Inverse of an affine transform**: Definition, calculation, inverse of a product
LIN	**Rigid transform**: definition, properties, rotation angle, translation, calculation, inverse, applications
LIN	**Similarity transforms**: definition, properties, applications, computing the scale, inverse

Quad

MOD	**Quad**: definition, interpolation, application to hight field
GRA	**Rendering**: Closed shape, linear border, smooth fill
MOD	**Bilinear interpolation** : Parametric form, calculation, implicit form
MOD	**Inverse of a bilinear interpolation** : Formulation, implementation, applications
MOD	**Triangulation of a quad** : diagonal, ambiguity, fan, error

Projects on 2D primitives

PRO	**Edge/edge intersection**: Implementation, visualization and testing
PRO	**Edge soup editor**: Implement display, select, add, edit, delete, save, load edge soup
PRO	**Ray path in triangle**: Trace finite length ray in triangle, interactive ray dragging
PRO	**Draw a spiral**: Using recursive pattern, iterative pattern and trigonometry
PRO	**Arc/arc intersection**: implementation
PRO	**Ricochet**: Compute direction of ray from A to reach B while bouncing off edge soup

Curves in 2D

General

CUR	**Representation**: Parametric/implicit, algebraic/procedural, interpolating/approximating
CUR	**Derivatives**: Tangent, curvature, from continuous function, from samples
CUR	**Continuity**: C^k vs G^k, measure, approximation
CUR	**Limitations**: Domain, accuracy, scalability

Parabola

CUR	**Parabola equation**: constant acceleration, integration, parametric form
ANI	**Free-fall simulation**: update P and V, correction, 3-point prediction
ANI	**Free-fall shooting**: Compute angle to hit target, maximum reach
CUR	**Bézier quadric from bi-linear diagonal**: demonstration, justification, implementation, sketch011.zip
CUR	**Parabolic interpolant from bi-linear diagonal**: demonstration, justification, implementation
CUR	**Non-uniform parabolic interpolant**: formulation, advantages, implementation

Cubic polynomial

CUR	**Cubic Bézier:** geometric construction, implementation, parametric form
CUR	**Cubic Bézier convex hull:** justification, union of triangles, applications
CUR	**Cubic Bézier subdivision:** geometric construction, justification, implementation, applications
CUR	**Cubic Bézier derivatives:** parametric form, at ends
CUR	**Hermite interpolation**: formulation, conversion to Bezier, reverse conversion
CUR	**HIgher order interpolation**: interpolate end points, velocities, and curvatures
CUR	**Interpolating cubic**: general solution
CUR	**Neville's algorithm**: geometric construction, justification, implementation

Bézier splines and Catmull Rom splines

CUR	**Bézier splines:** Hermite spans, C1 continuity
CUR	**Specify velocity and curvature**: 3 points per sample
CUR	**Catmull-Rom splines:** estimating derivatives, formulation as Hermite spans

B-splines

CUR	**Uniform B-spline:** spans, control polygon of span, evaluation
CUR	**Uniform B-spline conversion to Bézier:** Image of control point, conversion
CUR	**Non-uniform B-spline:** formulation, intuition, sevaluation

Polyloop smoothing

CUR	**Polyloop**: definition, representation, class, n(), p(), display, edit points, edit curve, archive
CUR	**Polyloop operations**: split, dual, tuck(s), implementation
CUR	**Smoothing**: tuck(⅔)*, shrinking problem, (tuck(⅔),tuck(-⅔))*

Piecewise circular curves (PCCs)

CUR	**Bi-arc**: definition, degree-of-freedom, computation
CUR	**Bi-arcs**: inscribed, interpolating

Polyloop subdivision

CUR	**Quadratic B-spline subdivision**: (split, tuck(½))*
CUR	**Cubic B-spline subdivision**: (split, tuck(½), tuck(½))*
CUR	**Non-uniform cubic B-spline subdivision:** formulation, intuition, application, benefits
CUR	**Quintic B-spline subdivision**: (split, tuck(½), tuck(½), tuck(½), tuck(½))*
CUR	**FourPoint subdivision**: (split, tuck(½),tuck(-1))*
CUR	**FourPoint subdivision using parabola average:** formulation, left
CUR	**J-splines**: **definition** J_s=(split, tuck(½), tuck(s/2-1))*
CUR	**J-splines properties**: J_0=FourPoint, J_1=Cubic B-spline, J_(3/2)=Quintic B-spline
CUR	**Jarek's curve**: J_½ ,motivation, popping minimization, area preservation
CUR	**Variable width**: normal, radial, ball offsets, definitions, computation
CUR	**Ringing**: Footprint, rings, algorithm, further improvements

Polyline subdivision

CUR	**FourPoint subdivision using parabola average:** no average at the end
CUR	**End-extension heuristics:** parabolic prediction

Polyloop resampling

MOD	**Arc-length measure**: definition, implementation, total length
MOD	**Uniform arc-length resampling**: algorithm
MOD	**Equidistant resampling**: Definition (constant edge length), challenge

Curvature

CUR	**Curvature estimators**: circle, parabola

CUR	**Curvature sensitive resampling**: benefits, strategies, implementation

Animation

ANI	**Animate a point**: parametric forms, subdivision form, resampling
ANI	**Ease-in and -out**: Bezier splines, B-splines, time warp
ANI	**Animate a curve**: animate its control points, show/edit keyframes or trajectories, sketch012.zip
ANI	**Compare curve animations**: cyclic keyframed motion

Projects

PRO	**Visualize acceleration and jerk**: vectors, dynamic, 3D

Animations in 2D

Physics

PHY	**Free fall:** parabolic path, advection/correction, prediction, aim,
PHY	**Dynamic:** translation, forces, rotation, torques
PHY	**Inertia:** formulae of inertia moments, computation for polygons
PHY	**Drag:** physical drag with constraint or using a spring
PHY	**Collision test:** detection/prediction (continuous) for disks and polygons
PHY	**Collision response:** contact plane, forces, friction
PHY	**Complex collisions:** strategy, examples
PHY	**Particles:** simulate motion, density, pressure, viscosity
PHY	**Fluid:** Eulerian/Lagrangian formulations, Navier-Stokes equation, advection, area preservation, interaction with air and solids

Point and curve animation

ANI	**Animate a point**: parametric forms, subdivision form, resampling
ANI	**Ease-in and -out**: Bezier splines, B-splines, time warp
ANI	**Animate a curve**: animate its control points, show/edit keyframes or trajectories
ANI	**Curvature morph**: compute curvature, morph, reconstruct, register
ANI	**Compare curve animations**: cyclic keyframed motion

Primitive motions

ANI	**Rigid motions**: Definition of congruency, rotation/translation, formulation, computation of morph, comparison with dynamics
ANI	**Spiral motions**: Definition of similarity, spiral, formulation, computation of morph, comparison with dynamics
ANI	As-Rigid-As-Possible: Definition, computation, propertites
ANI	**Steady motions**: Definition of affinity, steady, formulation, computation of morph, solvability, properties, comparison with
ANI	**3 frame interpolant**: Definition, solutions, bi-SAM, benefits, UAM, comparison

Composite motions

ANI	**Subdivision of key framed polySAMs**: Definition, computation, examples, special cases

Cloud motions

MOD	**Registration**: Rigid, similar, affine
ANI	**Moving Least Square Blending:** Definition, computation, examples, special cases
ANI	**SAMBA:** Problem statement, formulation, computation, examples, special cases

Projects

PRO	**Cyclic keyframe animation editor**: design, archive, play smooth animations

Geometric arrangements in 2D

Point cloud

LIN	**Transformation**: translation, rotation, rigid motion, scaling, similarity, shear/squash, affine
LIN	**Centroid**: formula, rotate / scale around centroid
MOD	**Point cloud registration**: rigid, similar, affine
MOD	**Delaunay triangulation**: definition, computation, applications (spanning tree, triangulation)
LIN	**Convex hull**: Intuition, definition, computation
MOD	**Voronoi regions**: definition, point-in-cell test, Voronoi vertices, duality with Delaunay
MOD	**Alpha hull, shape, and complex**: definition, properties, computation, applications
MOD	**Gabriel graph**: definition, computation, properties, applications
MOD	**Computational geometry**: computational complexity
MOD	**Fast convex hull**: Jarvis march, Graham scan, Aki-Toussaint acceleration
MOD	**Fast Delaunay**: Insertion (edge test and flip, propagation), sweep line

Half-space and line arrangements

MOD	**General configurations**: definition, importance, realization
MOD	**Cells**: Definition, properties, point inclusion test, identification
MOD	**Counting cells**: count edges and faces in general configuration of lines
MOD	**BSP**: definition, representation, construction, point classification
MOD	**CSG**: definition, representation, construction, point classification, stack
MOD	**Active Zone in CSG**: definition, properties, use, applications
MOD	**BSP/CSG comparison and optimization**: classification cost, footprint, optimization
MOD	**Blist**, definition, CSG-to-Blist conversion, footprint, OBF

Polygons

MOD	**Simple polygon**: Definition, interior, representation, display, editing
MOD	**Face**: disjoint, holes, non-manifold vertices, represent with simple polygon
MOD	**Self-crossing polygons**: winding number, alternating parity
MOD	**Point-in-polygon**: ray-intersection parity
MOD	**Point-in-polygon**: point-in-triangle parity
MOD	**Point-in-polygon**: fast and robust implementation

Edge soup

MOD	**Edge soup:** definition, interactive editing
MOD	**Intersections:** testing, computing, ordering
MOD	**Sidewalk loops**: Right-most turn, follow loop
MOD	**Point-in-loop test**: ray intersection parity
MOD	**Faces**: Loop containment graph

Triangle meshes in 2D

Planar graphs

MOD	**Planar graph definition:** Vertices, reactively open edges, open faces, disjoint, valence
MOD	**Connected components of a planar graph:** Definition, computation (union find)
MOD	**Euler-Poincaré formula:** formula, examples, importance
MOD	**Euler-Poincaré formula proof:** derivation/proof

Triangle mesh representation and traversal

MOD	**Maximal planar graph**: planar triangulation, T=2v-4, average valence
MOD	**Meshes with borders**: definition, counting. tracing borders

MOD	**Corner operators**: n, p, s, u, o, l, r, t, v, c
MOD	**Representations**: Corner Table (CT), Swing Table (ST), Winged Edge (WE), others
MOD	**Construction**: Quadratic algorithm
MOD	**Fast construction**: Swing lists, sorted swing lists, hash table
MOD	**Streaming**: Problem statement, motivation, approaches
MOD	**Graph drawing**: Integer coordinates, Schnyder woods
MOD	**Compact representations**: SOT, SQuad, Zipper, Meshlets
MOD	**Non-manifold border vertices**: definition, counting, tracing borders, extending CT and ST

Mesh processing

MOD	**Valence**: local computation, global computation
MOD	**Border**: identify border edges
MOD	**Components**: identify connected components
MOD	**Border loops**: count, trace border loops, trace border between two parts
MOD	**Smoothing**: border, interior
MOD	**Regular subdivision**: definition, implementation, point estimation
MOD	**Adaptive subdivision**: definition, criteria, implementation

Connectivity compression

MOD	**Swirl traversal**: definition and C, L, E, R, S triangle labels, shallow stack implementation
MOD	**Vertex spanning edge tree (VSET)**: definition, computation, encoding
MOD	**Vertex spanning triangle tree (VSTT)**: definition, encoding, border
MOD	**Topological surgery**: definition, encoding
MOD	**Compression with offsets**: definition and reconstruction algorithm
MOD	**EdgeBreaker**: formula for implicit offsets, decompression (build and zip)

Projects

PRO	**Point cloud editor**: Append, pick, move, delete, translate, transform all (rotate, scale, around centroid)
PRO	**Graph editor**: Add/delete edge; move add, pick, delete, move vertices; transform all
PRO	**Ray path in edge soup**: Trace finite length ray in edge soup, interactive ray dragging
PRO	**Edge soup connectivity**: Edit, save, read, edges. Detect all edge/edge intersections. Compute connected sets.
PRO	**Graph processing**: Test if connected, compute minimum spanning tree
PRO	**Draw fat stroke**: Capture speed, variable weight, draw as 2 rows of quads, smooth shading with flat normals
PRO	**Edge soup walks**: Trace and count sidewalk loops for crossing edge soup

Projects

PRO	**Delaunay triangulation**: Show Delaunay triangles and closest neighbor oriented edges
PRO	**Constrained Delaunay**: Delaunay triangulation that respects non-crossing edge soup

Morphology

Distance

MOD	**Distance measures**: Minimum, Hausdorff, Frechet, Ball
MOD	**Voronoi:** Definition, computation

Correspondence

MOD	**Correspondence formulations**: parametric, arc-length, closest, normal, ball
MOD	**Compatibility**: normal/ball, definition, properties

Offset and blends

MOD	**Blending functions**: Ricci’s blends, RBF, r-sets, issues.
MOD	**Grow & shrink**: Definition, computation (discrete, continuous), applications
MOD	**Area compensation**: offset distance formula, curvature-dependent offset, motivation, formula

MOD	**Fiil & round**: Definition, applications
MOD	**Tightening**: Definition, computation, applications
MOD	**Relative rounding**: Motivation, definition, computation, applications

Medial Axis

MOD	**MAT**: Definition, computation (discrete, continuous), applications
MOD	**Envelop**: Normal/ball,radial, definition, computation
MOD	**Curve average**: Definition, computation (discrete, continuous), applications

Minkowski sums and morphs

MOD	**Minkowski sum**: Definition, computation, implementation for convex polygons
MOD	**Minkowski morphs**: Definition, computation, properties, applications to animations
MOD	**Minkowski morphs of non-convex polygons**: Self-crossing, index, selection

Ball morphs

MOD	**Ball morph of PCCs:** Definition, computation, implementation
MOD	**Ball morph of polygons:** Definition, computation, implementation

Inflation

MOD	**Distance transform:** Definition, computation, implementation, application
MOD	**Regularity transform:** Definition, computation, implementation, application
MOD	**Inflation:** From medial axis transform

Topology

Mathematical concepts

MOD	**Point set definition**: Concept, notation, motivation
MOD	**Booleans**: Union, intersection, difference, XOR, properties
MOD	**Interior, boundary, exterior**: Definition, examples, closure, open, closed, regularized (open/closed)

Polygons

MOD	**What is a polygon**: Examples, definition, options, loops, non-manifold
MOD	**Point-in-polygon test**: Ray stab parity, triangle inclusion parity, fast implementation, alternating difference, XOR
MOD	**Representation**: Loops, loop, edge soup, constructive, axis-aligned
MOD	**Self-crossing loop**: Examples, index, semantics, applications to Minkowski sums, offsets, drawings, animation
MOD	**Triangulation**: Motivation, definition, algorithms

Simplicial complexes

MOD	**Simplicial complexes**: Definition of simplices and complexes, operators, applications

3D geometry

Points and vectors in 3D

LIN	**Cross-product (3D)**: definition, notation, computation, properties, use in constructions
MOD	**Angles**: computing the angle between two vectors, rotating a vector
MOD	**Special alignment**: testing whether two vectors are parallel or orthogonal
LIN	**Decomposition**: tangent/normal decomposition of a vector and applications to reflections
LIN	**Reflection**: Formula, derivation, implementation, applications
MOD	**Mixed product**: Formula, intuition, properties, applications to signed volume, flatness

Intersections and containment tests in 3D

MOD	**Ray/triangle and edge/triangle intersection tests**: Formula, intuition, implementation
MOD	**Point-in-tetrahedron test**: Formula, intuition, implementation
MOD	**Tet-in-tet test**: Formula, intuition, implementation
MOD	**Tet/tet intersection test**: Formula, intuition, justification, implementation
MOD	**Tet/sphere & sphere/tet containment test**: Formula, intuition, justification, implementation

Frames and transforms in 3D

LIN	**Orthonormal frame**: definition, orientation, matrix, construction from I, from z, , composition, inverse
LIN	**Quaternions**: formulation, rotation, composition, implementation
LIN	**Affinity**: definition, matrix, computation from 4 point displacements, properties

Motions in 3D

MOD	**Direction interpolation**: linear, minimal angle rotation
MOD	**Rotation**: around given axis
MOD	**Screw**: definition, computation, applications
MOD	**SAM in 3D**: definition, computation, solvability
MOD	**Patterns**: definition, patterns of patterns, animation
MOD	**bi-SAM**: parabolic interpolant, Bezier evaluation, implementation, results
MOD	**Poly-SAM subdivision**: definition, computation, application

Curves and surfaces

MOD	**Curves in 3D**: lifting definitions and construction from 2D schemes, topology (tubes, tori)
MOD	**Tensor-product surfaces**: Bezier, Neville, NUBS, mixed,
MOD	**Surface tangent & normals**: computing parametric tangents, computing surface normals
MOD	**Normals under non-similar transforms**: problem statement, motivation, solution

3D graphics basics

Basics graphics and user interaction in 3D

GRA	**Renderer**: invoke P3D, coordinates, transforms, primitives
GRA	**Primitives**: sphere, triangle, quad (bilinear and triangulated), block, cone, cylinder
GRA	**Light and surface properties**: position, color, ambient, Lambertian, specular

Basics graphics and user interaction in 3D

GUI	**Pick surface point**: 3D pick, feedback, applications
GUI	**Screen coordinates**: screen aligned coordinates, obtaining them
GUI	**Drag point in 3D**: 3D pick, feedback, screen coordinates
GRA	**Rotate view**: set view, rotate view around fixed point
GRA	**Rotate view around picked point**: formulation, implementation

3D triangle mesh processing

Topology

MOD	**Water tight meshes**: definition, properties, detection
MOD	**Manifold**: non-manifold edges and vertices, detection
MOD	**Genus and holes**: definitions, Euler-Poincaré equation for manifold shell
MOD	**Orientation**: definition, encoding in order of vertex references, computation
MOD	**Shells**: definition, identification (swirl), topological classification, containment tree, solid parts

Normals and shading

MOD **Normals**: triangle normals, area weighted, vertex normals
MOD **Normals at border and sharp edges**: border vertices, sharp edges, per-corner normals
MOD **Gouraud shading**: Definition, implementation, benefits, limitation
MOD **Phong shading**: Definition, implementation, limitations

Traversal

MOD **Silhouettes**: Definition, proof of loop, tracing, subdivision for rendering
MOD **Geodesic walk**: Definition, exit edge, new direction
MOD **Geodesic walk**: Definition, exit edge, new direction
MOD **Geodesic path**: Definition, computation, acceleration

Smoothing and subdivision

MOD **Smoothing**: Dual, tuck, tuck-untuck, Beltrami-Laplace
MOD **Filling holes**: Build fan, construct center
MOD **Subdivision**: 1-to-4 split, implementation, butterfly, Loop, adaptive subdivision

Simplification and Levels-of-Detail

MOD **Vertex clustering**: Quantization, degenerate triangles, degenerate edges, implementation, properties
MOD **Edge collapse**: Implementation, error measures: normal displacement, minimal, quadric
MOD **Simplification strategies**: greedy, optimal vertex placement, memory-less
MOD **Levels of Detail**: Static vs progressive, stream of v-splits/e-col, adaptive
MOD **Multi-resolution representations**: simplicial complex, Quad-tree

Geometry/texture compression

MOD **Coordinate quantization**: Application requirements, non-uniform quantization,
MOD **Geometry prediction**: Quad predictor, more advanced predictors
MOD **Correction encoding**: Quantized correction, variable length encoding, Huffman tree, entropy
MOD **Normal compression**: Normal representation, quantization, prediction
MOD **Texture compression**: Texture representation, quantization, prediction

Offsetting

MOD **Normal offsetting**: Vertex displacement, self-crossing, chamfering
MOD **Volume compensation**: Adjust displacement, curvature computation, applications
MOD **Constant flux**: local formula

Intersections and Booleans

MOD **Point-inclusion test**: ray-shooting, tet-inclusion parity, robustness, speed
MOD **Interference test**: edge-triangle pairs, co-planarity
MOD **Intersection loops**: labels, connection, tracing, splitting triangles, approximation
MOD **Booleans**: propagation vs testing, incremental, regularization
MOD **Numeric accuracy**: tolerance, extended precision rationals, determinant tests, SoS, general position
MOD **Self-crossing test**: edge-triangle pairs, why they are insufficient

3D model creation

Acquisition

MOD **3D scans**: Depth scanners (time of flight, triangulation), z-buffers
MOD **Surface meshing**: Triangulation vs z-buffer merging

Drawing contours

MOD **Contour inflation**: MAT, elevation model, Teddy, Teddy,

MOD	**Redraw silhouette**: Project, deform, RoI, smoothing

Constructive models

MOD	**Extrusions**: contours (polygon, PCC), linear, circular, morph
MOD	**Booleans**: primitive, transformations, CSG, parameterization, constraints

Skinning

MOD	**Canal surface**: Definition, 4D curve subdivision
MOD	**Spine**: Frénét frame, twist propagation, local coordinates, perfect loop
MOD	**Pipe skinning**: Quad-rings, checkerboard pattern
MOD	**Blending spheres**: contact rings, Dupin cyclide, interpolation, Basti11,

Interpolating points and curves

MOD	**Radial Basis Function (RBF)**: Definition, computation, example
MOD	**Curve networks**: Normals, patches, interpolation

3D deformations and morphs

Triangle mesh deformations

MOD	**Vertrex displacement**: Pick, drag in 3D, along normal
MOD	**Region of influence**: Decay function,
MOD	**Twister**: Frame grab and move, 3D GUI, screw, adaptive tessellation
MOD	**Twister two hands**: Overlap RoI, squashing, cos and sin squares,
MOD	**Bender**: Ribbon, PCC, twist distribution closest projection
MOD	**Silhouette**: Select, drag, apply

Triangle mesh morphing

MOD	**Correspondence**: One-to-one, partial, closest projection
MOD	**Registration**: Rigid, similar, affine, non-rigid
ANI	**Linear**: Two key-frames, more key-frames
ANI	**Ball morph**: Ball correspondence, circular trajectories, synchronization, incompatible areas
ANI	**Minkowski morph of convex shapes**: Normal correspondence, T-V and E-E pairs, algorithm, representation, multiple keyframes
ANI	**Minkowski morph of general shapes**: Self-crossing, point containment

3D rendering

Perception

PSY	**Color**: Tristimulus, RGB, HLS
PSY	**Perception**: Eye, cones/rods, foevia/periphery, acuity, motion, color separation/recognition
PSY	**Illusions**: Shape, color, size, motion,
PSY	**Aesthetics**: Gestalt, Prägnanz, complexity, steadiness, grace

Light and photorealistic rendering

PHY	**Light**: Photons, wave, speed, linear propagation, measure, dimension
PHY	**Photon paths**: Snell's law of refraction, Fermat's principle of reflection, wiki,
PHY	**Optics**: Pinhole camera, spherical lenses, focus
PHY	**Mirrors**: virtual images, multiple reflections, retroreflector, wiki,
PHY	**Specular reflection**: Highlight, spread, models (Phong, Cook-Torrance), dependence on distance, wiki,
PHY	**Diffuse reflection**: Lambertian surface, independence of viewer location
GRA	**BRDF**: general formulation, special cases

GRA	**Shadows**: Filler rays, shadow buffer
GRA	**Ray tracing**: from light source, from eye, primary highlights, secondary highlights
GRA	**Area light sources**: umbra/penumbra, Minkowski sums, sampling, curved boundaries
GRA	**Radiosity**: Justification, mutual transfer between two patches, solvers

Non-Photorealistic Rendering (NPR)

PSY	**Silhouettes**: Importance, definition, classification, rendering
PSY	**Characteristic lines**: Definitions, computation
PSY	**Hatching**: Models, computation, coherence during motion and deformation

Volumes

Voxels

MOD	**Voxel models**: binary (in, out, partial)
MOD	**Rasterize triangle meshes**: regular sampling, robust point-in-solid, speed-ups
MOD	**Compact representations**: run-length encoding, compression

Distance fields

MOD	**Definition:** point-mesh-distance, sign, point-surface distance, sharp features
MOD	**Computation:** exact, propagation, sweeps

Iso-surfaces

MOD	**Binary:** face cutting, consistency (the Marching Cube mistake), cube splitting

GPU

Graphics pipeline

GRA	**Transforms**: model space, screen space
GRA	**Lighting**: ambient, Lambertian, specular, finite/infinite light
GRA	**Clipping**: viewing frustum, polygon clipping
GRA	**Perspective**: projection, properties
GRA	**Rasterization**: drawing lines, filling triangles
GRA	**Texture**: texture coordinates, blending, assignments, procedural, distortion, correction
GRA	**Z-buffer**: principle, implementation, linear interpolation, accuracy issues, turning off
GRA	**Visibility**: depth order, correct perspective transform, homogeneous coordinates, inverse perspective

Shaders

GRA	**Programmable shaders**: Architecture, shaders, parallelization, languages
GRA	**Vertex shader**: function, input/output, coordinates, matrices, normals, effects
GRA	**Fragment shader**: function, input/output, coordinates, texture mapping, normals, Phong shading, bump mapping
GRA	**Image processing**: filters, blending images
GRA	**Geometry & tessellation shaders**: Bezier curves, subdivision surface, normals, silhouettes