1 Transformations and Projections

Basic 2D Transforms Scale, rotation, shear and reflection can be represented as 2x2 matrices. Transform matrices can be combined through multiplication. Order matters, transformations are applied from right to left.

Translations in 2D Translations cannot be modeled with a 2x2 matrix, need a 3x3 matrix (at least). Other transformation matrices can be extended to 3x3 and then combined as before — all 2D transformations can be represented by a single 3x3 matrix.

Viewport Transform M_{vp}. 1. Translate to origin 2. Scale to new dimensions 3. Translate to new position

Basic 3D Transforms Same thing as 2D — scale, rotation, shear and reflection can be modeled as 3x3 matrices, translations requires a 4x4 matrix.

Transforming Normal Vectors Normal vectors describe directions, not positions, so they can't be transformed with the previous methods because they might not remain normal. For a given transform M the normal vector transform is $N = (M^{-1})^T$.

Note that the last coordinate (the homogenous coordinate) in a vector controls whether or not a translation is done: zero corresponds to no translation. We set it to zero for direction vectors.

Inverse Transformations can be computed numerically. But usually we know the components of a matrix composition, in this case we reverse the order of multiplication and use the inverses of the matrices.

Orthographic Projection 1. View transformation M_v to map camera coordinate system to world coordinate system. 2. Orthographic projection M_o from user-defined scene volume (orthographic view volume) into normalized view volume (canonical view volume). 3. Viewport transform M_{vp}.

$p_s = M_{vp} M_o M_v v_{obj}$

Perspective Projection Size of an object on the screen is proportional to its distance. Vertices are projected toward camera center and appear where their line of sight intersects the viewing plane (which is some distance away from the center).

A perspective transform M_p maps these lines to ones that are parallel to the z axis while maintaining their intersection point with the viewing plane. The perspective transform also sets the homogenous coordinate to z, encoding how much the other coordinates must be scaled after the transform (homogenization or perspective division).

$M_{pp} = M_o M_p$

$p_s = M_{vp}(M_{pp} M_v v_{obj})/h$

2 Raster Algorithms and Depth Handling

Binary Space Partition Trees allow ordering of surfaces for depth handling. They work only if no surface crosses the plane of another surface. Pre-processing required: If a triangle passes through a plane of another triangle, cur it into multiple triangles to satisfy the condition.

Generalization: Pick a reference triangle as root. Build tree structure by dividing other triangles into groups based on which side of the triangles plane they are on. To ensure consistency, use camera position as reference point for deciding the side.

BSP-tree is good for static sceneries in games but inefficient for dynamic scenes.

Z-Buffer is well suited for dynamic scenes and has hardware support. For each pixel, a depth value is stored. The previously unused z components of vertices after projection are used for this.

3 Local Shading and Illumination

Lambertian Shading only considers the angle between light source and surfaces. This only applies for very (infinitely) distant light sources. In reality, light intensity drops off quadratically as the distance grows (square-distance attenuation).

Another problem is that surfaces pointing away from light sources appear black. Solution: Approximate constant environment light by averaging the color of all surfaces and introduce it as ambient shading term.

Flat Shading computes shading c for each triangle and applies it to every pixel during rasterization. Many triangles are needed for smooth appearance, which causes flat shading to perform badly. Gourad Shading has a similar idea but again performs badly with large geometric resolutions.

Phong Shading approximates specular highlights by also considering the viewing direction e and the natural direction of light reflection r (angle between in and normal = angle between normal and out). The specular highlight is brightest when $e = r$ falls off gradually as the delta angle increases.

To counteract unrealistically wide specular highlights the Phong exponent p is used. Small p results in wide specular highlights, large p in small highlights.

To avoid problems with negative values, there is a heuristic approximation for computing r: Get vector that is halfway between light and eye.

Phong Normal Interpolation Regular Phong shading still has problems with performance for high geometric resolution. Instead of interpolating colors, interpolate normal vectors (during rasterization) and apply the shading model for each rastered pixel.

BRDF Theory Describes material properties, how light is reflected at one surface point. Input and output directions are dual (Helmholtz reciprocity). Does not describe subsurface scattering (BSSRDF) or transmission of light (BTDF).

4 Texture Mapping (Basics and Advanced)

Mapping a 2D texture to a 2D parameterizable surface (e.g. sphere) is easy. To map onto a tessellated surface, determine texture coordinate for each vertex through biliniar interpolation via barycentric coordinates: Texture mapping is part of rasterization stage.

Perspective Correct Textures Interpolating texture coords in screen space results in incorrect perspectives. For correct perspective we want to interpolate before the perspective division (unlike with other parameters). The key observation is that the homogenous coordinate (used for the perspective division) is linear in screen space (can be interpolated without error).

Bump Maps „Bumpy“ surface normals are stored together with the surface refletance and mapped to the surface geometr in the same way as textures. These normals are then used for lighting during rasterization.

Displacement Map 2D maps that store small displacement offsets along the surface normal. Are mapped onto the surface geometry, but instead of alterering the reflectance they change the geometry itself.

Cube Maps are a special form of environment map (which allow simulating specular reflections of backgrounds, not only of light sources as seen with Phong shading). Basically a 360° photo applied to the inside of a cube. Reflections for an environment map are only correct for the single 3D point that was used for generating them.

In case a cube map is actually rendered centered to the viewport it is called a skybox.

Projective Texture Maps Texture coordinates are of vertex coordinates are computed such that they map to texels that are projected from a given perspective.

This is particularly useful for casting hard shadows, where the perspective of the light source is rendered first, to calculate the distance of all visible points to the light source. Then the scene is rendered from the perspective of the camera. If a point is closer to the light source than it is to the camera, it is in shadow.

MIP Mapping For each texture, multiple resolution versions are pre-computed. Different textures are loaded based on camera distance.

Billboarding Textured polygons that face the camera. Combine alpha texturing with animations. Geometry is transformed in 3D while the texture is updated, which can contain pre-rendered images of more complex 3D scenes.

A screen aligned billboard is called full-screen billboard.

Impostor Billboard that re-renders its texture depending on the estimated distortion over time. To re-render, the virtual camera points to the center of the corresponding bounding box. Only work well if object and viewpoint don’t change much relative to each other (frame-to-frame coherence). Best used if distance between camera and object is large.

Particle Systems Set of small objects that are animatred using some algorithm (smoke, fire, water, . . .). Billboards can be used for efficient representation. Hard particles are regular flat billboards (seam between particle and other objects). Soft particles are computed in the fragment shader (fading as they get closer to other geometry).

Glare Effects are rendered with impostors or billboards, consist of halo (light refraction) and ciliary corona (density fluctuation in lens).

Irradiance Maps Normal environment mapping allows simulating perfectly shiny surfaces. Textures of an environment map can be pre-filtered to simulate imperfect shininess. Irradiance mapping computes reflection directions and weights for each point (using the Phong equation) for all possible directions and sums up the weighted contributions per direction.

Light Maps are used for static scenes and objects, pre-computed illumination of surfaces. Static relationship of light and scene object is assumed.

5 Graphics Pipelines

Culling Triangles which are completely outside the viewing volume or which are backfacing are removed (culled). This only works for closed surfaces. (Can be very expensive in practice if carried out for individual triangles, culling bounding volumes are more efficient.)

Clipping If frontfacing triangles intersect the viewing volume they are clipped. Carried out after projection, using clip coordinates (where they get their name from).