First Principles & Practices of Projection Mapping

tom@tommarks.tv

Units & Metrics

Lumens	in	lm
Resolution - Res	in	(x)px (y)px
Canvas Area - CA	in	m^2
Pixel Loss - PL	as	%
Pixel Width - PW	in	mm
Projected Area - PA	in	m^2
Horizontal Resolution - xres	in	(x)px
Vertical Resolution - yres	In	(y)px
Throw Ratio - TR	as	TR:1
Field of View - FOV	in	x°
Margin of Error - MOE	in	(x)mm (y)mm (z)mm
Average Point of View	as	V

Mathematics

Trigonometry

Pythagorean Theorem

$$c^2 = \sqrt{a^2 + b^2}$$

Trigonometric Functions

$$sin\theta = \frac{opp}{hyp}$$

$$cos\theta = \frac{adj}{hyp}$$

$$tan\theta = \frac{opp}{adj}$$

Inverse Trigonometric Functions

$$\theta = sin^{-1}\frac{opp}{hyp} = arcsin\frac{opp}{hyp}$$

$$\theta = cos^{-1}\frac{adj}{hyp} = arccos\frac{adj}{hyp}$$

$$\theta = tan^{-1}\frac{opp}{adj} = arctan\frac{opp}{adj}$$

Laws of Sine

$$\frac{a}{sin\ \alpha} = \frac{b}{sin\ \beta} = \frac{c}{sin\ \gamma} = 2R$$

$$\frac{sin\ \alpha}{a} = \frac{sin\ \beta}{b} = \frac{sin\ \gamma}{c}$$

Projection Mapping Equations

Illumination of Canvas Area

$$I_c = \left(\frac{Lm}{PA}\right) \times (1 - PL)$$

Illumination of the Canvas is equal to the brightness of the total projected area divided by the Canvas Area, minus the Pixel Loss.

The term projected area is misleading as it requires a spherical canvas with a fixed radius

Pixel Width

$$PW = \left(\frac{d}{TR}\right) \div XRES$$

Pixel Width is equal to Distance divided by Throw Ratio, divided by the Horizontal Resolution

Pixel Width with Pixel Loss

$$PW = \left[\left(\frac{d}{TR}\right) \div XRES\right] \times (1 - PL)$$

Pixel Loss

$$PL = 1 - \frac{PA - CA}{PA}$$

Pixel Loss is equal to 1 minus Projected Area minus Canvas Area, divided by Projected Area.

Void Area

$$VA = PA - CA$$

Void Area is equal to Projected Area minus Canvas Area.

Throw Ratio

$$TR = TD \div IW$$

Throw Ratio is equal to Throw Distance divided by Image Width

Throw Distance

$TD = TR \times IW$

Throw Distance is equal to Throw Ratio multiplied by Image Width

Image Width

$IW = TD \div TR$

Image Width is equal to Throw Distance divided by Throw Ratio

Field of View

$FOV_x° = \frac{XRES}{YRES} \times FOV_y°$

...

$FOV_y° = \frac{YRES}{XRES} \times FOV_X°$

...

$FOV = 2 \times Tan^{-1}(\frac{0.5}{TR})$

Derivations of equations

Perspective, Parallax, and Scale

https://en.wikipedia.org/wiki/Category:Units_of_photometry

https://en.wikipedia.org/wiki/Perspective_(geometry)

https://en.wikipedia.org/wiki/Parallax

https://en.wikipedia.org/wiki/Inverse-square_law

https://led-ld.nichia.co.jp/api/data/spec/tech/ApplicationNote_SE-AP00041-E.pdf

https://www.translatorscafe.com/unit-converter/en-US/illumination/#05
https://www.electrical4u.com/radiant-flux-and-luminous-flux/

Brightness

Lumens

Lumens or (lm) is the SI-based unit that measures luminous flux, in other words the total amount of light produced by a light source per unit of time.

Radiant Flux

Luminous flux as a measurement is different than radiant flux because luminous flux measures only the electromagnetic waves that the human eye can see while radiant flux measures all electromagnetic waves emitted by a source.

Lux is similar to nits and ft-L in that it also measures brightness in terms of surface area, but in this case, it is defined as lumen per square meter.

Foot-Lamberts or ft-L

Foot-Lamberts or ft-L, takes the same unit of light used in defining nits, the candela, but applies it to the unit of area customary to the United States, square feet, meaning that 1 ft-L is equal to 1 candela per square foot.

Nits

Nits is a unit that measures brightness in terms of area, or in technical terms, candela (a standard unit of light equivalent to the light produced by a single candle) per square meter.

ANSI Lumens

ANSI Lumens is a unit, defined by the American National Standards Institute (ANSI), that measures the overall amount of light output by a projector, in other words the higher the lumen value for a projector the brighter the light it produces.

SOFTWARE

Useful list of software

https://projection-mapping.org/software/

Blender

Plugins

Spout

Spout tfo

WORKFLOWS

https://lucid.app/lucidchart/0353cd94-3eb6-4761-b2eb-c7b94db0a273/edit?invitationId=inv_0944e41a-949c-401a-856f-9db5fd9a5c11

Questions to ask before starting a projection mapping project.

What is the canvas?

- Geometry
- Materials
    - Albedo/Screen Gain
    - Colour
-

Where are the audience domains?

Where are the projector domains?

What are the constraints?

Throw distance, occlusions, traffic, ambient light,

Unrefined notes:
Diffusion to Projection Ratio, e.i. What percentage of the projected pixels diffuse on the canvas?

Method: Using ImageJ to calculate the mean average brightness of a rendered image of the 3D model of canvas from projectors point of view.

Diffusion-to-Projection Ratio = White Pixels/Total Pixels = Mean Average Brightness

AKA Pixel Loss

E.g. (100/255) * 100 = 39%

Per projector then mean average

$$PL = PL_1 + PL_2 + ... \div n$$

Diffusion to occlusion(VIP POV) ratio, e.i. What percentage of the diffused light is not occluded relative to the V

White texture

And for that matter what is

THINGS TO AVOID AND HOW TO AVOID THEM

Real shadows on canvas

Where there's a projector shadow on the surface the audience can see is where the illusion breaks down. To avoid this make sure the projectors have line of sight to everything your audience does.

Pixel Stretching (perpendicular angles)

Any surface that is not perfectly perpendicular to the projector will have some amount of distortion. Perfectly parallel surfaces have, in theory, have no diffusion, reflection, or refraction.
A plane at 45 degrees...

Mismatching Brightness

Physics questions to answer

1. 2D, Infinite distance (sun lamp), rotating a plane - What's the total brightness?
2. 2D, Infinite distance (sun lamp), rotating a square - What's the total brightness?
3. 3D, Infinite distance (sun lamp), rotating a cube - What's the total brightness?
4. 3D, Finite distance (point lamp), rotating a cube - What's the total brightness?

https://www.youtube.com/watch?v=ltLUadnCyi0
Build demo, get sample data, plot, model

Mismatching UV Projections

With multiple projectors, a 3D element will not inherit the same perspective if multiple projector-cameras are used as it will be duplicated and appear in multiple places from multiple perspectives.

1. Just a plane and 2x 45 degree projectors at equal distance
2. Math for parallax?

Mismatching Resolution & Pixel Pitch

If you're viewing the projections from the same vantage point as a single projector this doesn't apply, however this is almost never the case.

Think of each pixel as an individual projected source. If we stretch this over

No object but shadows are there and light is moving revealing the shadows

Latency

This is where projectors are out of sync with each other.

Unrefined Notes

3D Animation

Got an idea to have a statue get up and walk away relative to the audiences perspective, but to do this I'd need to render twice (well, technically five times).

Animate statue, render from audience perspective, reproject, capture from projectors POV, render with final projections but only on canvas, composite into scene, render.

Challenging part would be incorporating multiple audience members POVs, even a slight translation (at this scale) would break the illusion.

Plato's Cave —- PLATO'S RAVE

Sun is tracked and replicated on projected texture during the afternoon, thus selling the illusion more as the projected light source is the same as real life.

Platonic solids and the best way to light them with different numbers of projectors and constraints, with the maths included.

Types of Shadows

Real Shadow

Parts of the canvas that can’t be illuminated by any projector are considered real shadows.

Virtual Shadow

Parts of the canvas that can be illuminated by any projector but are not for the sake of creativity or shadow elimination.

Soft Shadow

Parts of the canvas that is illuminated by only some projectors are considered soft shadows.

Hard Shadow

Shadows that are cast by the canvas itself, usually resulting in a noticeable line between different illumination levels.

Auto-Alignment of Projection Mapping

Realigns virtual scene to match reality

How does a human do it?

Camera transformation matrix

Obvious vertices of the virtual model (e.g. corners) are re-aligned by projecting crosshairs onto the real canvas. One projector at a time. Basically the Structure from the Motion step of photogrammetry but with manual feature matching. Also see: CamSchapper.

Auto Shadow Elimination

Imagine a simple scene with two projectors

Every part of the canvas the audience can see should have a line of sight with one or more projectors.

When two projectors combine on part of the canvas it's the brightest, but when compared to just one there's a real partial shadow.

To compensate, the parts of the image that's can see both projectors

Pixel Stretching in Touch Designer

External Links
Spreadsheets
Flowcharts