ZH3: QUADRATIC ZONAL HARMONICS

THOMAS ROUGHTON, ARI SILVENNOINEN,
PETER-PIKE SLOAN, MICHAL IWANICKI,
PETER SHIRLEY

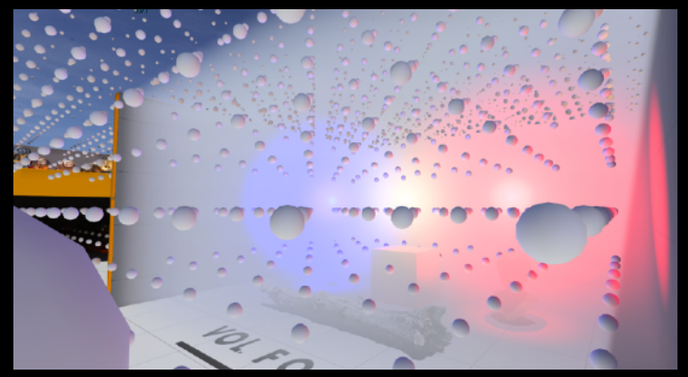
ACTIVISION
CENTRAL TECH

Today, we’re going to be talking about a new volumetric lighting format that sits somewhere between linear and quadratic SH in quality, but much closer to linear SH for storage requirements.

ENVIRONMENT LIGHTING

ACTIVISION CENTRAL TECH

- Compact representations of incident lighting at a point.
- Examples:
  - Spherical harmonics [RH01, Green03]
  - Spherical Gaussians [GKMD06, NP15]
  - Ambient Cube [McTaggart04]
  - Ambient Dice [IS17]
- Some function where you can query for the lighting in a direction.
- Usually stored volumetrically.



When we talk about volumetric lighting, we're referring generally to compact representations that encode the radiance or irradiance at a point. These compact representations, such as spherical harmonics, spherical Gaussians, the Ambient Cube, Ambient Dice, to name a few, are usually stored in some sort of grid or spatial data, so you can query the irradiance for any normal direction at any point in the world.

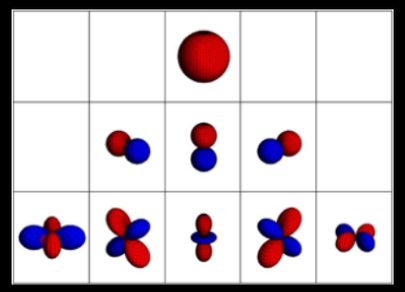
SPHERICAL HARMONICS

ACTIVISION CENTRAL TECH

- Bands based on the degree of polynomial.
  - Represent increasingly high frequency lighting.
  - f_l^m means basis function m in band l.
- Orthonormal over the sphere, i.e.

$$\int_S A(s)B(s)\mathrm{d}s = \begin{cases} 1 & \text{if } A = B \\ 0 & \text{otherwise} \end{cases}.$$

- Rotationally invariant.
- Easily convolved to compute irradiance:



SH bands through SH3 [Sloan08]

- If SH coefficients r represent radiance such that $\mathrm{Y}(s) \cdot r \approx \mathrm{Radiance}(s)$, to get i such that $\mathrm{Y}(s) \cdot i \approx \mathrm{Irradiance}(s)$ you simply multiply the bands of r by $\left[1 \quad \frac{2}{3} \quad \frac{1}{4} \quad \ldots\right]$
- This is convolution with a zonal harmonic (a normalized cosine lobe).

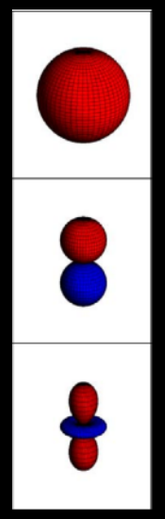
For this talk we're focusing in on spherical harmonics. If you're not familiar, spherical harmonics are a family of spherical basis functions – meaning functions you can evaluate in 3D directions - grouped into bands, where each band has polynomials of increasing order. The L0 band, also called DC, is just the average value of the lighting times a normalisation constant; the L1, or linear, band, adds a linear gradient of lighting, and then increasing orders add increasingly high-frequency detail.

SH are orthonormal over the sphere, which just means that the integral of the product of any two SH basis functions over the sphere is one if they're the same function and zero otherwise. They're also rotationally invariant, which essentially means that rotating the data and then encoding into SH will give you the same result as encoding into SH and then rotating the SH.

An important property of SH that makes them useful for lighting is that they can be efficiently convolved with rotationally symmetric BRDFs, such as the normalised cosine lobe for irradiance. This is due to a property of a special family of spherical harmonics known as the *zonal harmonics*, and it's that family of SH that we're going to focus on today.

ZONAL SPHERICAL HARMONICS

- The m = 0 functions are known as *zonal* harmonics:
  - $\frac{1}{2\sqrt{\pi}}, \sqrt{\frac{3}{4\pi}}z, \sqrt{\frac{5}{16\pi}}(3z^2-1), \sqrt{\frac{7}{16\pi}}(5z^3-3z), \ldots$
- Any radially symmetric function is a zonal harmonic (in a coordinate frame oriented along its axis):
  - Sphere lights, hemispheres, cones, cosine lobes.
  - Linear SH are radially symmetric around their "optimal linear direction" axis $\frac{[-l_1^1, -l_1^{-1}, l_1^0]}{\|l_1\|}$; that means any linear SH is a zonal harmonic!
- Convolution with a zonal harmonic in that ZH's coordinate frame is always a per-band scale [SLS05].



[Sloan08]

The zonal harmonics are the subset of spherical harmonics for which m = 0, which simply means that in Cartesian coordinates they're polynomials in z only. Any radially symmetric function is a zonal harmonic, and crucially that includes any linear SH. You can think of the L1 SH band as being a 3D vector, and so if you construct a coordinate frame along the direction of that vector you're left with the vector length in the m = 0 basis function and the local x and y components being zero.

SH LIGHTING FOR GAMES

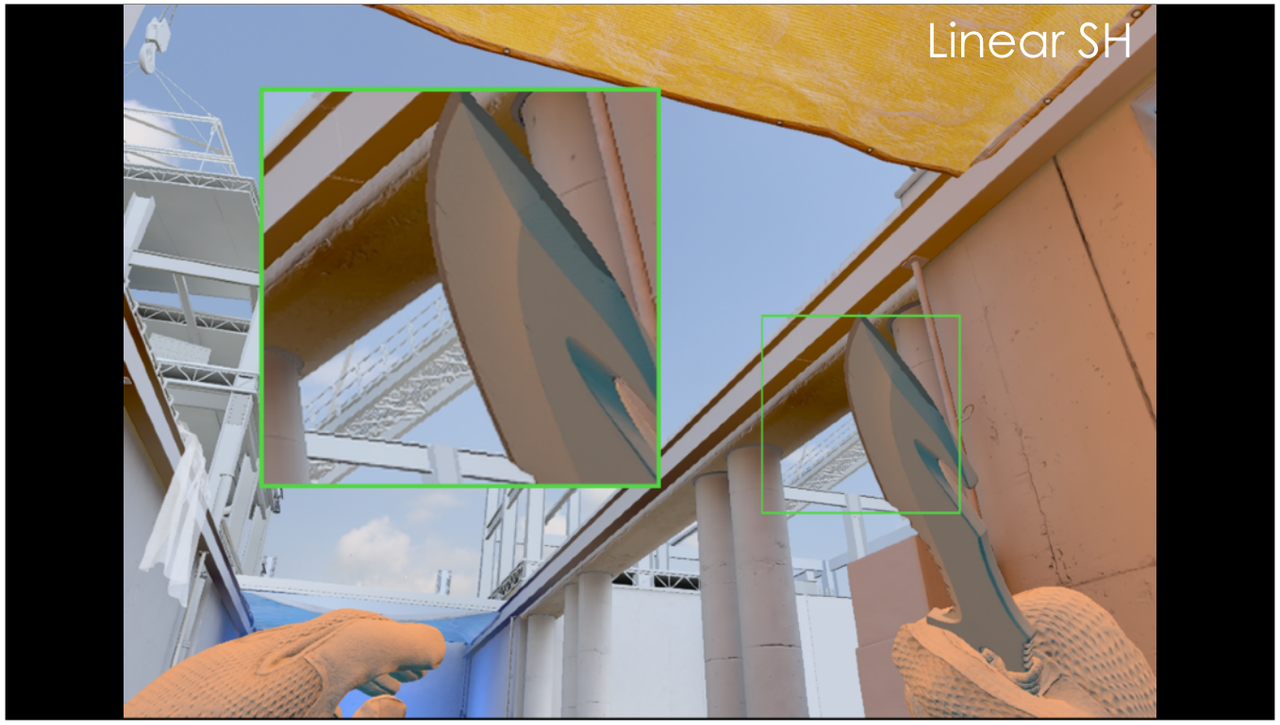
- Linear SH (SH2): four basis functions, four coefficients per channel.
  - Compact, efficient, not very accurate.

$$\mathrm{Y}(x,y,z) = \left[\frac{1}{2\sqrt{\pi}} \quad -\sqrt{\frac{3}{4\pi}}y \quad \sqrt{\frac{3}{4\pi}}z \quad -\sqrt{\frac{3}{4\pi}}x\right]$$

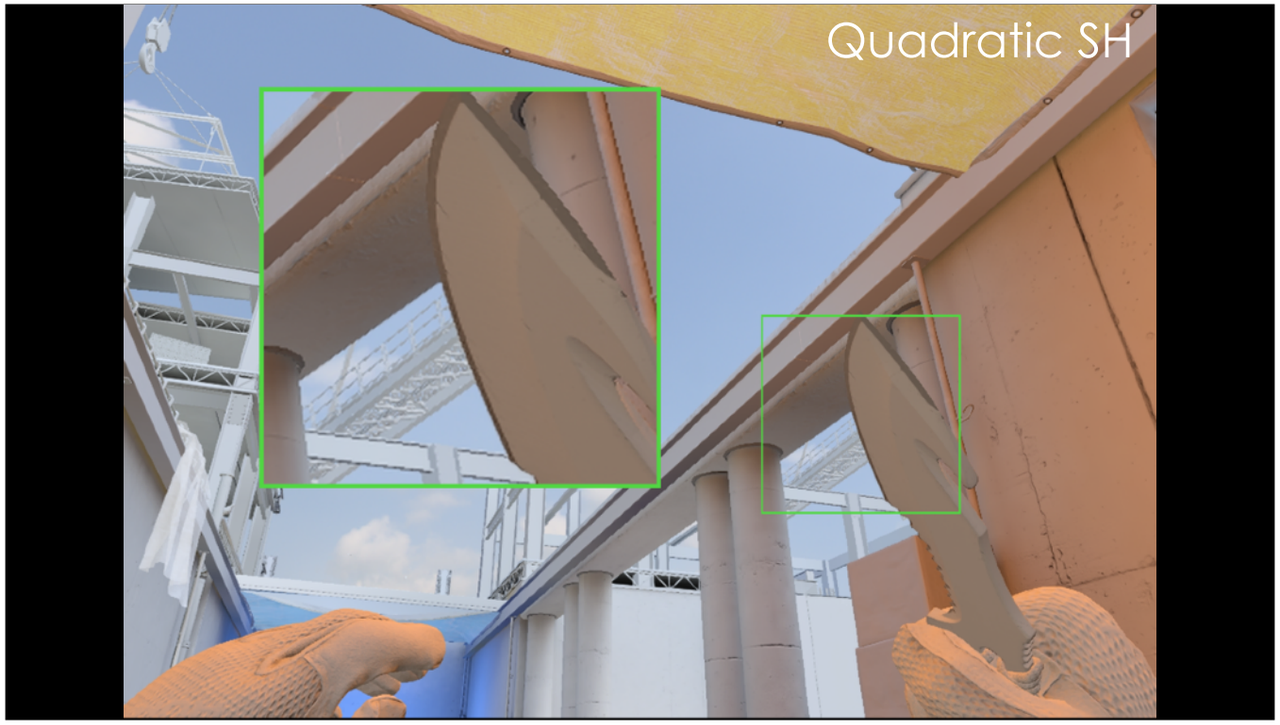
- Quadratic SH (SH3): nine basis functions, nine coefficients per channel.
  - Irradiance error of < 3% on average for environment maps [RH01].
  - Fairly heavy: 27 coefficients for RGB.

$$\mathrm{Y}_2(x,y,z) = \left[\sqrt{\frac{15}{4\pi}}xy \quad -\sqrt{\frac{15}{4\pi}}yz \quad \sqrt{\frac{5}{16\pi}}(3z^2-1) \quad -\sqrt{\frac{15}{4\pi}}xz \quad \sqrt{\frac{15}{16\pi}}(x^2-y^2)\right]$$

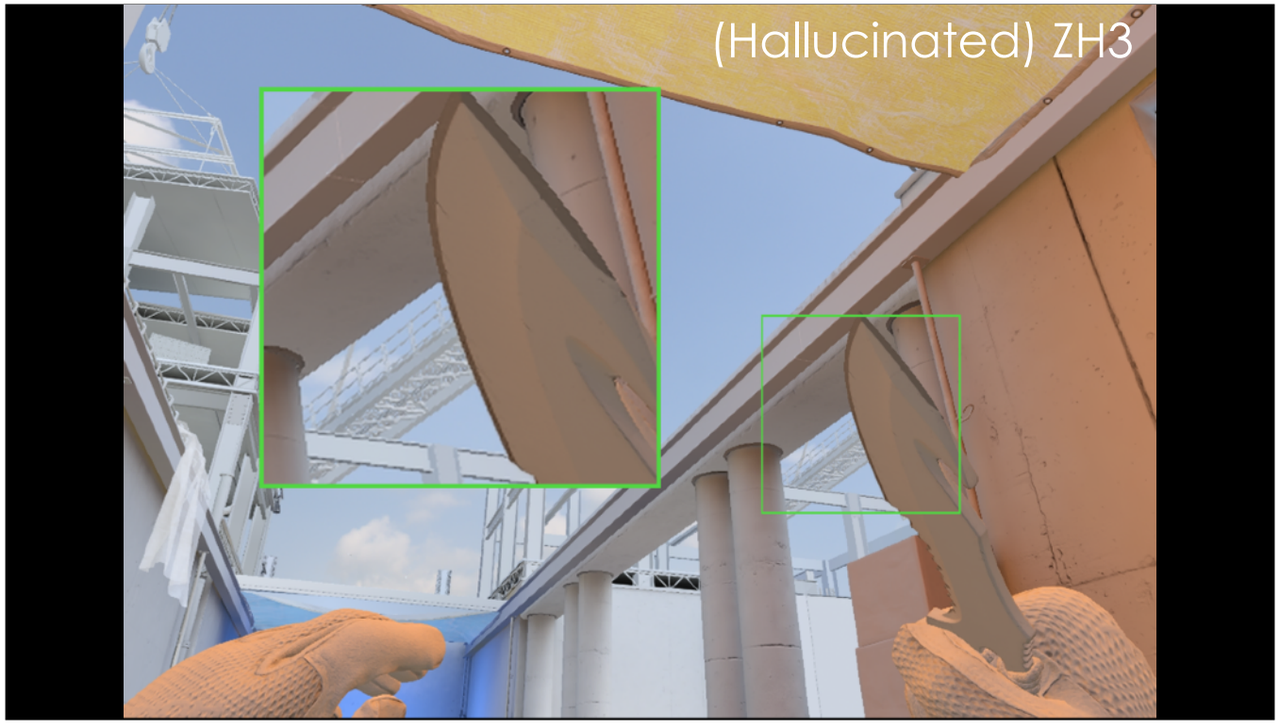
For spherical harmonic lighting in games, we're usually mainly concerned with irradiance, and want some sort of compact representation to store and evaluate. The two orders of SH that are of most interest are linear SH, which only has four basis functions, and quadratic SH, which has nine. Linear SH is very inexpensive, but is also of fairly poor quality; quadratic SH, by comparison, can generally store irradiance very accurately but also takes up a lot more memory.



Here’s a comparison on an in-game scene. On a scene lit with just linear SH we get a lot of ugly colour shifting and negative lobes, which you can see under the railing and on the knife. The tarpaulin is also pretty dark compared to what it should be.



SH3 is much cleaner; the colour shifting is gone, the tarpaulin is now properly lit from below, and the underside of the beam looks a lot more consistent with the rest of the lighting. However, we've more than doubled the storage, which can be a problem for constrained platforms like mobile. Ideally, we want a middle ground; something closer to the storage of linear SH but with visual quality closer to SH3.



Our solution, ZH3, is that middle ground. The image you're looking at here uses only one more basis function over linear SH but is visually much closer to SH3. As an extra trick, this is *hallucinated* ZH3, which means this is using no extra storage over linear SH. We'll get to the hallucination part in a bit, but first, let's talk about what ZH3 is and how it works.

ZH3

ACTIVISION CENTRAL TECH

- Add the zonal quadratic basis function to linear SH, oriented in the optimal linear direction.
  - Basically: what direction does the linear SH point in, and what's the quadratic SH band in that direction?
- Matches SH3 for all zonal light sources (directional, point, cone, sphere, etc.)
- ZH3 coefficient k_2 is given by a least-squares fit:
  - $k_2 = \frac{4\pi}{5} Y_2(\mathbf{d}) \cdot f_2$

  where f_2 is the quadratic SH band coefficients we're fitting to and $\mathbf{Y}_2(\mathbf{d})$ is the L2 SH band evaluated in the direction $\mathbf{d}$.
  - Hallucinated l_2 coefficient in the zonal frame is $l_2^0 = \sqrt{\frac{4\pi}{5}} Y_2(\mathbf{d}) \cdot f_2$

ZH3 extends linear SH by adding just the zonal basis function from quadratic SH. Having it be always oriented in Z wouldn't be very useful, so instead we orient it along the linear SH vector; that gives us a zonal harmonic in the linear SH's coordinate frame. We then compute, store, and evaluate coefficients for that oriented zonal basis function; the z in the ZH3 equation gets replaced with the projection or dot product of the direction being evaluated with the zonal axis.

ZH3 can exactly capture all SH3 zonal light sources, which means directional, point, cone, and sphere light sources all maintain the same accuracy as for SH3.

To compute the ZH3 coefficient, you just take the L2 band coefficients of the quadratic SH, dot it with the L2 basis functions evaluated in the zonal axis, then scale by 4pi/5. To reconstruct the SH3, you evaluate the L2 basis functions in the zonal axis and then multiply it by the ZH3 coefficients.

Here’s a comparison of ZH3 side by side with SH3 and SH2 on the Linz environment map. The lighting here is very zonal, so linear SH lacks a significant amount of detail. ZH3 has more contrast and definition and, although it’s not an exact match, generally looks a lot closer to SH3.

LUMINANCE AXIS

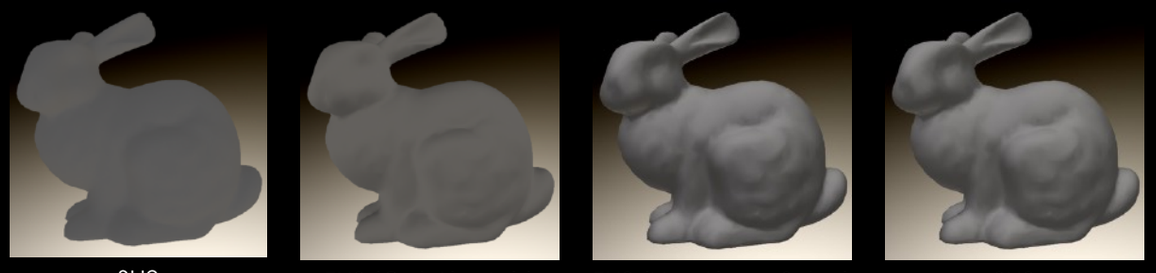
- ZH3 can suffer from color shifting.
- Solution: use the same zonal axis for all color channels.

ZH3 isn’t immune to colour shifting, however; sometimes, if the zonal axes are significantly different for each colour channel, the ZH3 basis functions interfere. Our solution for this is to use the same axis for all three channels, where the axis is given from a weighted average of the linear SH. Whether using a shared axis increases or decreases error depends on whether the quadratic band is more aligned with the shared axis or the per-channel axes; however, since colour fringing is much more objectionable than slightly increased error, using a shared axis usually produces better results.

SOLVING FOR THE AXIS

ACTIVISION CENTRAL TECH

- Sometimes the L2 (quadratic) band energy isn't well aligned with the L1 direction.
- Solution: search for linear SH that give us a better direction for ZH3.
  - Compromises linear SH accuracy but improves overall accuracy.



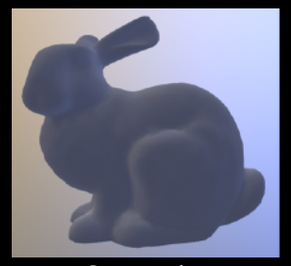
SH2 | ZH3 (Opt. Lin. Axis) | ZH3 (Non-Linear Solve) | SH3

Using the axis from the linear SH as our zonal axis isn't always the best choice; sometimes, most of the energy is in the L2 band in a different direction. In that case, it's worth solving for a new zonal axis and projecting both the linear SH and quadratic band onto that. In examples like this one, searching for the zonal axis can give dramatically better results than just using the linear SH. The solve is a non-linear optimisation using BFGS, and we plan to release our solver code in the near future.

When using a shared axis in the solve the choice of weighting becomes more important. We've experimented with using both equal weightings and luminance weightings; both are reasonable choices, but you do need to make sure the error function uses the same weights, since otherwise the solver will increase error in the lower-weighted channels. For example, if blue has a high weight on the axis but a lower weight for the error, the solver will push all the error into blue to get a better fit on the axis for the other channels, and this is very obvious visually.

HALLUCINATING COEFFICIENTS

- How far can we get without requiring *any* extra storage over linear SH?
- Prior work from Geomerics [Joseph15] introduced a method for higher-quality irradiance reconstruction from linear SH.
  - Geomerics uses the ratio of the linear band length to the constant/DC term to reconstruct guaranteed non-negative lighting.
  - Preserves the first moment (DC), but doesn't preserve the second (linear).
  - Tends to over-brighten the lighting [SS20].



Geomerics

Reference (MCIS)

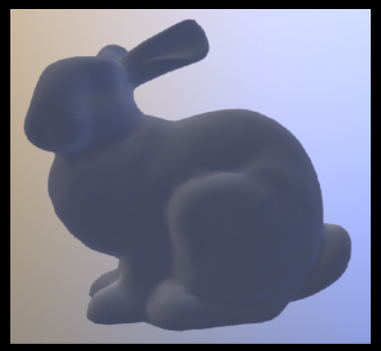
So that covers stored ZH3 coefficients. Let's shift focus to something I mentioned a bit earlier: *hallucinated* coefficients. Rather than adding more data, the idea behind hallucinated ZH3 is to achieve better reconstruction from linear SH by guessing what the ZH3 coefficients might have been if we had stored them. This lets you get better quality from linear SH essentially for free.

We're not the first to try to extract better irradiance reconstruction from linear SH. Geomerics is a fairly well-known algorithm that uses the ratio of the length of the linear band to the DC term to drive the reconstruction. Geomerics gives non-negative reconstruction, which is great, but it doesn't preserve the second moment of lighting – the linear SH gradient – and tends to overbrighten and wash out details.

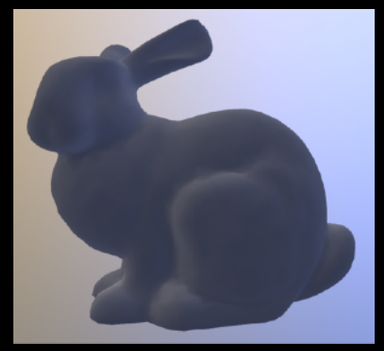
HALLUCINATING COEFFICIENTS

ACTIVISION CENTRAL TECH

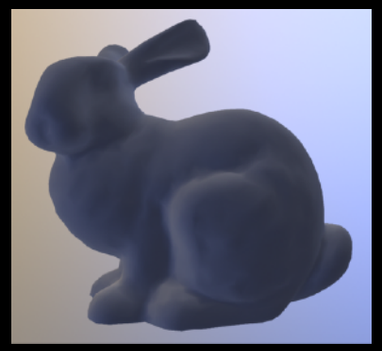
- Solution: use the ratio of linear to DC to *hallucinate* a ZH3 coefficient.



Geomerics



ZH3 (Hallucinate)



Reference (MCIS)

Using ZH3, on the other hand, we can get a result that looks much closer to the reference.

HALLUCINATING COEFFICIENTS

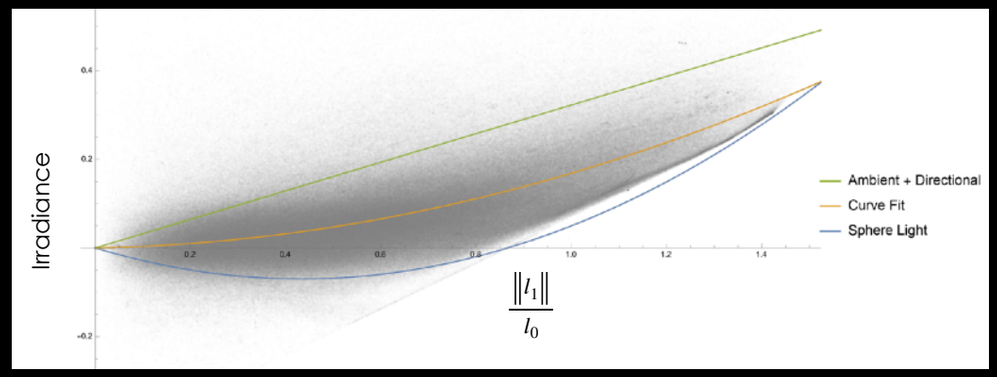
- A few different ways to interpret linear SH:
  - An ambient light plus a directional light: L1 band gives directional intensity and remainder in DC gives the ambient.
  - A sphere light/cone; you can infer the cone angle from the ratio [SA20].
  - Some combination of the above, with a fixed cone angle.
  - … or it could be something totally arbitrary.
- We evaluated the ambient + directional and sphere light models on production lighting data against linear SH.
  - Ambient + Directional: ~17% reduction in RMSE.
  - Sphere Light: ~0% reduction in RMSE.

There are a few different ways to handle the hallucination. There's an interesting idea here in that you can reconstruct any lighting setup that has the same number of degrees of freedom as linear SH – that is, four – from linear SH. For example, if you have an ambient light and a directional light, you've got the ambient intensity, directional intensity, and direction on the unit sphere; if you have a sphere light, you've got the direction to the light and angle of the cone formed.

We tried a couple of these models on a production data set of just under three million probes. Ambient + Directional worked well, with a over 17% reduction in error compared to just linear SH. The sphere light wasn't so successful; it looks better since it has higher-frequency lighting but is no more accurate than linear SH.

HALLUCINATING COEFFICIENTS

- Ambient + Directional and Sphere Light hallucinates are just lines/quadratics.
- A least-squares quadratic curve fit gives 23+% reduction in RMSE, vs. 17+% for ambient + directional.



If you look at these theoretical models more closely, you find that the hallucinated ZH3 terms are just lines and quadratics based on the ratio. Given that, we tried just doing a quadratic curve fit to our data set; compared with linear SH, that gave us a 23% reduction in error.

Note that this curve fit is for our lighting data, and the input SH3 have already undergone windowing and other processing, so it may be you can find a better fit for your own data sets. We did see a fairly consistent fit across multiple maps, however.

HALLUCINATING ZH3 COEFFICIENTS

ACTIVISION CENTRAL TECH

```
float3 SHHallucinateZH3Irradiance(float3 sh[4], float3 direction) {
    // Use the zonal axis from the luminance SH.
    const float3 lumCoeffs = float3(0.2126f, 0.7152f, 0.0722f); // sRGB luminance.
    float3 zonalAxis = normalize(float3(-dot(sh[3], lumCoeffs), -dot(sh[1], lumCoeffs), dot(sh[2], lumCoeffs)));

    float3 ratio = 0.0;
    ratio.r = abs(dot(float3(-sh[3].r, -sh[1].r, sh[2].r), zonalAxis));
    ratio.g = abs(dot(float3(-sh[3].g, -sh[1].g, sh[2].g), zonalAxis));
    ratio.b = abs(dot(float3(-sh[3].b, -sh[1].b, sh[2].b), zonalAxis));
    ratio /= sh[0];
    float3 zonalL2Coeff = sh[0] * (0.08f * ratio + 0.6f * ratio * ratio); // Curve-fit; Section 3.4.3

    float fZ = dot(zonalAxis, direction);
    float zhDir = sqrt(5.0f / (16.0f * PI)) * (3.0f * fZ * fZ - 1.0f);

    // Convolve sh with the normalized cosine kernel (multiply the L1 band by the zonal scale 2/3), then dot with
    // SH(direction) for linear SH (Equation 5).
    float3 result = SHLinearEvaluateIrradiance(sh, direction);

    // Add irradiance from the ZH3 term. zonalL2Coeff is the ZH3 coefficient for a radiance signal, so we need to
    // multiply by 1/4 (the L2 zonal scale for a normalized clamped cosine kernel) to evaluate irradiance.
    result += 0.25f * zonalL2Coeff * zhDir;
    return result;
}
```

STORED VS. HALLUCINATED ZH3

ACTIVISION CENTRAL TECH

- Stored has ~27% lower error.
  - Linear interpolation of non-linear quantities can have artifacts – ZH3 should shrink when blending between competing axes.
  - Therefore: should be expanded to SH3 before interpolation – more memory, more bandwidth.
- Hallucinate requires no extra data.
  - ~12 extra instructions/18 extra cycles vs. stored on RDNA (negligible).
  - Well-behaved interpolation.
- Heuristic: use hallucinate where you'd otherwise use linear SH, and consider stored ZH3 where you'd otherwise use quadratic SH.
  - SH3 is still higher quality, so consider the trade-off!

It's worth discussing when you might want to use stored ZH3 vs. hallucinated. Stored ZH3 has almost thirty percent lower error and only requires an extra three coefficients, so it's a fairly obvious choice on those grounds. However, because ZH3 is a non-linear basis when the axes differ, interpolating stored ZH3 coefficients can have artefacts; to give an example, if you have two perpendicular linear SH the ZH3 coefficient should be scaled by ¼ when those are averaged. That means that you usually want to expand ZH3 to SH3 as your in-memory representation before interpolating, which means higher memory and bandwidth costs. On the other hand, hallucinated ZH3 will naturally shrink the ZH3 coefficient when interfering linear SH are blended.

The way we think about them is that hallucinated ZH3 is a clear win over linear SH, so it should be used whenever linear SH would otherwise be used. It does add a little extra computation in computing the ZH3 axis, but the quality difference is more than worthwhile.
On the other hand, stored ZH3 is an option to replace SH3 at a lower storage footprint. You still probably want to use SH3 at runtime, so you decode ZH3 to SH3 and use that for interpolation.

RESULTS

ACTIVISION CENTRAL TECH

Reference	SH2	ZH3 (Lum. Axis)	Hallucinate (C.F.)	SH3
Ennis	RMSE = 0.556	RMSE = 0.110	RMSE = 0.159	RMSE = 0.095
Grace	RMSE = 0.0896	RMSE = 0.0140	RMSE = 0.0420	RMSE = 0.0132
Pisa	RMSE = 0.0244	RMSE = 0.0204	RMSE = 0.0214	RMSE = 0.0051
Uffizi	RMSE = 0.333	RMSE = 0.078	RMSE = 0.085	RMSE = 0.039
Wells	RMSE = 0.0552	RMSE = 0.0344	RMSE = 0.0343	RMSE = 0.0076

[Debevec01, Vogl10]

Here’s a comparison of linear SH, stored and hallucinated ZH3, and quadratic SH on a few environment maps. You can see that ZH3 is a clear improvement over linear SH in every case, and on Grace and Ennis it comes very close to matching SH3.

Note that in these results, hallucinate doesn’t use a shared axis, which is why it can have lower error on Wells than the stored coefficient solve.

BONUS: ENCODING SH

ACTIVISION CENTRAL TECH

- Lighting is strictly non-negative.
  - There are coefficient bounds on the ratio of each SH coefficient over DC for non-negative functions: $\pm\sqrt{2L+1}$ [WS22].
  - The ratio of the coefficient / DC can be stored as a quantized [-1, 1] number (or [-0.5, 1] in the case of the ZH3 basis function).
- Store DC as HDR (R11G11B10 or R9G9B9E5), every other coefficient as 8-bit (or fewer) quantized.
  - Linear SH: 13 bytes.
  - Stored ZH3: **16 bytes.**
  - Quadratic SH: 28 bytes.

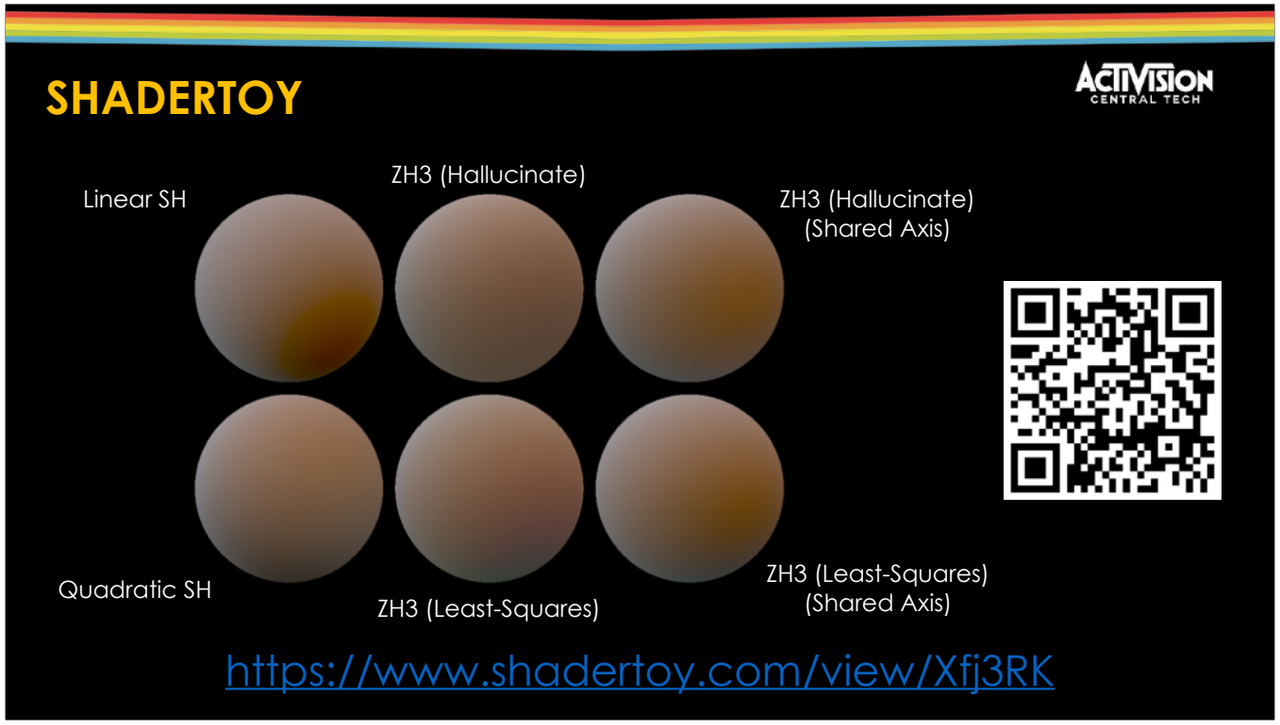
To make the storage advantage of ZH3 a bit more concrete: we use a ratio-based encoding scheme for SH where every coefficient other than DC is stored as a normalised -1 to 1 value. We can do this because there are upper bounds for SH ratios for non-negative functions, which lighting always will be. Using this scheme, linear SH fits in 13 bytes, stored ZH3 fits in 16 bytes, and quadratic SH in 28 bytes, which will usually be padded to 32 to more naturally fit cacheline sizes. That means that ZH3 takes half the storage of SH3, and comes pretty close in terms of visual quality.

HOW WE'RE USING IT

ACTIVISION CENTRAL TECH

- Hallucinated ZH3 has shipped in multiple past titles.
  - Curve fit/luminance axis is new.
- We're investigating stored ZH3 for mobile.
  - Halve our memory cost for volumetric lighting.
- We're also investigating stored ZH3 for dynamic lightsets [Sloan20].
  - Lights that can be adjusted at runtime get their own bake.
  - Potentially many dynamic lightsets per level.
  - ZH3 makes higher numbers of dynamic lights possible.

We've shipped hallucinated ZH3 in multiple games now, although we used a less accurate directional light/sphere light mix in older titles. We're looking at using ZH3 for our light grid on mobile in the future, and we're also experimenting with using ZH3 for dynamic lighting data.



We've published a ShaderToy that lets you experiment with different lighting environments and their ZH3 fits. This doesn't include the full solve, so you can get better results than the least-squares fit in practice, but it's still something interesting to play with.

Acknowledgements:

Peter-Pike Sloan

Ari Silvennoinen

Michal Iwanicki

Adrien Dubouchet

Liam Fike

Michal Drobot

Jennifer Velazquez

Michael Vance

I3D Reviewers

Yuriy O'Donnell/Probulator (https://github.com/kayru/Probulator)

In memory of Graham Madarasz

THANK YOU

https://careers.activisionblizzard.com/

REFERENCES

[Debevec01] P. Debevec, 'Light Probe Image Gallery', in Proceedings of SIGGRAPH, 2001, vol. 98. Available: https://vgl.ict.usc.edu/Data/HighResProbes/.

[GKMD06] P. Green, J. Kautz, W. Matusik, and F. Durand, 'View-Dependent Precomputed Light Transport Using Nonlinear Gaussian Function Approximations', in Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games, Redwood City, California, 2006, pp. 7–14.

[Green03] R. Green, 'Spherical Harmonic Lighting: The Gritty Details', in Game Developers Conference, 2003.

[IS17] M. Iwanicki and P.-P. Sloan, 'Ambient Dice', in 'Eurographics Symposium on Rendering - Experimental Ideas & Implementations', 2017.

[Joseph15] W. Joseph, 'Reconstructing diffuse lighting from spherical harmonic data', in CEDEC, 2015.

[McTaggart04] G. McTaggart, 'Half-Life 2 source shading', in Game Developers Conference, 2004.

[NP15] D. Neubelt and M. Pettineo, 'Advanced Lighting R&D at Ready At Dawn Studios', in SIGGRAPH 2015 Course: Physically Based Shading in Theory and Practice, 2015.

[RH01] R. Ramamoorthi and P. Hanrahan, 'An efficient representation for irradiance environment maps', in SIGGRAPH 2001 Conference Proceedings, August 12--17, 2001, Los Angeles, CA, 2001, pp. 497–500.

[SLS05] P.-P. Sloan, B. Luna, and J. Snyder, 'Local, Deformable Precomputed Radiance Transfer', ACM Trans. Graph., vol. 24, no. 3, pp. 1216–1224, Jul. 2005.

[SS20] P.-P. Sloan and A. Silvennoinen, 'Precomputed Lighting Advances in Call of Duty: Modern Warfare', in SIGGRAPH Course: Advances in Real-Time Rendering in Games, 2020.

[Vogl10] B. Vogl, Light probes. Sep-2010. Available: https://dativ.at/lightprobes/.

[WS22] T. Wiederien and P.-P. Sloan, 'Tighter Spherical Harmonic Quantization Bound', in ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - Posters, 2022.