Wavelet Representation and Sampling of Complex Luminaires — Supplementary Material

A. Atanasov and V. Koylazov

Chaos Software, Bulgaria

Our wavelet-based approach for rendering complex luminaires [AK25] is self-contained and offers our best set of techniques that work well together in practice. This document offers additional details. Section 1 provides a brief introduction to wavelet computations and Section 2 explains how we store the sparse wavelet coefficients, both with a focus on fast per-sample reconstructions. In Section 3, we show how our wavelet representation naturally admits levels of detail on rectangular domains, e.g., the six box sides of a field represented on the bounding box of a luminaire. This section is omitted from the paper because it does not work with our convex hull fields. Nonetheless, it is interesting and might prove useful in other situations, e.g., representing the exitant 4D radiance of rectangular light sources.

1. Wavelets introduction

Wavelets are a common and well-understood tool for image compression. Due to the development of orthogonal wavelets with compact support known as Daubechies wavelets [Dau92], wavelet compression is also highly efficient. Since we perform per-sample decompression at run-time, we pick the simplest and most computationally efficient wavelet from the Daubechies wavelet family — the Haar wavelet. This also happens to be the first described wavelet, in the early 20th century, long before the development of the wavelet theory. Here we briefly introduce the Haar wavelet transform from a computational perspective.

1.1. Haar transform in 1D

The Haar transform is an orthogonal linear transformation that acts on a signal of a power-of-two length.

Its main building block is the Haar transform of size two:

$$H_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{1}$$

which is not only orthogonal — it is its own inverse ($H_2^2 = I$). It acts on a signal of size two as a combination of a low-pass filter $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ and a high-pass filter $\left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$:

$$H_2 \begin{bmatrix} s_0 \\ s_1 \end{bmatrix} = \begin{bmatrix} l \\ h \end{bmatrix} \tag{2}$$

that separates the signal into a low-frequency component l (a smooth representation of the signal), and a high-frequency component h ($|h|$ is high if there is a jump in the signal).

The general Haar transform is a discrete wavelet transform (DWT) that applies H_2 hierarchically:

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} \xrightarrow{H_2} \begin{bmatrix} l_0 \\ h_0 \\ l_1 \\ h_1 \\ l_2 \\ h_2 \\ l_3 \\ h_3 \end{bmatrix} \xrightarrow{\text{permute}} \left[\begin{array}{c} l_0 \\ l_1 \\ l_2 \\ l_3 \\ \hline h_0 \\ h_1 \\ h_2 \\ h_3 \end{array}\right] \xrightarrow{H_2} \left[\begin{array}{c} L_0 \\ H_0 \\ L_1 \\ H_1 \\ \hline h_0 \\ h_1 \\ h_2 \\ h_3 \end{array}\right] \xrightarrow{\text{permute}} \left[\begin{array}{c} L_0 \\ L_1 \\ \hline H_0 \\ H_1 \\ \hline h_0 \\ h_1 \\ h_2 \\ h_3 \end{array}\right] \xrightarrow{H_2} \left[\begin{array}{c} \mathcal{L} \\ \hline \mathcal{H} \\ \hline H_0 \\ H_1 \\ \hline h_0 \\ h_1 \\ h_2 \\ h_3 \end{array}\right], \tag{3}$$

it applies H_2 to all subsequent pairs of elements of the signal, then pulls all low-frequency components at the beginning of the signal and leaves all high frequencies in the end, and then repeats this process by filtering the low frequencies in the first half of the signal. This process continues until it distills all frequency bands $\mathcal{B}$ which are contained in the signal: the lowest frequency $\mathcal{B}_0 = \{\mathcal{L}\}$, followed by a higher frequency $\mathcal{B}_1 = \{\mathcal{H}\}$, then higher frequencies $\mathcal{B}_2 = \{H_0, H_1\}$, and finally, in the second half of the array are the highest frequencies $\mathcal{B}_3 = \{h_0, h_1, h_2, h_3\}$. A signal of size 2^n is decomposed into $n+1$ bands — the lowest frequency plus n bands of higher frequencies of sizes $|\mathcal{B}_k| = 2^{k-1}$.

The process described in Equation (3) is a direct instruction on how to efficiently compute the Haar transform. It was inspired by a similar illustration from the book Numerical Recipes [PTVF07].

Note that the Haar transform can be represented by a single orthogonal matrix H ($H^{-1} = H^T$) since all operations between the column vectors in Equation (3) can be represented by orthogonal matrices: successive H_2 filters can be represented by a band diagonal matrices with H_2's or ones in the appropriate places on the diagonal and permutation matrices. Hence, the inverse Haar transform is its transpose H^T.

For our approach, we are interested in computing a single sample from the original signal from its Haar transform and not computing the full inverse transform. To reconstruct a single sample, only one frequency from each band is needed. For example, in Equation (3)

© 2025 The Author(s).

the sample s_5 depends only on l_2 and h_2, l_2 depends on L_1 and H_1, and L_1 depends on $\mathcal{L}$ and $\mathcal{H}$. Therefore, s_5 can be reconstructed starting from the lowest frequency $\mathcal{L}$ and introducing higher frequencies $\mathcal{H}$, H_1 and h_2. For a signal of size 2^n we can reconstruct each sample using the formula

$$s_i = \left(\frac{1}{\sqrt{2}}\right)^n \mathcal{L} + \sum_{k=1}^{n} \left(\frac{1}{\sqrt{2}}\right)^{n-k+1} (-1)^{i_k^n} \mathcal{B}_k\left(i_{k-1}^n\right), \quad (4)$$

where the signs and indices of the band frequencies are computed using

$$i_k^n = \left\lfloor \frac{i}{2^{n-k}} \right\rfloor. \quad (5)$$

1.2. Haar transform in 2D

The two most common ways to generalize the 1D wavelet decomposition to higher dimensions are the *standard* and the *non-standard* wavelet decompositions [Dau92]. The standard decomposition performs full 1D decompositions on the rows of the signal followed by full 1D decompositions on the columns, similarly to the 2D Fourier transform [PTVF07]. On the other hand, the non-standard decomposition performs one level of the hierarchy (one pairwise application of the H_2 transforms followed by one permutation) first on the rows and then on the columns, and then continues to decompose the low frequencies in the same manner until it reaches the lowest frequency. The two decompositions have different band structures. For a 2D signal of size $2^n \times 2^n$, the standard decomposition has a total of $(n+1)^2$ partitions, while the non-standard has $3n+1$ partitions, see Figure 1.

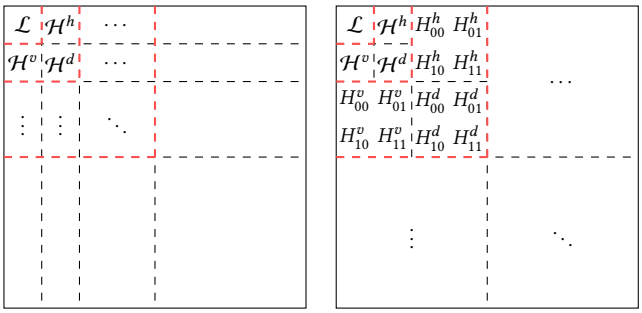


Figure 1: *Band structures of the standard (left) and the non-standard (right) 2D wavelet decomposition of a signal of size* 8×8. *These decompositions result in different partitions of the wavelet coefficients (drawn with dashed lines): the standard decomposition has 16 partitions, while the non-standard has 10. Both have 4 frequency bands (indicated in red).*

To reconstruct a single sample from a 2D decomposition one needs to account for one wavelet coefficient from each band partition. The standard decomposition yields more partitions which increase with each subsequent band, thus potentially achieving higher sparsity for anisotropic data. The non-standard decomposition has a constant of three partitions per band, thus admitting faster and simpler run-time reconstructions. Therefore, we opt for the simpler non-standard decomposition, given that we achieve good compression rates without illumination artifacts.

In Figure 1, we see the band structure of the non-standard decomposition. After the lowest frequency $\mathcal{L}$, all bands are partitioned into horizontal, vertical and diagonal higher frequencies:

$$\begin{aligned} \mathcal{B}_1 &= \{\mathcal{H}^h, \mathcal{H}^v, \mathcal{H}^d\}, \\ \mathcal{B}_2 &= \{H_{00}^h, \ldots, H_{11}^h, H_{00}^v, \ldots, H_{11}^v, H_{00}^d, \ldots, H_{11}^d\}, \\ &\cdots \\ \mathcal{B}_n &= \{\ldots\}. \end{aligned} \quad (6)$$

An analogous formula to Equation (4) for the decompression of a single sample can be derived for the non-standard 2D decomposition:

$$\begin{aligned} s_{ij} = \left(\frac{1}{2}\right)^n \mathcal{L} + \sum_{k=1}^{n} \left(\frac{1}{2}\right)^{n-k+1} \Big[\\ (-1)^{i_k^n} \mathcal{B}_k^h\left(i_{k-1}^n, j_{k-1}^n\right) + (-1)^{j_k^n} \mathcal{B}_k^v\left(i_{k-1}^n, j_{k-1}^n\right) \\ + (-1)^{i_k^n} (-1)^{j_k^n} \mathcal{B}_k^d\left(i_{k-1}^n, j_{k-1}^n\right) \Big]. \end{aligned} \quad (7)$$

2. Wavelet coefficients preparation

After the compression, only a small fraction of the wavelet coefficients is kept. In this section, we explain how to organize the sparse wavelet coefficients to facilitate fast run-time reconstructions.

2.1. Change of wavelet basis

Compression based on the Haar transform preserves the overall structure of the signal by boosting the lower frequencies to reduce the chance of truncation, which is exactly what we want. However, after the coefficients are truncated, we can fold all constants from Equations 4 and 7 into the wavelet coefficients.

A wavelet transform/basis that is not orthogonal, but is still invertible is referred to as a biorthogonal transform/basis. In particular, we propose the arithmetic mean wavelet transform, which is similar to the Haar transform but is not orthogonal. The 2×2 arithmetic mean transform and its inverse are

$$M_2 = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad M_2^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (8)$$

A 2-component vector transformed by M_2 has its arithmetic mean in the first component and its mean difference in the second component. Note also that the inverse transform does not have scaling factors — it is just one addition and one subtraction. Therefore, the hierarchical wavelet decomposition corresponding to M_2 (either 1D or 2D) has two advantages:

- *Forward decomposition.* The lowest frequency $\mathcal{L}$ is the arithmetic mean of the signal and all intermediate low frequencies L, l and so on are arithmetic means of the signal's sub-quadrants. This property is useful for implementing levels of detail of the signal.
- *Inverse decomposition.* The sample reconstruction for the arithmetic mean transform corresponding to Equations (4) and (7)

© 2025 The Author(s).

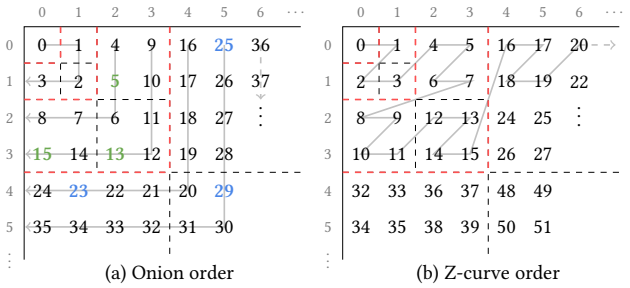
will have only additions and subtractions resulting in simpler and more efficient reconstructions.

For this reason, we change the basis of the 2D sparse wavelet coefficients that are left after the truncation from the Haar basis to the arithmetic mean basis. This is done by multiplying $\mathcal{L}$ by 2^{-n} and all subsequent bands $\mathcal{B}_k$ by 2^{k-n-1}, see Equation (7).

2.2. Sorting the wavelet coefficients

An important question is how to organize and access the sparse wavelet coefficients. There are different options, e.g., building a hierarchical data structure and run queries or sorting the coefficients by index and run binary searches. Given that our transforms are relatively small (usually 32×32 or 64×64) and after compression only a small fraction of all coefficients is kept, we decided to order the wavelet coefficients in ascending frequency band order. For a run-time signal reconstruction, we compute Equation 7 term by term using a single and cache-friendly pass through the coefficients.

To this end, we experimented with two ordering schemes that are illustrated in Figure 2: *Onion order* and *Z-curve order*. Both are implemented in our paper source code and according to our experiments they perform identically.



(a) Onion order (b) Z-curve order

Figure 2: *Two ordering schemes (gray lines) for the non-standard 2D wavelet transform guaranteeing that all coefficients come in ascending frequency bands (red dashed lines).* ***Left:*** *Onion order — all coefficients come in consecutive layers. When $i \leq j$ the frequencies have onion indices in the order horizontal-diagonal-vertical (e.g., green indices 5-13-15), while for $i > j$ the order is vertical-horizontal-diagonal (e.g., blue indices 23-25-29).* ***Right:*** *Z-curve order that always traverse all coefficients in horizontal-vertical-diagonal order.*

Currently, we use the *Onion order* in our V-Ray implementation and also for all Chaos Cosmos library assets [Cha25] that support the technique because this was our initial implementation. The only reason for choosing to publish the paper using the *Z-curve order* was to reduce the paper size — this order is widely known and used throughout computer graphics [PJH23]. The *Z-curve order* is also slightly more elegant to implement because the high frequencies in each band always follow horizontal-vertical-diagonal order.

The idea behind the *Onion order* is to index a square array starting from the top left corner and adding layers of subsequent indices, hence the name. The onion index corresponding to the Cartesian coordinates (i, j) of a coefficient is computed as:

$$o(i,j) = \begin{cases} i^2 + j, & i \geq j \\ (j+1)^2 - i - 1, & \text{otherwise.} \end{cases} \tag{9}$$

Note also that the high-frequency coefficients in each band are ordered differently for a requested sample below and above the main diagonal, see Figure 2.

3. Level of detail

For luminaire fields represented on a box, LoD can be implemented efficiently. The arithmetic mean transform provides a natural way to reconstruct the average intensities of the signal. Starting from the quad corresponding to the entire box side, each quad is subdivided into four equal quadrants until the resolution of the signal is reached.

In Equation (7), the first term is the average intensity of the compressed signal. The sum of the first two terms is the average intensity of the quadrant containing the reconstructed point s_{ij} and so on. Essentially, adding higher frequencies is upscaling the signal.

We devise an evaluation strategy for the field that utilizes this property of the transform. For shading points that are close to the luminaire we reconstruct the full signal while for distant shading points, we compute partial sums of Equation (7) that reconstruct the signal's quadrant averages. This strategy has two advantages:

- The field's tiles are filtered identically to MIP maps.
- The field evaluation becomes more efficient with distance.

3.1. Choosing LoD

We select an appropriate level of detail based on the distance from the shading point to the center of the side of the field's box. For each axis $v \in \{x, y, z\}$, we compute a maximum LoD distance d_v, beyond which we only return the average tile intensity (the first coefficient in the mean average wavelet transform). This maximum LoD distance is chosen so that the solid angle of the directional bins σ is larger than the solid angle subtended at the shading point by the side of the field's box. When this condition is fulfilled, the whole tile is seen roughly from a single bin, and we can return the tile's average intensity without further reconstruction.

We use a conservative estimate of d_v which ensures the accurate filtering of the field

$$d_v = \frac{\max(w_v, h_v)}{\sqrt{\sigma}} \geq \sqrt{\frac{w_v h_v}{\sigma}}, \tag{10}$$

where w_v and h_v are the width and the height of the box sides along axis v, and we have approximated the solid angle of the box side at a distance d_v with $w_v h_v / d_v^2$. This formula is illustrated in Figure 3. We have tried tighter estimates of d_v which bring very slight performance gains, but they result in over-filtering manifested as aliasing.

For a tile of size $2^n \times 2^n$ and a shading point at distance d, we reconstruct the full signal when $d \leq d_v 2^{-n}$ and we linearly interpolate between bands $k-1$ and k when

$$d_v 2^{-k} < d \leq d_v 2^{1-k}, \; k = 1 \ldots n. \tag{11}$$

© 2025 The Author(s).

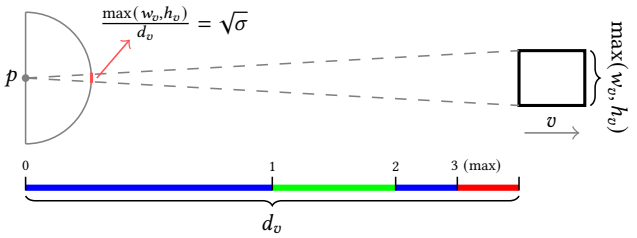


Figure 3: *We choose the maximum LoD distance d_v for each axis v so that the longer edge of the two sides along v subtends a planar angle of size $\sqrt{\sigma}$ at a point p centered above the side at a distance d_v. Then we subdivide the distance d_v for all bands as follows: in the first half of the distance (blue) we interpolate between bands 0 and 1, then in the first half of the remaining part (green) we interpolate between bands 1 and 2, and so on. Once the maximum band is reached, we reconstruct the full field (red).*

These interpolation intervals are illustrated in Figure 3. Finally, for $d > d_v$, we return $2^{-n}\mathcal{L}$, the average tile intensity, see Equation (7).

3.2. Discussion

We observed that usually there is no visible difference when rendering with and without LoD. In Figure 4, a small difference can be noticed due to the strong directional illumination of the Mustang headlights. The speed-up for this scene due to LoD is around 15% while for common light fixtures with weaker illumination it is less than that. In this figure, we also include a comparison with V-Ray's progressive caustics algorithm which is based on the technique by Šik and Křivánek [ŠK19]. The LoD is implemented in the paper source code and works correctly only when the field is represented on a box. Such fields however, could lead to box artifacts close to the box, see Figure 4. Therefore, in the paper we only showcase our convex hull fields without LoD.

Acknowledgements

We thank Dian Nikolov for suggesting we try the Z-curve order and Bozhidar Stoyanov for setting up the Mustang scene.

References

[AK25] ATANASOV, ASEN and KOYLAZOV, VLADIMIR. "Wavelet Representation and Sampling of Complex Luminaires". *Computer Graphics Forum* (2025). ISSN: 1467-8659. DOI: 10.1111/cgf.70163 1.

[Cha25] CHAOS. *Chaos Cosmos Asset Library*. [Online; accessed 8-June-2025]. 2025. URL: https://cosmos.chaos.com/ 3.

[Dau92] DAUBECHIES, INGRID. *Ten Lectures on Wavelets*. USA: Society for Industrial and Applied Mathematics, 1992. ISBN: 0898712742 1, 2.

[PJH23] PHARR, MATT, JAKOB, WENZEL, and HUMPHREYS, GREG. *Physically Based Rendering: From Theory to Implementation*. 4th. Cambridge, MA, USA: The MIT Press, Mar. 2023, 1266. ISBN: 978-0262048026 3.

[PTVF07] PRESS, WILLIAM H., TEUKOLSKY, SAUL A., VETTERLING, WILLIAM T., and FLANNERY, BRIAN P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd ed. USA: Cambridge University Press, 2007. ISBN: 0521880688 1, 2.

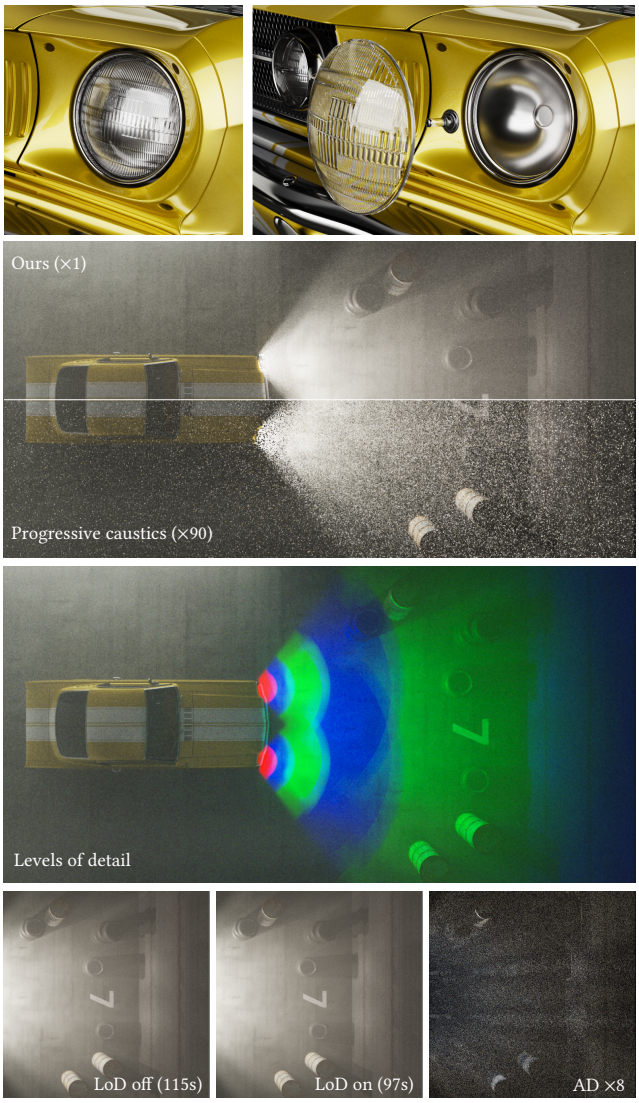


Figure 4: *Mustang headlights shinning in fog.* ***First row:*** *A close-up of a headlight which consists of a reflector, a light bulb and a glass cover. A spherical emitter is placed inside the glass tube of the light bulb.* ***Second row:*** *Equal time comparison for 1 min between our method (top) and the progressive caustics technique by Šik and Křivánek [ŠK19].* ***Third row:*** *Illustration of the levels of detail: full signal reconstruction (red) and the subsequent decreasing odd (blue) and even (green) bands.* ***Fourth row:*** *LoD disabled (left), LoD enabled (middle) and their absolute difference, magnified eight times (right).*

[ŠK19] ŠIK, MARTIN and KŘIVÁNEK, JAROSLAV. "Implementing One-Click Caustics in Corona Renderer". *Eurographics Symposium on Rendering - DL-only and Industry Track*. Ed. by BOUBEKEUR, TAMY and SEN, PRADEEP. The Eurographics Association, 2019, 61–67. ISBN: 978-3-03868-095-6. DOI: 10.2312/sr.20191221 4.

© 2025 The Author(s).