# A Generalized 2D and 3D Hilbert Curve

Jakub Červený and Zzyv Zzyzek

## Abstract

Hilbert curves are classic space-filling curves with strong locality properties, widely used for linearizing 2D and 3D data. In their discrete form, however, the standard Hilbert construction applies most naturally to square or cubic grids with side lengths that are exact powers of two. We present the *Gilbert curve*, a simple recursive, Hilbert-like construction that produces paths on 2D rectangles and 3D rectangular cuboids. Our construction explicitly manages endpoint parity constraints, which allows us to extend naturally from 2D to 3D.

## Introduction

Space-filling curves map a one-dimensional ordering to a multi-dimensional grid while preserving locality. Discrete Hilbert curves, in particular, can be used to linearize images or volumes for cache-coherent traversal, construct spatial indexes, visualize data layout, and more [1, 2, 3, 5, 6]. In practical settings, however, underlying grids might not be square or cubic with power-of-two side lengths. Images and volumes can come in arbitrary dimensions. Tiling, padding, or resampling to fit a power-of-two Hilbert curve introduces overhead and can harm locality near boundaries.

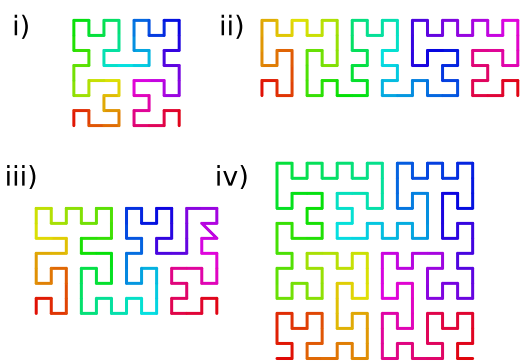


**Figure 1:** *2D Gilbert curves for i)* $8 \times 8$, *ii)* $18 \times 6$, *iii)* $13 \times 8$ *(with diag. move), iv)* $14 \times 14$.

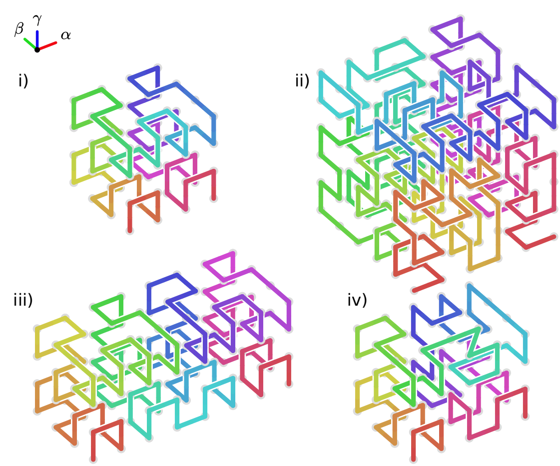


**Figure 2:** *3D Gilbert curves for i)* $4 \times 4 \times 4$, *ii)* $6 \times 6 \times 6$, *iii)* $8 \times 4 \times 4$, *iv)* $5 \times 4 \times 4$ *(with diag. move).*

This paper presents the *Gilbert curve*, a recursive construction that produces paths for 2D rectangles and 3D cuboids. Our goal is a path construction that is easy to implement, has a uniform recursive structure, and reduces to a Hilbert curve in the power-of-two square/cube cases. In this paper, we:

- We describe a simple recursive construction of Hilbert-like paths on arbitrary 2D rectangular grids
- We formalize a parity condition on corner endpoints and use it to guide valid recursive subdivisions
- We extend the same construction principles naturally to arbitrary 3D cuboid grids

In certain conditions, combinations of side length sizes and the parity of path endpoints preclude a strict non self-intersecting path that visits every cell exactly once (a *Hamiltonian path*) without any diagonal moves. We will briefly address the conditions under which the Gilbert curve algorithm introduces diagonal moves throughout this paper.

## Related Work

Hilbert defined a continuous, space filling curve that maps the unit interval to the unit square, which as been since named the *Hilbet curve* [4]. One discretization variant applies to square (2D) and cubic (3D) grids with equal sides that are powers of two.

In a website application and scanned note, Tautenhahn [7] presented a 2D construction for arbitrary sizes based on combining $3 \times 3$ Peano and $2 \times 2$ Hilbert block types. While Tautenhahn's approach is 2D only and has non-uniform subdivision schemes dependent on side length parity, it motivates an important endpoint-parity constraint that we make explicit.

Zhang, Kamata, and Ueshige [8] propose a pseudo-Hilbert scan for arbitrary rectangles. Compared to our construction, their method follows a more intricate case structure and does not suggest a direct extension to 3D.

## Parity Constraints on Corner Endpoints

We use a parity argument to determine whether a Hamiltonian path is possible between two grid corner endpoints. If we consider the standard checkerboard coloring of an $m \times n$ grid graph (or an $m \times n \times p$ grid graph in 3D), then adjacent vertices always have opposite color. Any path alternates colors at every step, so the parity of the endpoints is constrained by the total number of visited vertices.

**Lemma 1** (Color compatibility)**.** *Let $G$ be a rectangular 2D grid graph with $(m{\times}n)$ vertices, or a rectangular 3D grid graph with $(m{\times}n{\times}p)$ vertices. Fix a start corner, $s$, and an end corner, $t$. If there exists a Hamiltonian path from $s$ to $t$, then $s$ and $t$ have the* same *color when the number of vertices is odd, and* opposite *colors when the number of vertices is even.*

*Proof.* Along any path, vertex colors alternate. A Hamiltonian path visits all $N$ vertices exactly once and therefore has $N - 1$ steps. If $N$ is odd, then $N - 1$ is even and the endpoints must have the same color. If $N$ is even, then $N - 1$ is odd and the endpoints must have opposite colors. □

We call a pair of endpoints *color compatible* if they satisfy Lemma 1. Color compatibility is necessary, in general, for Hamiltonicity. For the specific family of corner-to-corner subproblems produced by our construction, it is also sufficient, and we implicitly use it to choose endpoints and subregions for the recursion.

## Algorithms

Algorithms 1 and 2 give pseudocode for the 2D and 3D Gilbert curve constructions. We represent an oriented grid-aligned rectangle or cuboid by an origin $p \in \mathbb{Z}^d$ and axis vectors $\alpha, \beta, \gamma \in \mathbb{Z}^d$, $d \in \{2, 3\}$, whose nonzero components encode the side lengths and directions. The function $\delta(\cdot)$ returns the unit axis direction of its argument, and $\mathrm{div}(\cdot,\cdot)$ denotes integer division (rounding toward zero). To make things more concise, a convenience function is used, $(\Delta(\rho_2, \rho) \stackrel{\text{def}}{=} (\ \textit{if}\ (|\rho_2| \equiv 1 \bmod 2 \text{ and } |\rho| > 2) \rightarrow \delta(\rho),\ \textit{else}\ \rightarrow 0))$, to help coerce values to be even in the the non-degenerate case.

The next sections go into detail about the 2D construction and its extension to 3D.

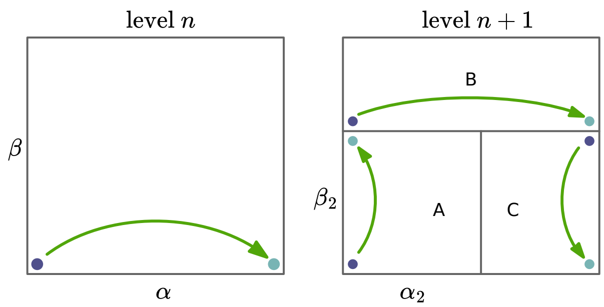


**Figure 3:** *Subdivision template for the 2D Gilbert curve.*

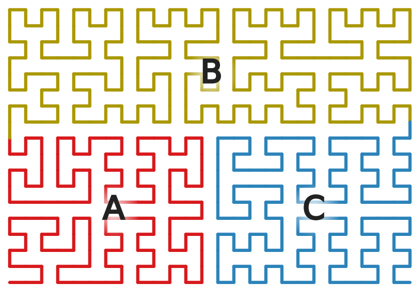


**Figure 4:** *Example curve with subdivision regions highlighted.*

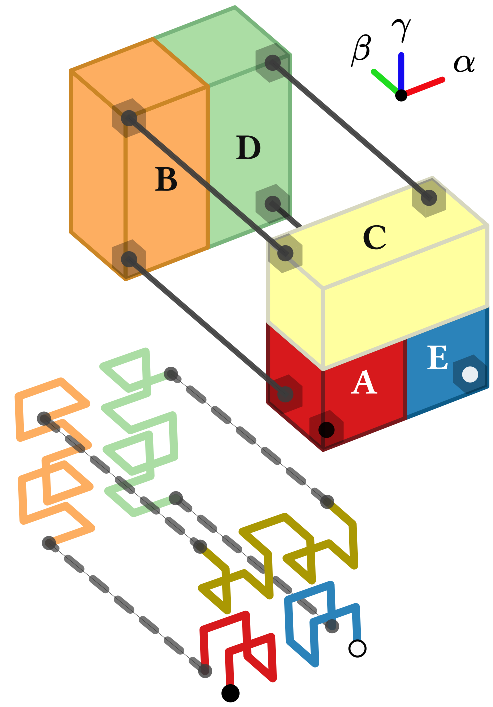


**Figure 5:** *The main subdivision template for the 3D Gilbert curve.*

## The 2D Gilbert Curve

The 2D Gilbert curve construction recursively partitions an $m \times n$ rectangle into a small number of axis-aligned sub-rectangles and concatenates the subpaths into a single path. Figure 3 shows the main 2D subdivision template and Figure 4 illustrates one recursion with subregions highlighted.

The vectors $\alpha, \beta \in \mathbb{Z}^2$ provide generalized notions of width ($\alpha$) and height ($\beta$). Each of $\alpha, \beta$ will only have one non-zero component and will be orthogonal to each other, representing a snapshot of the current frame of reference.

When dividing into subregions, we choose integral side lengths and prefer an even local width-like length for the first region (e.g. region $A$ in Figures 3, 4) to satisfy the parity constraints of Lemma 1. Each subregion is solved recursively and then stitched to its neighbors. Figure 8 shows a color coded subdivision template and Figure 9 shows a color coded example with stitched endpoints highlighted.

As a heuristic to keep subregions square like, the parent region is bisected if the aspect ratio of width to height exceeds a $(3/2)$ threshold. After bisection, the algorithm recursively proceeds on each half as normal. See Figure 6 for the subdivision template and Figure 7 for an example of the bisection subdivision.

If the width like side length is odd with the height even, a diagonal move is forced as the endpoints won't be color compatible anymore. In such a case, there will be a single diagonal move in the resulting path and it will be guided to appear in the upper right hand corner, as the $B$ region (Figure 3) will retain the odd width and even height.

In all other cases (even width, odd width and height), a Hamiltonian path is possible without diagonal moves. When the side lengths are exact powers of two, the resulting curve is identical to the 2D Hilbert curve. Algorithm 1 gives pseudo code for the 2D Gilbert curve.

**Algorithm 1** Gilbert 2D

# $p, \alpha, \beta \in \mathbb{Z}^2$
**function** GILBERT2D($p$, $\alpha$, $\beta$)
  $\alpha_2, \beta_2 = \mathrm{div}(\alpha, 2), \mathrm{div}(\beta, 2)$
  **if** $(|\beta| \equiv 1)$ **then**
    **yield** $p + i \cdot \delta(\alpha)$ **forall** $i \in |\alpha|$
  **else if** $(|\alpha| \equiv 1)$ **then**
    **yield** $p + i \cdot \delta(\beta)$ **forall** $i \in |\beta|$
  **else if** $(2|\alpha| > 3|\beta|)$ **then**
    **if** $(|\alpha_2| > 2)$ and
      $(|\alpha_2| \bmod 2 \equiv 1)$ **then**
      $\alpha_2 \leftarrow \alpha_2 + \delta(\alpha)$
    **end if**
    **yield** Gilbert2D($p$, $\alpha_2$, $\beta$)
    **yield** Gilbert2D($p + \alpha_2$, $\alpha - \alpha_2$, $\beta$)
  **else**
    **if** $(|\beta_2| > 2)$ and
      $(|\beta_2| \bmod 2 \equiv 1)$ **then**
      $\beta_2 \leftarrow \beta_2 + \delta(\beta)$
    **end if**
    **yield** Gilbert2D($p$,
        $\beta_2, \alpha_2$)
    **yield** Gilbert2D($p + \beta_2$,
        $\alpha, (\beta - \beta_2)$)
    **yield** Gilbert2D($p + (\alpha - \delta(\alpha)) +$
        $(\beta_2 - \delta(\beta))$,
        $-\beta_2, -(\alpha - \alpha_2)$)
  **end if**
**end function**

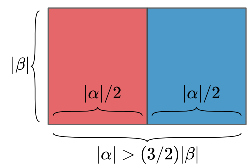


**Figure 6**

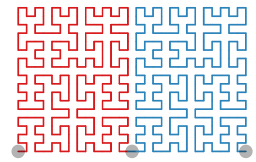

**Figure 7**

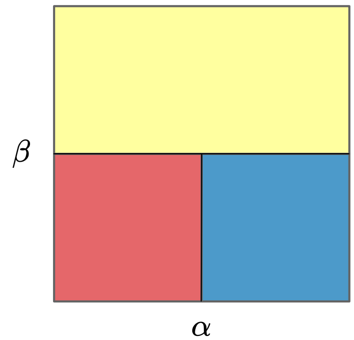


**Figure 8**

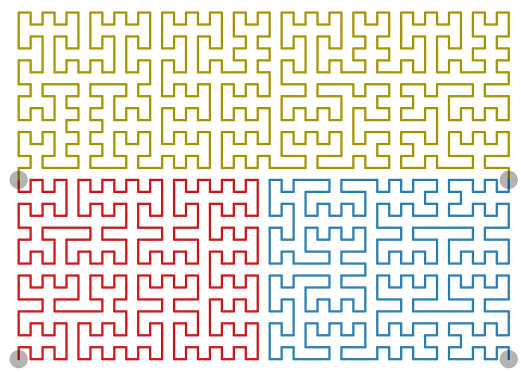

**Figure 9**

## Extension to 3D

In the basic case, a subdivision scheme is used to split the cuboid into five regions. Two cube like regions are partitioned where the path starts and stops, and three oblong cuboid regions are partitioned for the middle portion of the path. Figure 5 provides an exploded view of the main subdivision and where endpoints connect. See also Figure 16 for the template and Figure 17 for a $10 \times 10 \times 10$ example, with each region color coded.

During the course of subdivision, if the cuboid has an aspect ratio of width to either height or depth that exceeds $(3/2)$, a horizontal bisection is done. Two other anisotropic cases are tested for when the aspect ratio of height to depth or depth to height exceeds a $(4/3)$ threshold. The reader is referred to Algorithm 2 for further details.

Endpoints within a sub-divided region are kept on the exterior of the parent cuboid and joined after the resulting recursion has completed. In the case all side lengths are even, the subdivision will always choose an even length until the base case is encountered, ensuring a Hamiltonian path and a realization free from diagonal moves. As with the 2D Gilbert curve, when side dimensions are equal and exact powers of two, the resulting curve is identical to one variant of the 3D Hilbert curve.

**Algorithm 2** Gilbert 3D

```
# p, α, β, γ ∈ ℤ^3
function GILBERT3D(p, α, β, γ)
    if (|β| ≡ 1) and (|γ| ≡ 1) then
        yield p + i · δ(α) forall i ∈ |α|
    else if (|α| ≡ 1) and (|γ| ≡ 1) then
        yield p + i · δ(β) forall i ∈ |β|
    else if (|α| ≡ 1) and (|β| ≡ 1) then
        yield p + i · δ(γ) forall i ∈ |γ|
    end if
    α_2 ← div(α, 2) + Δ(α_2, α)
    β_2 ← div(β, 2) + Δ(β_2, β)
    γ_2 ← div(γ, 2) + Δ(γ_2, γ)
    if (2|α| > 3|β|) and (2|α| > 3|γ|))
      yield Gilbert3D(p,α_2,β,γ)
      yield Gilbert3D(p + α_2,α − α_2,β,γ)
    else if (3|β| > 4|γ|)
      yield Gilbert3D(p,β_2,γ,α_2)
      yield Gilbert3D(p + β_2,α,β − β_2,γ)
      yield Gilbert3D(p+
                          (α − δ(α))+
                          (β_2 − δ(β)),
                          −β_2,γ,−(α − α_2))
    else if (3|γ| > 4|β|)
      yield Gilbert3D(p,γ_2,α_2,β)
      yield Gilbert3D(p + γ_2,α, β, γ − γ_2)
      yield Gilbert3D(p+
                          (α − δ(α))
                          (γ2 − δ(γ)),
                           −γ_2,−(α − α_2), β)

    else
      yield Gilbert3D(p,β_2,γ_2,α_2)
      yield Gilbert3D(p + β_2,γ,α_2,(β − β_2))
      yield Gilbert3D(p+
                          (β_2 − δ(β))+
                          (γ − δ(γ)),
                           α,−β_2,−(γ − γ_2))
      yield Gilbert3D(p+
                          (α_2 − δ(α))+
                          β_2 + (γ − δ(γ)),
                           −γ,−(α − α_2),(β − β_2))
      yield Gilbert3D(p+
                          (α − δ(α))+
                          (β_2 − δ(β)),
                           −β_2,γ_2,−(α − α_2))
end function
```

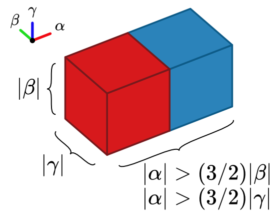


Figure 10

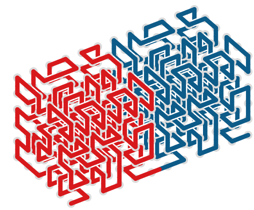

Figure 11

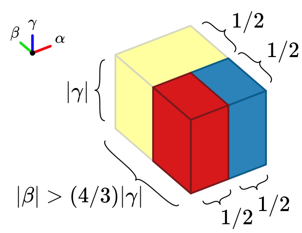


Figure 12

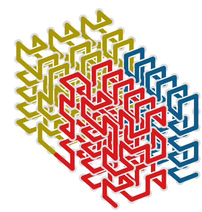

Figure 13

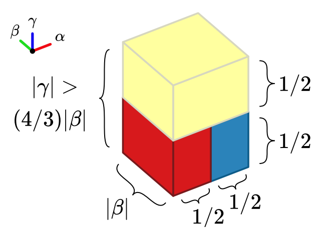


Figure 14

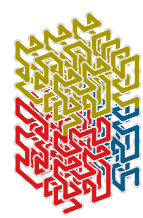

Figure 15

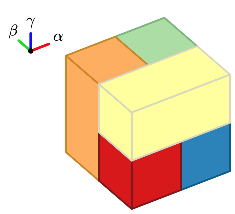


Figure 16

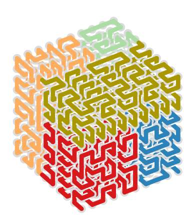

Figure 17

If any of the subdivided regions has endpoints that lay in a direction with odd side length, a diagonal move will appear. As such, with the subdivision scheme presented, the number of diagonal moves in the resulting curve isn't limited to one.

## Summary and Conclusions

The Gilbert curve provides a practical alternative to discrete Hilbert curves when the target grid has arbitrary rectangular dimensions. This avoids padding or resampling that is otherwise required to fit power-of-two constructions, while maintaining a recursive, locality-preserving traversal. Such layouts are useful for cache-coherent image/volume traversal, spatial indexing, and visualization of 1D data on 2D/3D grids [1, 2, 3, 5].

While our anisotropic-split thresholds are simple and work well in practice, future work could define more explicit optimization criteria for when and how to switch subdivision templates. In 3D, relaxing subproblem endpoint constraints may allow constructions that further reduce the occurrence of forced diagonal steps when one or more side lengths are odd.

A libre/free/open implementation for the Gilbert curve in 2D and 3D has been developed and can be downloaded from its repository [1].

## Acknowledgments

AI tools were used in assisting the language and readability of this paper.

## References

[1] S. Anders. "Visualization of genomic data with the Hilbert curve." *Bioinformatics*, vol. 25, no. 10, pp. 1231-1235, 2009. https://doi.org/10.1093/bioinformatics/btp152

[2] C. Böhm, M. Perdacher, and C. Plant "A Novel Hilbert Curve for Cache-Locality Preserving Loops." *IEEE Transactions on Big Data*, vol. 7, 2021, pp. 241-254

[3] A. Cortesi. *Visualizing binaries with space-filling curves*. 2011. https://corte.si/posts/visualisation/binvis/.

[4] D. Hilbert. "Ueber die stetige Abbildung einer Linie auf ein Flächenstück.", *Mathematische Annalen*, vol. 38, 1891, pp. 459-460

[5] B. Moon, H. Jagadish, C. Faloutsos, and J. Saltz. "Analysis of the clustering properties of the Hilbert space-filling curve." *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 1, 2001, pp. 124-141.

[6] R. Munroe. *Map of the Internet*. xkcd. 2006. https://xkcd.com/195/

[7] L. Tautenhahn. *Draw a space-filling curve of arbitrary size*, 2003. https://lutanho.net/pic2html/draw_sfc.html.

[8] J. Zhang, S. Kamata, and Y. Ueshige. "A Pseudo-Hilbert Scan Algorithm for Arbitrarily-Sized Rectangle Region." *Advances in Machine Vision, Image Processing, and Pattern Analysis*, Berlin, Heidelberg, 2006, pp. 290-299.

[1] https://github.com/jakubcerveny/gilbert