

A Generalized 2D and 3D Hilbert Curve

Jakub Červený and Zzyv Zzyzek

Abstract

The two and three dimensional Hilbert curves are fractal space filling curves that map the unit interval to the unit square or unit cube while preserving a notion of closeness, or locality, after the map. We present the Gilbert curve, a conceptually straight forward generalization of the Hilbert curve, that works on arbitrary rectangular regions, overcoming the limitation of the Hilbert curve that requires side lengths be exact powers of two. Our construction of the Gilbert curve provides a notion of *harmony* that provide alternate subdivision schemes when a side length is much larger than the rest, creating more visual pleasing curves as a result.

Generalized Hilbert Curves

In a website application and scanned note [1], Tautenhahn provided the basis for a generalized 2D space filling curve. Tautenhahn also included policies for when to subdivide regions preferentially in only one dimension that help to create more *harmonious* curve realizations. Tautenhahn’s exploration details the parity arguments necessary for when a subdivision scheme can be employed without creating diagonal moves, which we call *notches* here.

In this paper, we extend Tautenhahn’s ideas to create a 2D and 3D generalized Hilbert curve. We further extend Tautenhahn’s core ideas on when to use alternate subdivision schemes when one length is much larger than the rest, what we call *eccentric cases*, and apply them to a 3D generalized Hilbert curve. Tautenhahn’s unpublished reasoning behind the constants used in the 2d eccentric split case are briefly discussed later¹.

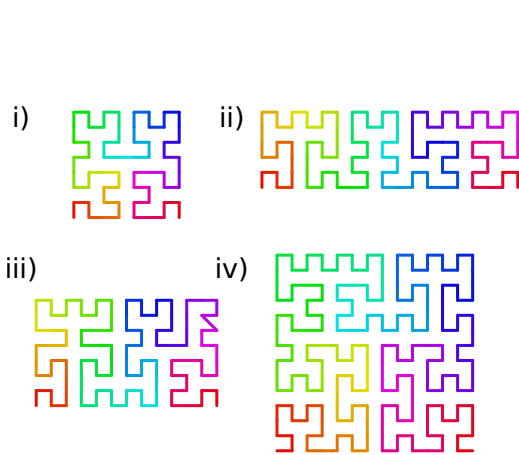


Figure 1: 2D Gilbert curves for i) 8×8 , ii) 18×6 ,
iv) 13×8 (with notch), iv) 14×14

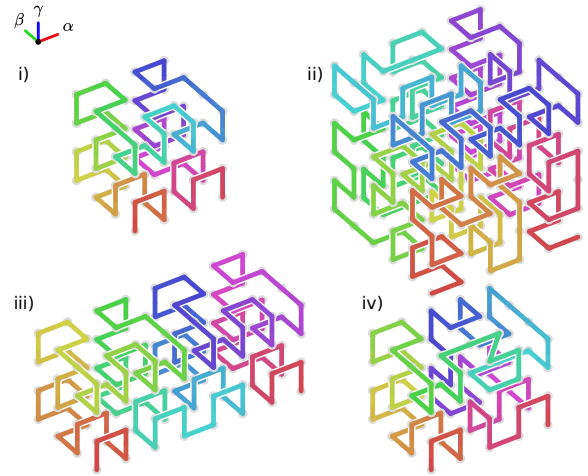


Figure 2: 3D Gilbert curves for i) $4 \times 4 \times 4$, ii)
 $6 \times 6 \times 6$, iii) $8 \times 4 \times 4$, iv) $5 \times 4 \times 4$
(with notch)

¹Through personal communication with the authors, Tautenhahn kindly provided the reasoning for the constants used in the eccentric split

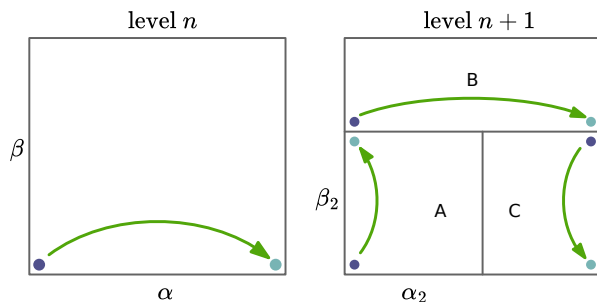


Figure 3: Subdivision template for the 2D Gilbert curve.

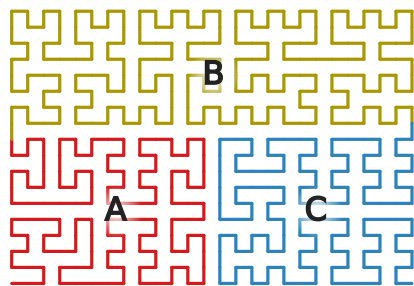


Figure 4: Example curve with subdivision regions highlighted.

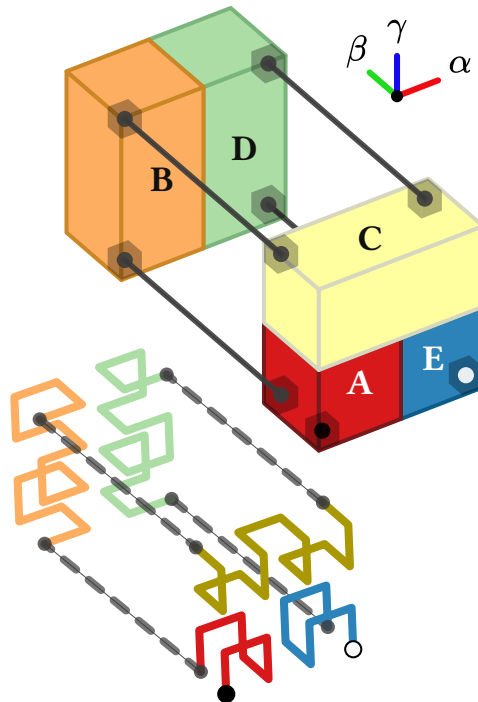


Figure 5: The main subdivision template for the 3D Gilbert curve.

Subdivision Templates

For both the 2D and 3D Gilbert curve, a subdivision template is chosen to recursively partition the region. Figure 3 shows the 2D template and figure 5 shows the 3D template.

The vectors $\alpha, \beta, \gamma \in \mathbb{Z}^3$ provide generalized notions of width (α), height (β), and depth (γ), so that we can transform rectangular or cuboid regions as necessary for the recursion. Each of α, β, γ will only have one non-zero component and will be orthogonal to each other, representing a current snapshot of the current frame of reference.

When dividing into sub-regions, side lengths are chosen to be integral and with even side lengths preferred for the first subdivided region (region A in figures 3 and 5). A path is recursively chosen for each of the subregions and, after resolution, endpoints are connected. Figure 6 shows the choice in 2D of even or odd regions depending on side length parity.

When regions have a side length that is much larger than the rest, another subdivision scheme is used. We call these cases *eccentric subdivisions*, as the larger side length makes the region lopsided.

If one side length is much larger than the rest, both the 2D and 3D Gilbert curve split the region in two. For the 3D case, if one side length is much smaller than the rest but with the remaining lengths roughly equal, the region is subdivided as if it were a 2D Gilbert curve, taking the smaller side length as the “flat” dimension.

The eccentric subdivision schemes provide a more visually pleasing solution as regimented subdivisions might take paths that are long and skinny or wide and squat. In the next section, we will briefly go over motivations for the threshold values of when to use the eccentric subdivision schemes.

For the 2D Gilbert curve, if $|\alpha|$ is odd, there will be a diagonal move, or *notch*, for the resulting path. The subdivision scheme will keep the notch in a single region, limiting the notch count to one.

Figure 1 illustrates the proof of Theorem 1, showing the construction of a path between two points in a grid. The figure is divided into four quadrants based on the parity of the dimensions h and w .

- odd/even:** Shows a grid of size $(2h+1) \times (4w)$. The grid is divided into two parts of size $(2h+1) \times (2w)$ each. The path starts at a blue dot and ends at a green dot, passing through a red dot.
- odd/odd:** Shows a grid of size $(2h+1) \times (2w+1)$. The grid is divided into two parts of size $(2h+1) \times (2w)$ each. The path starts at a blue dot and ends at a green dot, passing through a red dot.
- even/even:** Shows a grid of size $(4h) \times (4w)$. The grid is divided into two parts of size $(2h) \times (2w)$ each. The path starts at a blue dot and ends at a green dot, passing through a red dot.
- even/odd:** Shows a grid of size $(4h) \times (2w+1)$. The grid is divided into two parts of size $(2h) \times (2w)$ each. The path starts at a blue dot and ends at a green dot, passing through a red dot.

Eccentric Subdivision

For each eccentric subdivision, a proportion threshold needs to be chosen to determine when a side lengths become too lopsided to warrant an alternate subdivision. Here we motivate the eccentric subdivision constants for the 2D case.

$$\lambda_2(|\alpha|, |\beta|) = \frac{|\alpha| \cdot |\beta|}{\min(|\alpha|, |\beta|)^2}$$
$$\lambda_s(V_1) = \sum_k \frac{\text{Vol}(V_{0,k})}{\text{Vol}(V_0)} \cdot \lambda_2(V_{0,k})$$

The defect gives a coarse idea of how lopsided or *eccentric* a region is. If the defect is too high, we might want to split the larger sides while keeping the smaller sides the same size. Reducing the average defect attempts to make each subdivided region more square-like. When subdivided areas are more square-like, we say that the subdivided regions are more *harmonious*.

If the α side length is significantly longer than the β side length, we want to subdivide the rectangle into two nearly equal regions and recursively find a Gilbert curve in each region. We will justify the constant $(3/2)$ as the ratio threshold to split on, using an argument originally developed by L. Tautenhahn ¹.

The defect of a rectangle with side length $|\alpha|$ and $|\beta|$, assuming $|\alpha| > |\beta|$, is $(|\alpha|/|\beta|)$. After subdivision, if we assume $|\alpha| < 2|\beta|$, the resulting defect is $2|\beta|/|\alpha|$. For a defect reduction, we search for conditions in which the ratio of the resulting defect to the original defect is less than unity, giving the equation $|\alpha|/|\beta| > \sqrt{2}$.

Since $\sqrt{2} \approx 1.4142 < (3/2)$, if we choose $|\alpha|/|\beta| > (3/2)$ we can be assured a defect reduction. In the case when $|\alpha| > 2|\beta|$, it is easy to verify that the defect is always reduced ².

Defect reductions for the 3D case can be done, but are more complicated. The constants for the 3D subdivision in this paper were observed to yield visually pleasing results without concern for optimizing defect reduction.

Algorithm 1 Gilbert 2D

```

#  $p, \alpha, \beta \in \mathbb{Z}^3$ 
function GILBERT2D( $p, \alpha, \beta$ )
   $\alpha_2, \beta_2 = \text{div}(\alpha, 2), \text{div}(\beta, 2)$ 
  if  $(|\beta| \equiv 1)$  then
    yield  $p + i \cdot \delta(\alpha)$  forall  $i \in |\alpha|$ 
  else if  $(|\alpha| \equiv 1)$  then
    yield  $p + i \cdot \delta(\alpha)$  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

```

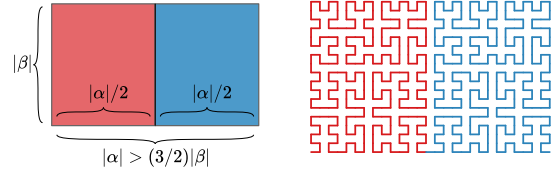


Figure 7

Figure 8

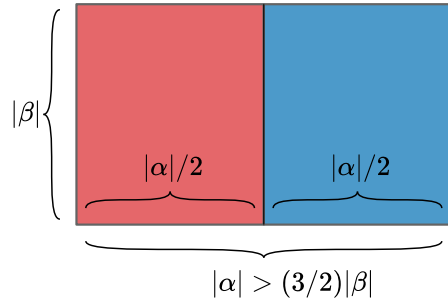


Figure 9

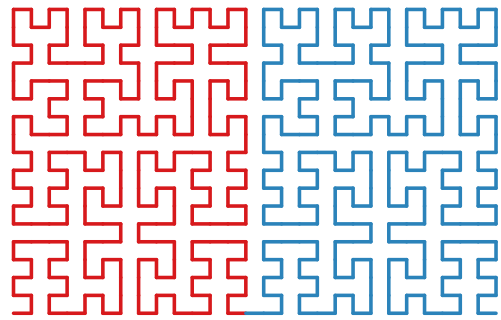


Figure 10

¹Given to us through personal communication with L. Tautenhahn

²The ratio of the defects before and after the split is $(1/2)$

Algorithm 2 Gilbert 3D

$p, \alpha, \beta, \gamma \in \mathbb{Z}^3$

function GILBERT3D(p, α, β, γ)

return Gilbert2D(p, β, γ) **if** ($|\alpha| \equiv 1$)

return Gilbert2D(p, α, γ) **if** ($|\beta| \equiv 1$)

return Gilbert2D(p, α, β) **if** ($|\gamma| \equiv 1$)

$\alpha_2 \leftarrow \text{div}(\alpha, 2) + \Delta(\alpha_2, \alpha)$

$\beta_2 \leftarrow \text{div}(\beta, 2) + \Delta(\beta_2, \beta)$

$\gamma_2 \leftarrow \text{div}(\gamma, 2) + \Delta(\gamma_2, \gamma)$

if ($2|\alpha| > 3|\beta|$) and ($2|\alpha| > 3|\gamma|$)

yield Gilbert3D($p, \alpha_2, \beta, \gamma$)

yield Gilbert3D($p + \alpha_2, \alpha - \alpha_2, \beta, \gamma$)

if ($3|\beta| > 4|\gamma|$)

yield Gilbert3D($p, \beta_2, \gamma, \alpha_2$)

yield Gilbert3D($p + \beta_2, \alpha, \beta - \beta_2, \gamma$)

yield Gilbert3D($p +$
 $(\alpha - \delta(\alpha)) +$
 $(\beta_2 - \delta(\beta)),$
 $-\beta_2, \gamma, -(\alpha - \alpha_2)$)

if ($3|\gamma| > 4|\beta|$)

yield Gilbert3D($p, \gamma_2, \alpha_2, \beta$)

yield Gilbert3D($p + \gamma_2, \alpha, \beta, \gamma - \gamma_2$)

yield Gilbert3D($p +$
 $(\alpha - \delta(\alpha))$
 $(\gamma_2 - \delta(\gamma)),$
 $-\gamma_2, -(\alpha - \alpha_2), \beta$)

else

yield Gilbert3D($p, \beta_2, \gamma_2, \alpha_2$)

yield Gilbert3D($p + \beta_2, \gamma, \alpha_2, \beta_2$)

yield Gilbert3D($p +$
 $(\beta_2 - \delta(\beta)) +$
 $(\gamma - \delta(\gamma)),$
 $\alpha, -\beta_2, -(\gamma - \gamma_2)$)

yield Gilbert3D($p +$
 $(\alpha_2 - \delta(\alpha)) +$
 $\beta_2 + (\gamma - \delta(\gamma)),$
 $-\gamma, -(\alpha - \alpha_2), (\beta - \beta_2)$)

yield Gilbert3D($p +$
 $(\alpha - \delta(\alpha)) +$
 $(\beta_2 - \delta(\beta)),$
 $-\beta_2, -\gamma_2, -(\alpha - \alpha_2)$)

end function

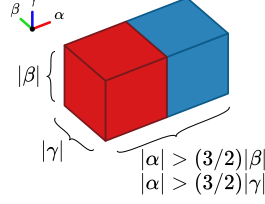


Figure 11

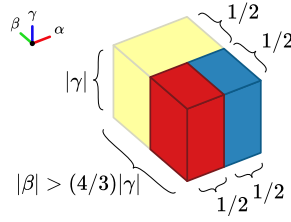


Figure 13

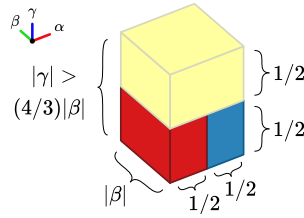


Figure 15

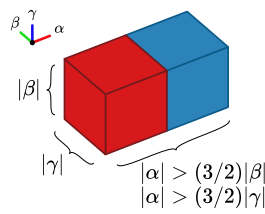


Figure 17

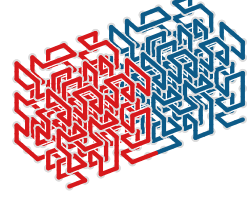


Figure 12

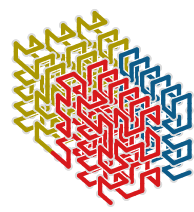


Figure 14



Figure 16

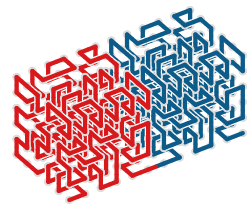


Figure 18

Algorithm

Algorithm 1 and algorithm 2 shows the pseudo-code for computing the Gilbert curve in 2D and 3D respectively. After each subdivision, the origion and local axes are rotated to align with the sub region and recursively solved. In the base case where there is a unit length side, a curve in one lower dimension is output (a line for 2D and a 2D Gilbert curve for 3D). If the width like dimension, α , is oblong relative to the height-like dimension, β , an eccentric template is used. For the 3D case, additional checks to see if the remaining side lengths are oblong and, if so, the appropriate eccentric curves are applied. Otherwise, the bulk recursion is executed.

Note that even in the 2D case, α and β are taken to be vectors in 3D. The generalization to 3D allows the Gilbert2D function to be used unaltered when the 3D Gilbert curve needs to trace out in-plane sub-curves.

The standard Euclidean two norm ($|v| = \sqrt{v_0^2 + v_1^2 + v_2^2}$) is used but abuses notation by allowing scalar to vector multiplication ($i \in \mathbb{Z}, v \in \mathbb{Z}^3, i \cdot v \rightarrow (i \cdot v_0, i \cdot v_1, i \cdot v_2)$). where the context is clear. The $\delta(\cdot)$ function returns one of the six directional vectors indicating which of the major signed axis aligned directions the input vector points to ($(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)$). The $\div(a, b)$ function is used to do integer division, rounding towards zero in the case of a fractional result. To make things more concise, a convenience function is used, $(\Delta(\rho_2, \rho) \stackrel{\text{def}}{=} \delta(\rho) \text{ if } |\rho_2| \equiv 1 \bmod 2 \text{ and } |\rho| > 2)$, that prefers even lengths in the non-degenerate case.

Random access functions to convert from index to position or position to index can be created by modifying the presented algorithms with short circuit conditionals, bypassing path calculation but keeping a running total of positions that would have been traversed. Each subregion is an axis-aligned rectangle or cuboid, allowing for easy calculation of path length as the product of the side lengths for the given area or volume.

Conclusion and Future Directions

The Gilbert curve presented in this paper provides one interpretation for a generalized Hilbert curve in 2D and 3D to arbitrary side lengths. Generalizing a Hilbert space filling curve to arbitrary side lengths is open to interpretation and the solution presented here is one such adaptation.

When desiging an algorithm to generalize the Hilbert curve, four features that might be desirable are:

<i>Hilbert-esque</i>	Reduces to a Hilbert curve when side lengths are equal and are exact powers of two
<i>Stabilty</i>	Curve realizations converge as the cuboid side lengths increase proportionally
<i>Harmony</i>	Realizations use eccentric splits to reduce the <i>defect</i>
<i>Notch-Limited</i>	Curve realizations are notch free or limited to a single notch if necessary

The 2D Gilbert curve presented here has all of these properties while the 3D Gilbert curve lacks harmony and the notch-limited feature.

Future work could focus on creating 3D Gilbert curve that has all of these features, perhaps by relaxing strict endpoint requirements at the cuboid corners. Some other natural ideas are to extend the concepts above to other space filling curves or circuits, or, more generally, to try and create an arbitrary space filling curve in a cuboid region with arbitrary endpoints.

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

³ <https://github.com/jakubcerveny/gilbert>

References

- [1] L. Tautenhahn. "Draw a space-filling curve of arbitrary size." 2003.
https://lutanho.net/pic2html/draw_sfc.html.
- [2] S. Anders. "Visualization of genomic data with the Hilbert curve." 2009.
10.1093/bioinformatics/btp152.
- [3] J. Zhang, S. Kamata, and Y. Ueshige. "A Pseudo-hilbert Scan Algorithm for Arbitrarily-Sized Rectangle Region." 10.1007/11821045 31.
- [4] A. Cortesi. "Visualizing binaries with space-filling curves." <https://corte.si/posts/visualisation/binvis/>.
- [5] H. Saga. "Space-Filling Curves." 9783540942658.