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, where diagonal moves are called *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 Tautenhahn's unpublished reasoning behind the constants used in the 2D eccentric split case are briefly discussed later[1].

.

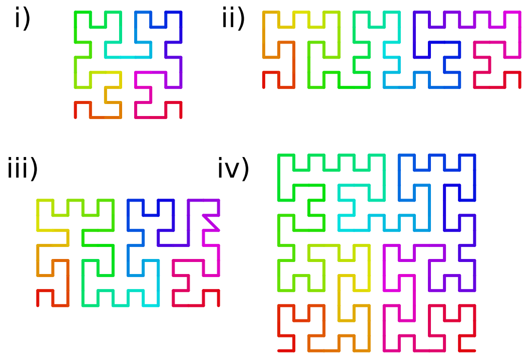

Figure 1: *2D Gilbert curves for i)* 8×8, *ii)* 18×6, *iv)* 13×8 *(with notch), iv)* 14×14

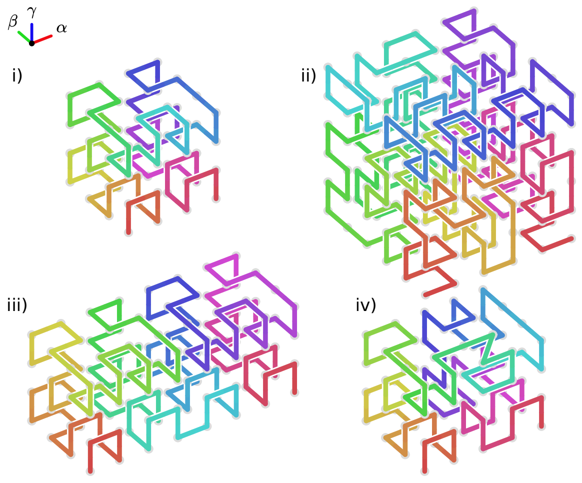

Figure 2: *3D Gilbert curves for i)* $4 \times 4 \times 4$, *ii)* $6 \times 6 \times 6$, *iii)* $8 \times 4 \times 4$, *iv)* $5 \times 4 \times 4$ *(with notch)*

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

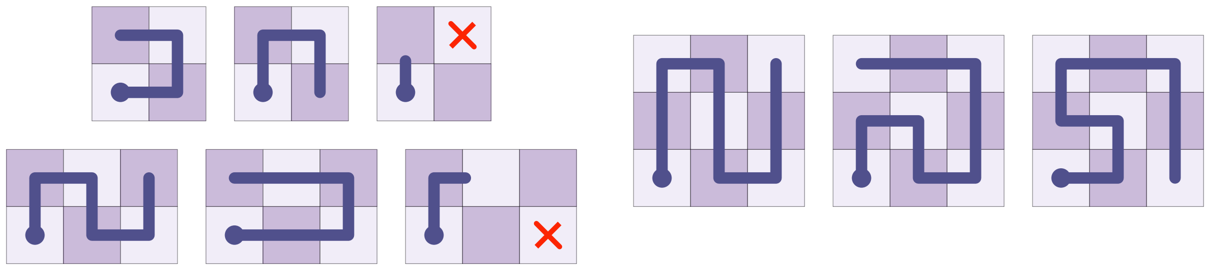

Figure 3: *Examples of Hamiltonian paths for small grid sizes. A red 'x' corresponds to no possible path for the chosen endpoints.*

Valid Paths from Grid Parity

The feasibility of determining whether there exists a non intersecting path, called a *Hamiltonian path*, connecting endpoints on the corners in a rectangular cuboid grid region can be accomplished through parity arguments. Label grid cell points in a volume as 0 or 1, alternating between labels with every axis-aligned single step move. Any Hamiltonian path that ends at one of the three remaining corners has to have the same parity as the starting point if the volume is odd, or different parity if the volume is even.

Figure 3 illustrates this for starting position $(0, 0)$ with areas (2×2), (3×2) and (3×3), where a red cross indicating a precluded endpoint. When a Hamiltonian path is not precluded by the parity of the start and end point, we say that the endpoints are *color compatible*. In general, color compatibility is a necessary, but insufficient, test for a Hamiltonicity.

In the simpler case when the region is a rectangular cuboid and endpoints are on the corners of the cuboid region, color compatibility is a sufficient test for Hamiltonicity. This allows us a simple test to see if there is a Hamiltonian path within a rectangular region purely based on the parity of side lengths and the corner locations of the start and end points of the path.

For the constructed 2D and 3D Gilbert curves in this paper, we don't use endpoints that are diagonal within a cuboid region and only use endpoints that are in line in a single axis-aligned direction.

Subdivision Templates

For both the 2D and 3D Gilbert curve, a subdivision template is chosen to recursively partition the region. Figure 4 shows the 2D template and figure 6 shows the 3D template. Figure 5 provides a 2D example of the main subdivision with regions labelled and color coded.

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 snapshot of the current frame of reference.

When dividing into sub-regions, side lengths are chosen to be integral. Even side lengths are preferred for the first subdivided region (region A in figures 4 and 6). A path is recursively chosen for each of the sub-regions and, after resolution, endpoints are connected. Figure 7 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.

The eccentric subdivision schemes provide a more visually pleasing partition of the space, as regimented subdivisions might take paths that are long and skinny or wide and squat. In the next section, We will briefly

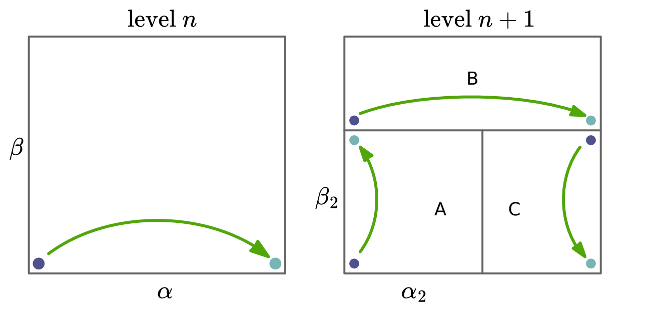

Figure 4: *Subdivision template for the 2D Gilbert curve.*

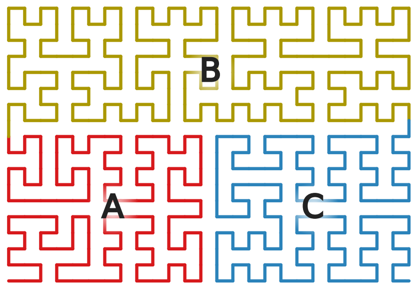

Figure 5: *Example curve with subdivision regions highlighted.*

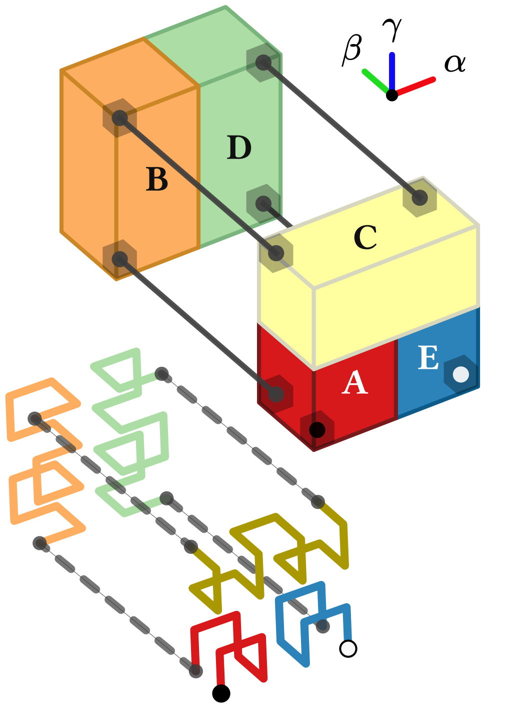

Figure 6: *The main subdivision template for the 3D Gilbert curve.*

go over motivations for the threshold values for the 2D eccentric subdivision scheme in the next section but first provide a more detailed desciption of the template subdivision schemes for the 2D and 3D Gilbert curve below.

2D Gilbert Curve

For the 2D Gilbert curve, if the width like dimension (α) is much larger, the curve is split in half along the α length (figure 8, 9). If no eccentric split is done, the region is subdivided into two squares at the bottom and a rectangle on top, with endpoints joined on the outer perimeter.

Each regions parity is chosen so as to try to keep color compatibility, ensuring a valid path if possible. When the side lengths are exact powers of two, the resulting curve is identical to the 2D Hilbert curve.

If the region is odd parity, with both side lengths odd, a notch is forced and is directed to the upper right region, as illustrated by figure 7.

3D Gilbert Curve

In the normal case, a subdivision scheme is used to split the cuboid into five regions, two cube like regions where the path starts and stops, and three oblong cuboid regions for the middle portion of the path. Figure 6 provides an exploded view of the main subdivision and where the endpoints line up. See also figure for the template and figure for a $10 \times 10 \times 10$ example, with each region color coded.

Endpoints within a sub-divided region are kept on the exterior of the region and joined after the resulting recursion has completed. As with the 2D Gilbert, when side dimensions are equal and exact powers of two, the resulting curve is identical to the corresponding 3D Hilbert curve.

Durcing the course of sub-division, if the width like dimension (α) is much larger, the curve is split in half along the α length (figure 12, 13). If the α eccentric split is not triggered, the height-like dimension, β, is then compared to depth-like dimension, γ and if very much larger, an eccentric subdivision scheme is used that splits in the α and β direction (figure ,). If both α and β are not much larger, a check is done to see if the depth-like dimension, γ, is much larger than than the height-like dimension, β, splitting in the α and γ direction if so (figure ,).

If any of the subdivided regions has endpoints that lay in a direction with odd length, a notch will appear. This means the number of notches for the 3D Gilbert curve isn't limited to one in the case a side length is odd.

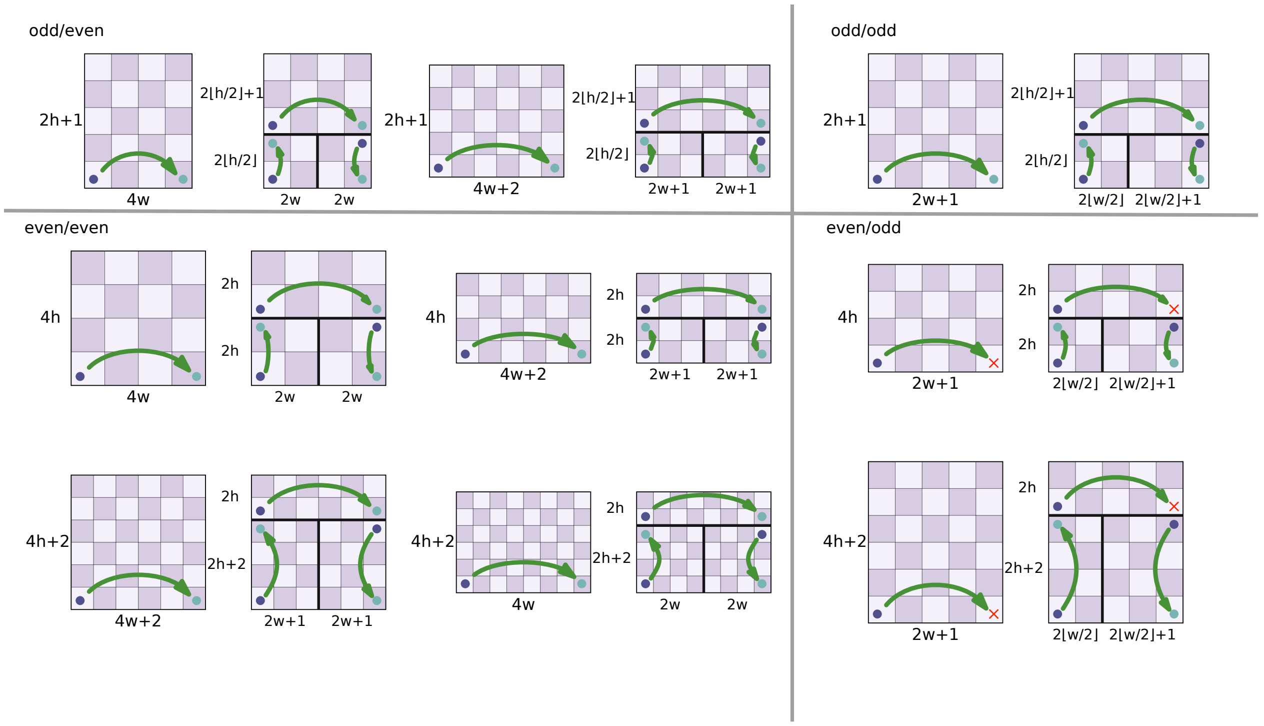

Figure 7: *Enumeration of the subdivision template depending on different parities of α and β dimensions.*

Eccentric Subdivision

When side lengths become too lopsided, an alternate *eccentric* subdivision scheme is chosen to split along the larger side length.

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.

Consider a *defect* function, $\lambda_2 : \mathbb{N}^2 \mapsto \mathbb{N}$, that measures the area relative to what the area would be if just the minimum side length were taken:

$$\lambda_2(|\alpha|, |\beta|) = \frac{|\alpha| \cdot |\beta|}{\min(|\alpha|, |\beta|)^2}$$

If there is a disjoint subdivision of a volume V_0 to $V_1 = (V_{0,0}, V_{0,1}, \ldots, V_{0,m-1})$, ($V_0 = \cup_k V_{0,k}$), define the subdivided volume's *average defect* to be the sum of defects weighted by their proportional volume:

$$\lambda_s(V_1) = \sum_k \frac{\text{Vol}(V_{0,k})}{\text{Vol}(V_0)} \cdot \lambda_2(V_{0,k})$$

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
```

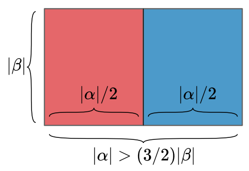

Figure 8

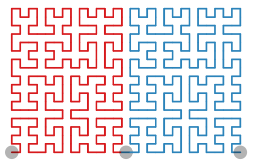

Figure 9

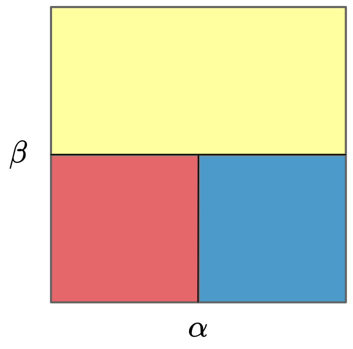

Figure 10

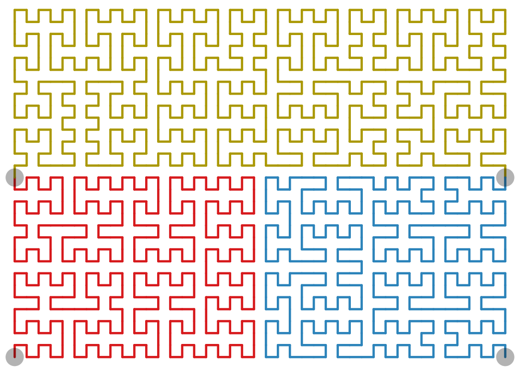

Figure 11

[1]Given to us through personal communication with L. Tautenhahn

[2]The 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($p$, $\alpha$, $\beta$, $\gamma$)
    return Gilbert2D($p$,$\beta$,$\gamma$) if ($|\alpha| \equiv 1$)
    return Gilbert2D($p$,$\alpha$,$\gamma$) if ($|\beta| \equiv 1$)
    return Gilbert2D($p$,$\alpha$,$\beta$) 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
```

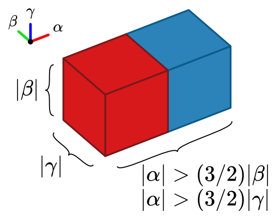

Figure 12

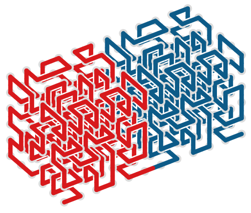

Figure 13

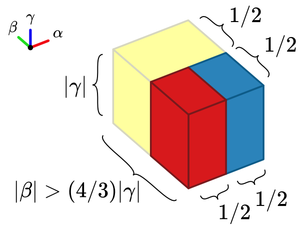

Figure 14

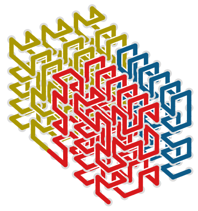

Figure 15

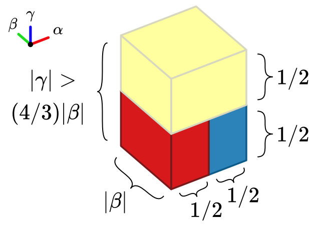

Figure 16

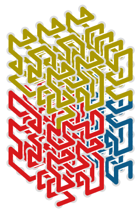

Figure 17

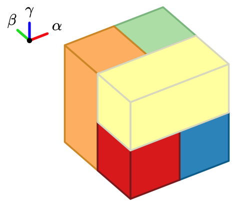

Figure 18

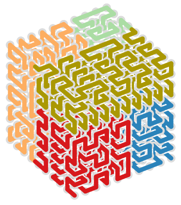

Figure 19

Algorithm

Algorithm 1 and algorithm 2 shows the pseudo-code for computing the Gilbert curve in 2D and 3D respectively. After each subdivision, the origin and local axis 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, $(\Delta(\rho_2, \rho) \overset{\text{def}}{=} \delta(\rho)$ **if** $|\rho_2| \equiv 1 \bmod 2$ 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 sub-region 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 designing an algorithm to generalize the Hilbert curve, four features that might be desirable are:

Hilbert-esque	Reduces to a Hilbert curve when side lengths are equal and are exact powers of two
Stability	Curve realizations converge as the cuboid side lengths increase proportionally
Harmony	Realizations use eccentric splits to reduce the *defect*
Notch-Limited	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.