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 locality, or notions of distance. 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. The construction provides reasonable worst case run-time guarantees for random access lookups for both two and three dimensions. We provide experiments showing comparable quality in locality of the Gilbert curve to the Hilbert curve for a range of cuboid regions and discuss some limitations of the construction algorithm.

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 (*notches*).

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 extend them to 3D. Tautenhahn's unpublished reasoning behind the constants used in the 2d eccentric split case are briefly discussed later in this paper [1].
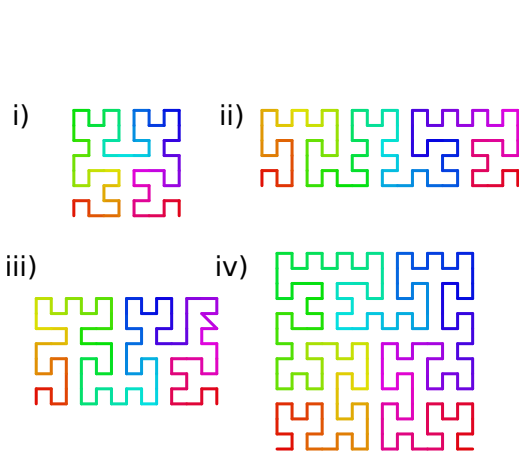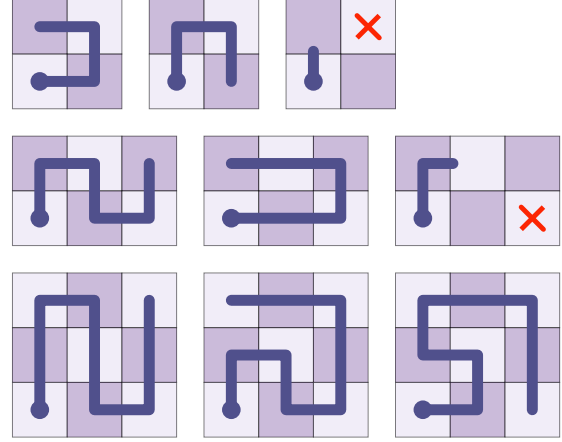

Figure 1: *2D Gilbert curves for i)* 8×8, *ii)* 18×6, *iv)* 13×8 *(with notch), iv)* 14×14
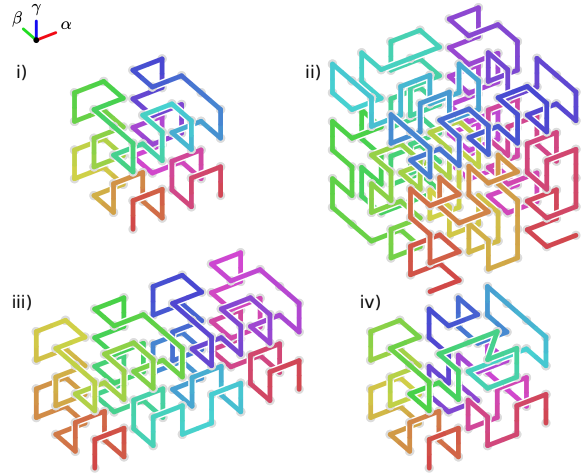
Figure 2: *3D Gilbert curves for i)* $4 \times 4 \times 4$, *ii)* $6 \times 6 \times 6$, *iii)* $8 \times 4 \times 4$, *iv)* $5 \times 4 \times 4$ *(with notch)*.

Path Possible	Volume	
	even	*odd*
$\|\alpha\|$ *even*	Yes	Yes
$\|\alpha\|$ *odd*	**No**	Yes

Figure 3: *$|\alpha|$ is the distance of endpoints. A Hamiltonian path is possible only when $|\alpha|$ is even or both $|\alpha|$ and the volume are odd.*

Figure 4: *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 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.

For a path starting at $(0, 0, 0)$ and ending $|\alpha|$ steps in one of the axis-aligned dimensions, then Table 3 enumerates this condition under which a valid path is possible. Figure 4 illustrates this for starting position $(0, 0)$ with areas (2×2), (3×2) and (3×3), where a red cross indicating a precluded endpoint.

Without loss of generality, we will assume a curve starts from position $p_s = (0, 0, 0)$ and has proposed endpoint at $p_e = ((w - 1), 0, 0)$, with a cuboid region as $\alpha = (w, 0, 0), \beta = (0, h, 0), \gamma = (0, 0, d)$. We state, without proof, that a Hamiltonian path is always possible from p_s to p_e when $|\alpha|$ is even or when all of $|\alpha|$, $|\beta|$ and $|\gamma|$ are odd ($|\alpha| \cdot (1 - |\beta| \cdot |\gamma|) \equiv 0 \bmod 2$).

When the condition ($|\alpha| \cdot (1 - |\beta| \cdot |\gamma|) \equiv 0 \bmod 2$) is met ($|\alpha|$ even or all of $|\alpha|, |\beta|, |\gamma|$ odd) any cuboid subdivision will always have a Hamiltonian path feasible within it. We can recreate a Hamiltonian path in the larger cuboid region by connecting neighboring endpoints in each of the adjacent subdivided regions.

For cuboids that violate this condition, there will be at least one notch. The 2D Gilbert curve limits the number of notches to one. The 3D Gilbert curve's subdivision strategy creates a notch when the distance between endpoints is odd, potentially creating more than one notch.

In both the 2D and 3D case, when the original side lengths are all even, no notches will be present.

2D Gilbert Curve Algorithm

Algorithm **??** shows the pseudo-code for computing the 2D Gilbert curve. Note that α and β are taken to be vectors in 3D, where the third dimension can be ignored if a purely 2D curve is desired. The generalization to

---

[1]Through personal communication with the authors, Tautenhahn kindly provided the reasoning for the constants used in the eccentric split
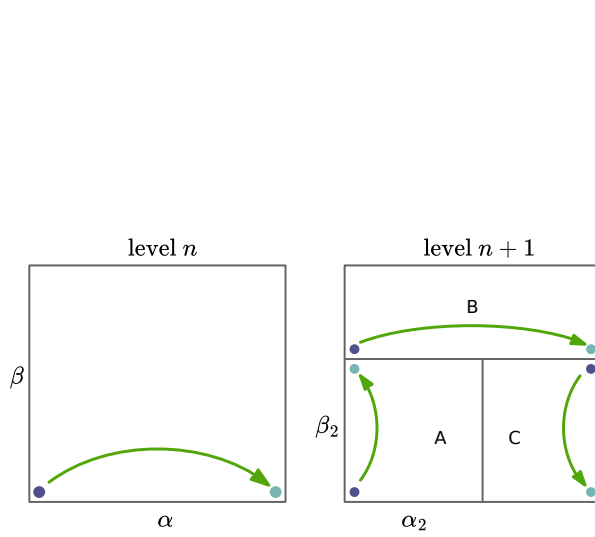
**Figure 5:** *Subdivision strategy for the 2D Gilbert curve.*
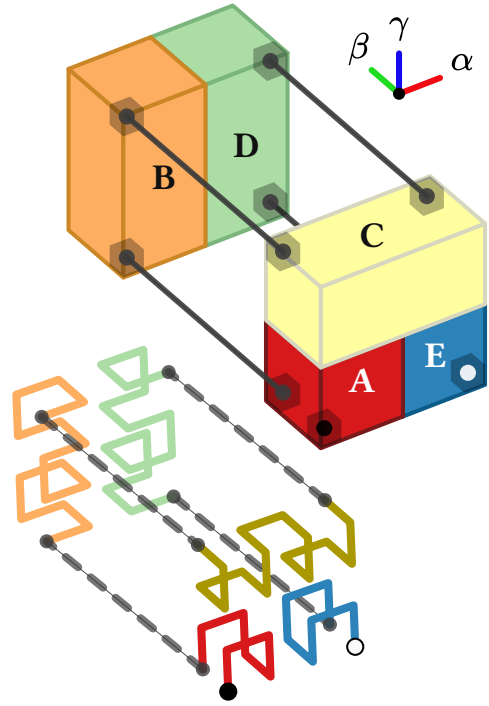
Figure 6: *The main subdivision strategy for the 3D Gilbert curve.*

3D allows the Gilbert2D function to be used unaltered when the 3D Gilbert curve needs to trace out in-plane sub-curves.

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 $((\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1))$. For completeness, the function is defined in Procedure **??** in Appendix **??**.

Algorithm **??** assumes standard Euclidean two norm $(|v| = \sqrt{v_0^2 + v_1^2 + v_2^2})$ and abuses notation by allowing scalar to vector multiplication $(i \in \mathbb{Z}, v \in \mathbb{Z}^3, i \cdot v \to (i \cdot v_0, i \cdot v_1, i \cdot v_2))$. where the context is clear.
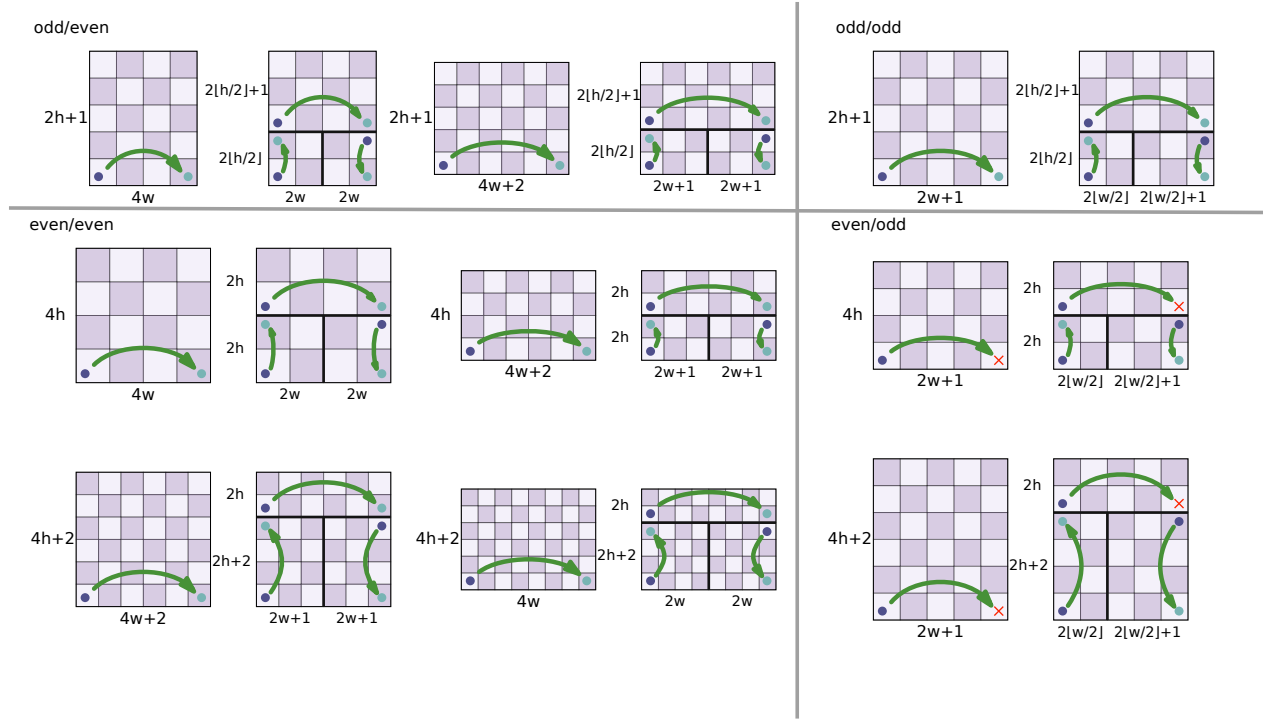
Figure 7: *Enumeration of the subdivision template depending on different parities of α and β dimensions.*

```
# p, α, β ∈ ℤ³
function GILBERT2D(p, α, β)
    α₂, β₂ = div(α, 2), div(β, 2)
    if (|β| ≡ 1) then
        yield p + i · δ(α) forall i ∈ |α|
    else if (|α| ≡ 1) then
        yield p + i · δ(α) forall i ∈ |β|
    else if (2|α| > 3|β|) then
        if (|α₂| > 2) and (|α₂| mod 2 ≡ 1) then
            α₂ ← α₂ + δ(α)
        end if
        yield Gilbert2D(p, α₂, β)
        yield Gilbert2D(p + α₂, α − α₂, β)
    else
        if (|β₂| > 2) and (|β₂| mod 2 ≡ 1) then
            β₂ ← β₂ + δ(β)
        end if
        yield Gilbert2D(p,
                        β₂, α₂)
        yield Gilbert2D(p + β₂,
                        α, (β − β₂))
        yield Gilbert2D(p + α − δ(α) + β₂ − δ(β),
                        β₂, −(α − α₂))
    end if
end function
```

```
# p, α, β, γ ∈ ℤ³
function GILBERT3D(p, α, β, γ)

    # Parity of dimensions
    α₀ ← (|α| mod 2)
    β₀ ← (|β| mod 2)
    γ₀ ← (|γ| mod 2)

    # Base cases
    if ((|α| ≡ 2) and (|β| ≡ 2) and (|γ| ≡ 2))
        return Hilbert3D(p,α,β,γ)
    return Gilbert2D(p,β,γ) if (|α| ≡ 1)
    return Gilbert2D(p,α,γ) if (|β| ≡ 1)
    return Gilbert2D(p,α,β) if (|γ| ≡ 1)

    # Eccentric cases
    if (3|α| > 5|β|) and (3|α| > 5|γ|))
        return S₀(p,α,β,γ)
    if (2|β| > 3|γ|) or (2|β| > 3|α|))
        return S₂(p,α,β,γ)
    if (2|γ| > 3|β|)
        return S₁(p,α,β,γ)

    # Bulk recursion
    return J₀(p,α,β,γ) if (γ₀ ≡ 0)
    return J₁(p,α,β,γ) if (α₀ ≡ 0)or(β₀ ≡ 0)
    return J₂(p,α,β,γ)
end function
```

References

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