On the Manhattan-Distance Between Points on Space-Filling Mesh-Indexings

Rolf Niedermeier* Peter Sanders†

May 23, 1996

Abstract

Indexing schemes based on space filling curves like the Hilbert curve are a powerful tool for building efficient parallel algorithms on mesh-connected computers. The main reason is that they are locality-preserving, i.e., the Manhattan-distance between processors grows only slowly with increasing index differences. We present a simple and easy-to-verify proof that the Manhattan-distance of any indices i and j is bounded by $3\sqrt{|i-j|}-2$ for the 2D-Hilbert curve. The technique used for the proof is then generalized for a large class of self-similar curves. We use this result to show a (quite tight) bound of $4.73458\sqrt[3]{|i-j|}-3$ for a 3D-Hilbert curve.

1 Introduction

It has become increasingly clear that mesh-connected processor arrays, grids for short, are among the most realistic models of parallel computation [1, 4, 14, 18]. The indexing of the processors is an important aspect in the design of mesh algorithms. Several indexing schemes are well-known. Most of these have been developed for the two-dimensional case but they usually have generalizations for multiple dimensions, for example, row-major or snakelike row-major [15]. However, these kinds of indexings do not preserve locality of computation and communication very well. So, e.g., for an r-dimensional grid with side length n and row-major indexing, processors 1 and n are at distance $n-1$ from each other. This is large compared to the distance of $\approx r\sqrt[r]{n}$ achievable if the first n processors could be arranged in a cube. A locality-preserving indexing should yield a Manhattan-distance of $O(\sqrt[r]{n})$. This should generalize to all pairs of processors within the grid, that is, processors indexed i and j should be at distance $O(\sqrt[r]{|i-j|})$ from each other.

Stout [17] seems to be the first who used so-called *proximity ordering*s in the context of 2D-grid algorithms. We subsequently call them *Hilbert indexing*s due to the direct relation to Hilbert's space-filling curve [7, 13]. According to Stout "there is a constant $c < 4$ such that processors numbered i and j are no more than $c \cdot \sqrt{|i-j|}$ communication links apart" [17, page 27]. Later, Miller and Stout [10] made use of Hilbert processor indexings for mesh computer algorithms in computational geometry. Recently, Kaklamanis and Persiano [9], seemingly not knowing of Miller and Stout's work, made use of Hilbert indexings to devise efficient backtrack and branch-and-bound search algorithms for grids. They showed that two arbitrary processors indexed with i and j have a distance of at most $4 \cdot \sqrt{|i-j|}$ from each other. This is the upper bound already claimed by Stout [17]. Very recently, a bound of $3 \cdot \sqrt{|i-j|}$ has been proved by Chochia, Cole, and Heywood [2]. However, the proof is quite complicated. Many cases of the proof are left to the reader and it is not even trivial to figure

*Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Sand 13, D-72076 Tübingen, niedermr@informatik.uni-tuebingen.de

†Fakultät für Informatik, Universität Karlsruhe, D-76128 Karlsruhe, sanders@ira.uka.de

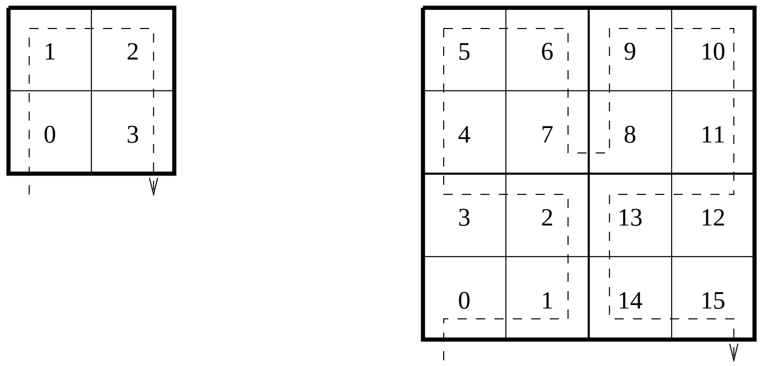

Figure 1: Hilbert indexings for grids of size 4 and 16.

out which cases have to be considered in order to complete the proof. In parallel to [2] we had developed a similar result which dissatisfied us for the same reasons.

Here we start by introducing the 2D-Hilbert indexing in Section 2. In Section 2.1 we identify segments (i.e., pairs of indices) which have a distance of $3\sqrt{|i-j|}-2$. In Section 2.2 we give a simple, complete, and easy-to-verify proof that this is also an upper bound. The methods used in this proof are generalized for higher dimensions and different recursion schemes in Section 3. In Section 4 the generalized method is applied to find bounds for a 3D-Hilbert indexing. Section 5 summarizes the results and points out some possible future developments.

2 The 2D-Hilbert indexing

The Hilbert indexing is a processor indexing for two-dimensional grids that in a sense preserves locality. Informally speaking, the optimization goal is, given two arbitrary processors with numbers i and j from $\{0, 1, \ldots n^2-1\}$, find an indexing such that the smaller $|i-j|$, the smaller the Manhattan-distance between i and j within the grid is.

The problem can be attacked by an inductive approach. We build indexings for larger and larger grids, starting with indexings for 2×2-grids and always put together four smaller grids to get the next larger one. Thus we obtain processor indexings for grids of size $n = 4^k$ for arbitrary $k \geq 1$. Iterating this process finitely often and considering the processor indexings as curves through the grid, this leads to "finite cases" of Hilbert's space-filling curve, known for slightly more than 100 years in mathematics [13].* In Figure 1 we present what we call *Hilbert processor indexing* for grids of size 4 and 16. For simplicity, we use the terms Hilbert curve and Hilbert indexing interchangeably when the meaning is clear from context. We get the following inductive construction principle. To obtain an indexing for a grid of size 4^k from the indexing for grids of size 4^{k-1}, proceed as follows. First observe that drawing the curve that belongs to the 2×2-grid of Figure 1, one gets a horseshoe-shaped curve. In a sense, this shape can also be discovered in the 4×4-grid of Figure 1 if one only considers the four 2×2-subgrids that build the 4×4-grid. First, the curve goes through the lower left, then through the upper left, then through the upper right, and finally through the lower right subgrid. In this way, larger and larger indexing curves can be constructed as shown in Figure 2.

In a sense, the Hilbert curve yields the "simplest" self-similar, recursive, locality-preserving

*Note that in this case it is assumed that we perform this iterative construction e.g. within the unit square. This justifies the naming space-filling curve, because in the limit the described curve "fills" the unit square (see [13] for details).

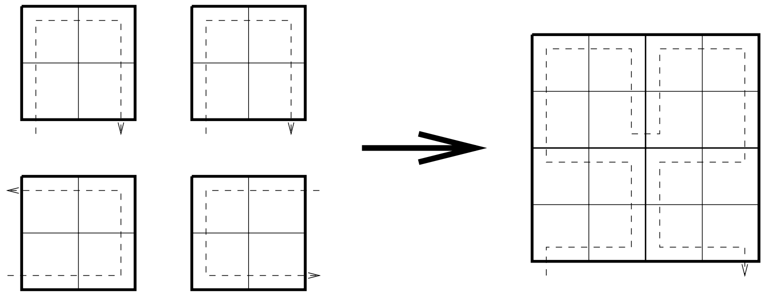

Figure 2: Inductive construction principle of Hilbert curves.

numbering scheme for square grids of size $2^k \times 2^k$. Still simpler recursive schemes can be defined which do not require rotation or reflection of the building blocks; but these are not locality-preserving. But if one insists on using identical (up to rotation and reflection) building blocks which are assembled using a fixed rule, the Hilbert curve is unique. In other articles the name Peano curve can be found which we consider less appropriate because this name is usually used in related construction schemes based on $3^k \times 3^k$ grids [13].

2.1 Lower bound

The following theorem provides a lower bound for the Manhattan-distance of two points on the Hilbert curve by giving "worst case pairs." In the next section we will show that this lower bound is tight by giving a matching upper bound.

Theorem 1. *For all $k \geq 1$, there are points i and j on the Hilbert curve such that $|i-j| = 4^{k-1}$ and the Manhattan-distance of i and j is exactly $3\sqrt{|i-j|} - 2 = 3 \cdot 2^{k-1} - 2$.*

Proof. Consider Figure 3. It shows parts of the Hilbert curve (rotated right by 90 degrees compared to Figures 1 and 2.)

In the following we argue that the points i and j in the lower left and upper right corner of the shaded area of Figure 3 do have Manhattan-distance $3\sqrt{|i-j|} - 2$ from each other, proving our claim. Let us first explain a bit further the idea behind Figure 3. The shaded area shall denote all points on the Hilbert curve lying between i and j. We always draw the largest subgrids filled by the Hilbert curve on the way from i to j. In this sense, the dotted line represents the path of the Hilbert curve respecting the sizes of the largest subgrids it passes through.

The central observation is that except for the lower left corner and upper right corner we have exactly three subgrids of each possible size within the shaded area. Note that, due to the inductive construction principle of Hilbert curves, this holds for $2^k \times 2^k$-subgrids for all integers $k \geq 0$. So we can count the number of points on the Hilbert curve lying between i and j. All in all, we get

$$3 \cdot \sum_{l=0}^{k-2} 4^l = 3 \cdot \frac{4^{k-1} - 1}{3} = 4^{k-1} - 1$$

points lying between i and j. This implies that $|i-j| = 4^{k-1}$. Computing

$$3\sqrt{|i-j|} - 2 = 3 \cdot 2^{k-1} - 2,$$

we exactly obtain the Manhattan-distance of i and j, where the latter can easily be read from Figure 3. $\square$

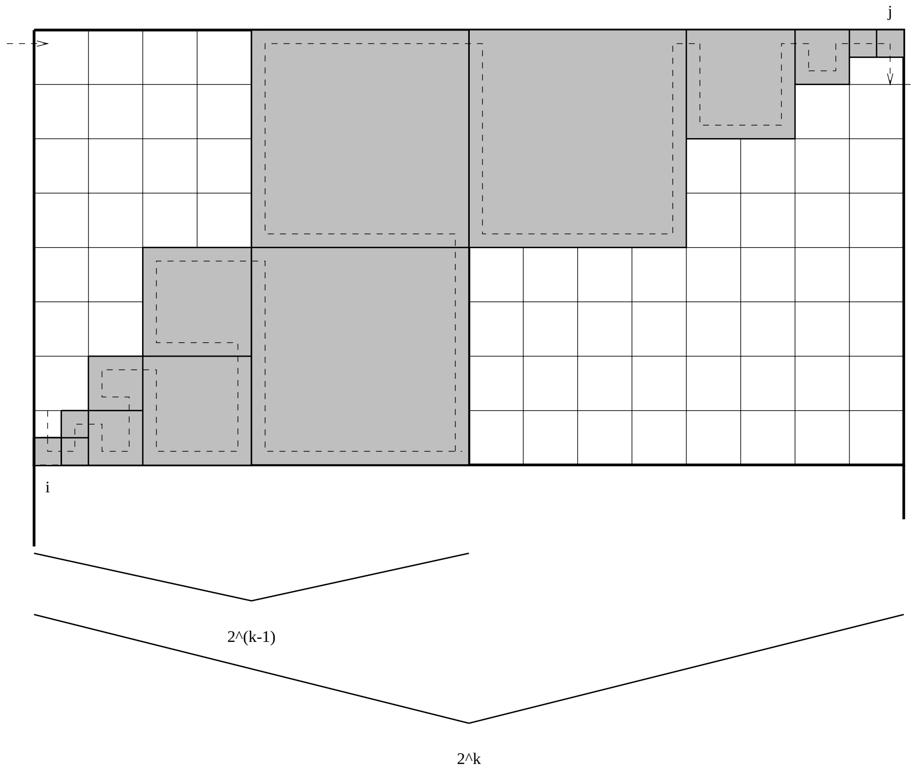

Figure 3: Worst-case for the Manhattan-distance between two points i and j.

Note that Theorem 1 in particular shows that there are worst case nodes for each "order of magnitude" of Manhattan-distances, that is, distances of the form $3 \times 2^{k-1} - 2$ for all positive integers k.

2.2 Upper bound

Theorem 2. *For the Manhattan-distance $d(i,j)$ of two arbitrary points i and j on the Hilbert curve with $i \neq j$ we have $d(i,j) \leq 3\sqrt{|i-j|} - 2$.*

Proof. The basic idea is to exploit the self-similarity of the Hilbert curve for an inductive proof over $|i-j|$. In principle, the proof is quite simple. However, it turns out that a special treatment is necessary for "small" grids and for indices i and j which are close to the worst case described in Theorem 1.

Case $|i-j| < 16$:

By inspection. All possible orientations of segments of length up to 16 can be found in an 8×8 mesh or in the upper left corner of Figure 4. (We have written a small computer program which automatically makes the necessary checks.)

Case $|i-j| \geq 16$:

By induction over $|i-j|$ we prove the following stronger statement: $d(i,j) \leq 3\sqrt{|i-j|} - 2.5$ or i and j are arranged as in Theorem 1 (Figure 3) and $d(i,j) = 3\sqrt{|i-j|} - 2$.

Basis of induction: $16 \leq |i-j| \leq 256$

By inspection. All possible orientations of segments of length up to 256 can be found in the 32×32-grid numbered in Figure 4. This can be done with the simple program mentioned

0	1	14	15	16	19	20	21	234	235	236	239	240	241	254	255	256	259	260	261	314	315	316	319	320	321	334	335	336	339	340	341
3	2	13	12	17	18	23	22	233	232	237	238	243	242	253	252	257	258	263	262	313	312	317	318	323	322	333	332	337	338	343	342
4	7	8	11	30	29	24	25	230	231	226	225	244	247	248	251	270	269	264	265	310	311	306	305	324	327	328	331	350	349	344	345
5	6	9	10	31	28	27	26	229	228	227	224	245	246	249	250	271	268	267	266	309	308	307	304	325	326	329	330	351	348	347	346
58	57	54	53	32	35	36	37	218	219	220	223	202	201	198	197	272	273	286	287	288	289	302	303	378	377	374	373	352	355	356	357
59	56	55	52	33	34	39	38	217	216	221	222	203	200	199	196	275	274	285	284	291	290	301	300	379	376	375	372	353	354	359	358
60	61	50	51	46	45	40	41	214	215	210	209	204	205	194	195	276	279	280	283	292	295	296	299	380	381	370	371	366	365	360	361
63	62	49	48	47	44	43	42	213	212	211	208	207	206	193	192	277	278	281	282	293	294	297	298	383	382	369	368	367	364	363	362
64	67	68	69	122	123	124	127	128	131	132	133	186	187	188	191	490	489	486	485	474	473	470	469	384	385	398	399	400	403	404	405
65	66	71	70	121	120	125	126	129	130	135	134	185	184	189	190	491	488	487	484	475	472	471	468	387	386	397	396	401	402	407	406
78	77	72	73	118	119	114	113	142	141	136	137	182	183	178	177	492	493	482	483	476	477	466	467	388	391	392	395	414	413	408	409
79	76	75	74	117	116	115	112	143	140	139	138	181	180	179	176	495	494	481	480	479	478	465	464	389	390	393	394	415	412	411	410
80	81	94	95	96	97	110	111	144	145	158	159	160	161	174	175	496	499	500	501	458	459	460	463	442	441	438	437	416	419	420	421
83	82	93	92	99	98	109	108	147	146	157	156	163	162	173	172	497	498	503	502	457	456	461	462	443	440	439	436	417	418	423	422
84	87	88	91	100	103	104	107	148	151	152	155	164	167	168	171	510	509	504	505	454	455	450	449	444	445	434	435	430	429	424	425
85	86	89	90	101	102	105	106	149	150	153	154	165	166	169	170	511	508	507	506	453	452	451	448	447	446	433	432	431	428	427	426
938	937	934	933	922	921	918	917	874	873	870	869	858	857	854	853	512	515	516	517	570	571	572	575	576	577	590	591	592	595	596	597
939	936	935	932	923	920	919	916	875	872	871	868	859	856	855	852	513	514	519	518	569	568	573	574	579	578	589	588	593	594	599	598
940	941	930	931	924	925	914	915	876	877	866	867	860	861	850	851	526	525	520	521	566	567	562	561	580	583	584	587	606	605	600	601
943	942	929	928	927	926	913	912	879	878	865	864	863	862	849	848	527	524	523	522	565	564	563	560	581	582	585	586	607	604	603	602
944	947	948	949	906	907	908	911	880	883	884	885	842	843	844	847	528	529	542	543	544	545	558	559	634	633	630	629	608	611	612	613
945	946	951	950	905	904	909	910	881	882	887	886	841	840	845	846	531	530	541	540	547	546	557	556	635	632	631	628	609	610	615	614
958	957	952	953	902	903	898	897	894	893	888	889	838	839	834	833	532	535	536	539	548	551	552	555	636	637	626	627	622	621	616	617
959	956	955	954	901	900	899	896	895	892	891	890	837	836	835	832	533	534	537	538	549	550	553	554	639	638	625	624	623	620	619	618
960	961	974	975	976	979	980	981	810	811	812	815	816	817	830	831	746	745	742	741	730	729	726	725	640	641	654	655	656	659	660	661
963	962	973	972	977	978	983	982	809	808	813	814	819	818	829	828	747	744	743	740	731	728	727	724	643	642	653	652	657	658	663	662
964	967	968	971	990	989	984	985	806	807	802	801	820	823	824	827	748	749	738	739	732	733	722	723	644	647	648	651	670	669	664	665
965	966	969	970	991	988	987	986	805	804	803	800	821	822	825	826	751	750	737	736	735	734	721	720	645	646	649	650	671	668	667	666
1018	1017	1014	1013	992	995	996	997	794	795	796	799	778	777	774	773	752	755	756	757	714	715	716	719	698	697	694	693	672	675	676	677
1019	1016	1015	1012	993	994	999	998	793	792	797	798	779	776	775	772	753	754	759	758	713	712	717	718	699	696	695	692	673	674	679	678
1020	1021	1010	1011	1006	1005	1000	1001	790	791	786	785	780	781	770	771	766	765	760	761	710	711	706	705	700	701	690	691	686	685	680	681
1023	1022	1009	1008	1007	1004	1003	1002	789	788	787	784	783	782	769	768	767	764	763	762	709	708	707	704	703	702	689	688	687	684	683	682

Figure 4: Hilbert indexing for a 32×32-grid. The largest worst-case segments are 85–341 and 682–938.

before.

Inductive step for $|i - j| > 256$:

We look at the "coarsened" indexing defined by considering each 2×2 subgrid starting at even coordinates as a single grid node. Due to the self-similarity of the Hilbert indexing the coarsened indexing is itself a Hilbert indexing.
Define $a \in \mathbb{N}$ and $b \in \{0, 1, 2, 3\}$ such that $i = 4a + b$, and $c \in \mathbb{N}$ and $d \in \{0, 1, 2, 3\}$ such that $j = 4c + d$. a and c are the positions of i and j in the coarsened indexing. Since $|a - c| \geq 16$ we can apply the induction hypothesis. Furthermore, $d(i, j) \leq 2d(a, c) + 2$ because for each of the four grid-positions in subgrid a there is a corresponding grid-position in subgrid c which is $2d(a, c)$ steps away; at worst j can be another two steps away from the grid-position corresponding to i. We now distinguish two cases regarding the relative positions of a and c.

a and c are *not* arranged as in Theorem 1:

By the induction hypothesis we have $d(a, c) \leq 3\sqrt{|a - c|} - 2.5$ and therefore

$$d(i, j) \leq 2(3\sqrt{|a - c|} - 2.5) + 2 = 6\sqrt{|a - c|} - 3 \ .$$

Substituting $a = \frac{i-b}{4}$ and $c = \frac{j-d}{4}$ we get

$$|a - c| = \frac{|(i - b) - (j - d)|}{4} \leq \frac{|i - j| + |d - b|}{4} \leq \frac{|i - j| + 3}{4}$$

and therefore

$$d(i, j) \leq 3\sqrt{|i - j| + 3} - 3 \ .$$

A simple calculation[†] shows that $3\sqrt{|i - j| + 3} \leq 3\sqrt{|i - j|} + 0.5$ for $|i - j| \geq 80$ and therefore

$$d(i, j) \leq 3\sqrt{|i - j|} - 2.5 \ .$$

[†]This is the only part of the inductive step where we need to exclude all $|i - j| < 256$. It is probably possible to modify the proof in such a way that the induction step works for $|i - j| \geq 64$. In the base case it is then sufficient to check all segments with $|i - j| \leq 64$ in a 16×16 grid.

a and c *are* arranged as in Theorem 1:

Up to symmetric cases the 2×2-subgrids for i and j are numbered $\left(\begin{smallmatrix} 0 & 3 \\ 1 & 2 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 0 & 1 \\ 3 & 2 \end{smallmatrix}\right)$ and the subgrid for j is above and to the right of the subgrid for i (refer to Figure 3). There are two subcases:

$b = d = 1$: i and j are also arranged as in Theorem 1 and therefore

$$d(i,j) = 3\sqrt{|i-j|} - 2 \ .$$

Else: We can use the estimate $d(i,j) \le 2d(a,c)+1$ because the worst case in which $d(i,j) = 2d(a,c)+2$ has already been covered by the case $b = d = 1$. A similar calculation as before now shows that

$$d(i,j) \le 2(3\sqrt{|a-c|}-2)+1 = 6\sqrt{|a-c|} - 3 \le 3\sqrt{|i-j|} - 2.5 \ .$$

□

3 Generalizing the results

There are few places in the proof of Theorem 2 where we make specific use of the properties of the Hilbert curve. We now present a generalized technique which can be applied to a wide spectrum of self-similar curves in r-dimensional grids with side-lengths $q_1^k, \ldots, q_r^k$. For simplicity, however, we restrict the presentation to cubic grids with side-length q^k and only show how slightly looser upper bounds than that of Theorem 2 can be proved. The latter constraint allows us to avoid the special treatment of the worst case segments as necessary in the proof of Theorem 2.

Theorem 3. *Given any indexing scheme of an r-dimensional grid with side-length q^k with the property that combining each q^r-cube of grid points into a single meta-point yields the numbering for side-length q^{k-1}. If*

$$\forall q^{(k-1)r} \le |i-j| \le q^{kr} : d(i,j) \le \alpha(\sqrt[r]{|i-j|} - \delta) - r$$

where

$$\delta \ge \frac{\sqrt[r]{q^{kr}+q^r-1} - q^k}{q-1}$$

then

$$\forall |i-j| \ge q^{(k-1)r} : d(i,j) \le \alpha(\sqrt[r]{|i-j|} - \delta) - r \ .$$

Proof. By induction over $|i-j|$. Let $a = \lfloor i/q^r \rfloor$, $b = i \bmod q^r$, $c = \lfloor j/q^r \rfloor$, and $d = j \bmod q^r$. Due to the self-similarity of the indexing scheme, we can apply the induction hypothesis to a and c if $|i-j| \ge q^{kr}$. We have $d(i,j) \le qd(a,c) + r(q-1)$ because for each of the q^r grid-positions in subgrid a there is a corresponding grid-position in subgrid c which is $qd(a,c)$ steps away; at worst j can be another $r(q-1)$ steps away from the grid-position corresponding to i. ($(q-1)$ along each dimension.) By the induction hypothesis we have $d(a,c) \le \alpha(\sqrt[r]{|a-c|} - \delta) - r$ and therefore

$$d(i,j) \le q(\alpha(\sqrt[r]{|a-c|} - \delta) - r) + r(q-1) = q\alpha(\sqrt[r]{|a-c|} - \delta) - r \ .$$

Substituting $a = \frac{i-b}{q^r}$ and $c = \frac{j-d}{q^r}$ we get

$$|a-c| = \frac{|(i-b)-(j-d)|}{q^r} \le \frac{|i-j|+|d-b|}{q^r} \le \frac{|i-j|+q^r-1}{q^r}$$

and therefore

$$d(i,j) \leq \alpha(\sqrt[r]{|i-j| + q^r - 1} - q\delta) - r \ \ .$$

A simple calculation shows that

$$\sqrt[r]{|i-j| + q^r - 1} - q\delta \leq \sqrt[r]{|i-j|} - \delta$$

for $|i-j| \geq q^{kr}$ and $\delta \geq \frac{\sqrt[r]{q^{kr}+q^r-1}-q^k}{q-1}$. □

Theorem 3 can be applied as follows to yield upper bounds of the form $d(i,j) \leq \alpha\sqrt[r]{|i-j|} - r$ for the Manhattan-distance of self-similar curves:

- Determine q and r from the definition of the curve.
- Fix a k.
- Try all i, j with $|i-j| \leq q^{kr}$ in a grid of side-length q^{k+1}. Due to the self-similarity of the indexing scheme we know that this suffices to find all possible Manhattan-distances for $|i-j| \leq q^k$.
- Set $\delta = \frac{\sqrt[r]{q^{kr}+q^r-1}-q^k}{q-1}$.
- Set
$$\alpha_1 = \max_{|i-j| \leq q^{kr}} \frac{d(i,j)+r}{\sqrt[r]{|i-j|}}.$$
This is sufficient to guarantee that $d(i,j) \leq \alpha_1 \sqrt[r]{|i-j|} - r$ for $|i-j| \leq q^{kr}$.
- Set
$$\alpha_2 = \max_{q^{(k-1)r} \leq |i-j| \leq q^{kr}} \frac{d(i,j)+r}{\sqrt[r]{|i-j|} - \delta}.$$
This is sufficient to guarantee that $d(i,j) \leq \alpha_2(\sqrt[r]{|i-j|} - \delta) - r$ for $q^{(k-1)r} \leq |i-j| \leq q^{kr}$. Applying Theorem 3 we can conclude that the same is true for $|i-j| \geq q^{kr}$.
- Set $\alpha = \max(\alpha_1, \alpha_2)$. This guarantees that for all i, j, $d(i,j) \leq \alpha\sqrt[r]{|i-j|} - r$.

For example, Theorem 3 yields $d(i,j) \leq 3.07\sqrt{|i-j|} - 2$ for the 2D-Hilbert curve ($q = r = 2$). if we are willing to check all $|i-j| \leq 256$ (i.e., $k = 8$) in a 32×32-grid. By checking larger grids we can get closer to the tight bound from Theorem 2.

4 A 3-D Hilbert Indexing

We have applied the result of the preceding section to the 3D-Hilbert indexing described in [12]. It is defined using the following L-System:

```
A -> B-F+CFC+F-D&F^D-F+&&CFC+F+B//
B -> A&F^CFB^F^D^^-F-D^|F^B|FC^F^A//
C -> |D^|F^B-F+C^F^A&&FA&F^C+F+B^F^D//
D -> |CFB-F+B|FA&F^A&&FB-F+B|FC//
```

`A` is the start-symbol, `ABCD` are nonterminals. The terminals `F+-&^\/|` are instructions for 3D-Turtlegraphics. They stand for "move one step," "turn left," "turn right," "pitch down," "pitch up," "roll left," "roll right," and "turn around," respectively. We applied the "recipe" from Section 3 for all k from $\{1,2,3,4,5\}$ and obtained the following values for α_1 and α_2:

k	α_1	α_2
1	4.678428	6.432089
2	4.718078	4.922481
3	4.733298	4.758845
4	4.733936	4.737128
5	4.734176	4.734575

In particular, using the highest value of k tried we get:

Theorem 4. *The Manhattan-distance between indices i and j for the 3D-Hilbert curve from [12] is bounded by*

$$d(i,j) \le 4.73458\sqrt[3]{|i-j|} - 3 \ .$$

We conjecture that $d(i,j) \le 4\sqrt[3]{63/38}\sqrt[3]{|i-j|} - 3$ is a tight bound. (Note that $4\sqrt[3]{63/38} \approx 4.73419$.) The worst case segments of the curve consist of a one large cube and two branches which form a chain of cubes for each size (so far similar to the 2-D case). One branch consists of two cubes of each size. The other branch consists of an alternating sequence of two and four cubes of each size. Summing the sizes of the subcubes in a similar way as in the proof of Theorem 1 yields the conjecture above.

5 Conclusions and Future Work

In the past, several people used indexing schemes like Hilbert curves independently and sometimes reinvented it without initially knowing the existing work. We expect that this will change. Locality-preserving indexing schemes will more and more become a standard technique for devising simple and efficient algorithms for mesh-connected computers. So far, most applications look somewhat specialized [3, 8, 9, 10, 16, 17]. With an increasing interest in irregular computations this may also change. For example, we have developed a variant of parallel Quick-sort which is based on the Hilbert indexing. This has been implemented by Thomas Hansch in [5, 6]. On large mesh-connected computers like the Parsytec GCel-3/1024 which is a 32×32 grid, the algorithms turns out to be one of the fastest available sorting methods.

We hope that this paper helps to simplify the reasoning about theoretical properties of recursive indexing schemes. The methods used here not only give a simple and easy-to-verify proof for the 2D-Hilbert curve but also show how the result can be generalized. To our best knowledge, it is the first to report an (almost) tight bound for a 3D-Hilbert curve. In contrast to the 2D-case, there are several variants of the 3D-curve. It might be interesting to investigate whether there are better variants. There are also well known other curves which can be investigated using our techniques. For example, the Peano curve [11, 13] shown in the upper part of Figure 5 and its relative shown in in the lower part, can be treated setting $r = 2$ and $q = 3$.

When high-dimensional or otherwise complex curves are to be investigated, the simple brute force inspection programs we have used above get quite slow. They check all pairs of indices up to a certain distance. (More precisely, the computational expense is in $\Theta\left(q^{2kr}\right)$). Checking the 3D-Hilbert indexing for $k = 5$ in Section 4 took several CPU hours on a 110 MHz SPARC 4. By tolerating some more complex programming, this situation can be improved. We can use a hierarchical approach in which conservative estimates for the distance of all pairs of indices in two large cubes are computed. Only when these estimates suggest that high diameter segments might be present, the cubes have to be subdivided further. Since most index pairs are far from extreme, a dramatic speedup could be achieved.

Acknowledgment We are grateful to Henning Fernau for valuable pointers to the literature.

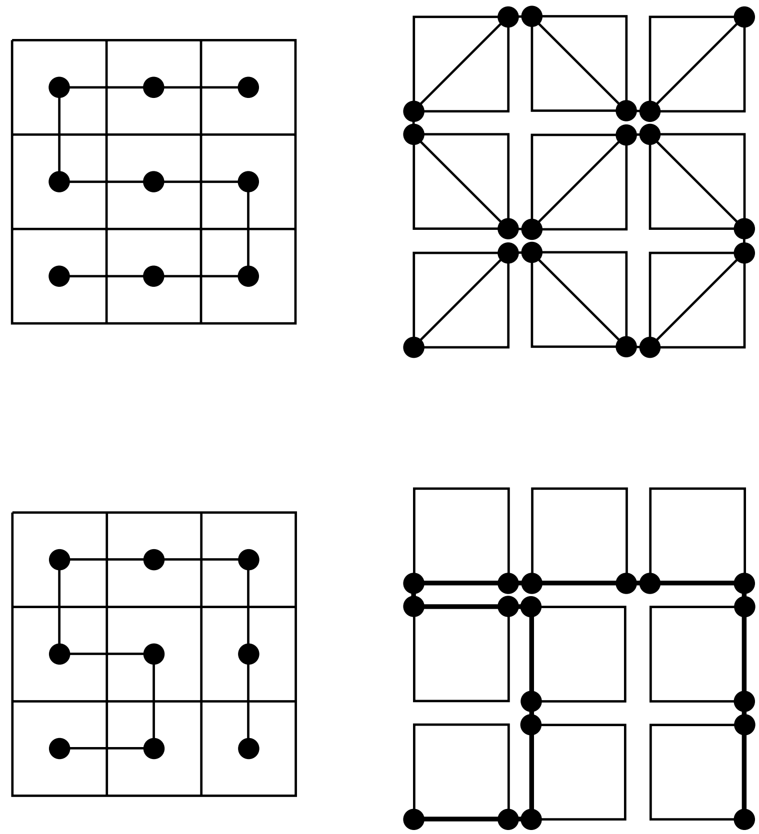

Figure 5: Construction principle for space filling curves with $r = 2$, $q = 3$.

References

[1] G. Bilardi and F. P. Preparata. Horizons of parallel computation. *Journal of Parallel and Distributed Computing*, 27:172–182, 1995.

[2] G. Chochia, M. Cole, and T. Heywood. Implementing the Hierarchical PRAM on the 2D mesh: Analyses and experiments. In *IEEE Symp. on Parallel and Distributed Processing*, 1995.

[3] A. Dingle and I. H. Sudborough. Efficient mappings of pyramid networks. *IEEE Transactions on Parallel and Distributed Systems*, 5(10):1009–1017, 1994.

[4] Y. Feldman and E. Shapiro. Spatial machines: A more realistic approach to parallel computation. *Communications of the ACM*, 35(10):61–73, October 1992.

[5] T. Hansch. Segmentierte kollektive Operationen auf Gittern mit Hilbertnumerierung. Studienarbeit, 1995.

[6] T. Hansch. Sortieren großer Datenmengen auf Gittern mit Quicksort. Diplomarbeit, 1996.

[7] D. Hilbert. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Mathematische Annalen*, 38:459–460, 1891.

[8] M. Kaddoura, C. Ou, and S. Ranke. Mapping unstructured computational graphs for adaptive and nonuniform computational environments. Technical report, Syracuse University, 1995. submitted to IEEE Parallel and Distributed Technology.

[9] C. Kaklamanis and G. Persiano. Branch-and-bound and backtrack search on mesh-connected arrays of processors. *Mathematical Systems Theory*, 27:471–489, 1994.

[10] R. Miller and Q. F. Stout. Mesh computer algorithms for computational geometry. *IEEE Transactions on Computers*, 38(3):321–340, March 1989.

[11] G. Peano. Sur une courbe qui remplit toute une aire plane. *Mathematische Annalen*, 36:157–160, 1890.

[12] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer, 1991.

[13] H. Sagan. *Space-Filling Curves*. Universitext. Springer-Verlag, 1994.

[14] A. Schorr. Physical parallel devices are not much faster than sequential ones. *Information Processing Letters*, 17:103–106, 1983.

[15] J. F. Sibeyn. Overview of mesh results. Technical report MPI-I-95-1-018, Max-Planck-Institut für Informatik, Saarbrücken, 1995.

[16] J. P. Singh. *Parallel Hierarchical N-Body Methods and their Implications for Multiprocessors*. PhD thesis, Stanford University, 1993.

[17] Q. F. Stout. Topological matching. In *ACM Symposium on the Theory of Computing*, pages 24–31, 1983.

[18] P. Vitányi. Locality, communication, and interconnect length in mul ticomputers. *SIAM J. Comput.*, 17(4):659–672, August 1988.