(t, m, s)-Nets and Maximized Minimum Distance

Leonhard Grünschloß[1], Johannes Hanika[2], Ronnie Schwede[3], and Alexander Keller[4]

[1] Ulm University, Germany
leonhard.gruenschloss@googlemail.com
[2] Ulm University, Germany
johannes.hanika@uni-ulm.de
[3] Eawag, Swiss Federal Institute of Aquatic Science and Technology, Switzerland
ronnie.schwede@eawag.ch
[4] Ulm University, Germany
alexander.keller@uni-ulm.de

Summary. Many experiments in computer graphics imply that the average quality of quasi-Monte Carlo integro-approximation is improved as the minimal distance of the point set grows. While the definition of (t, m, s)-nets in base b guarantees extensive stratification properties, which are best for $t = 0$, sampling points can still lie arbitrarily close together. We remove this degree of freedom, report results of two computer searches for $(0, m, 2)$-nets in base 2 with maximized minimum distance, and present an inferred construction for general m. The findings are especially useful in computer graphics and, unexpectedly, some $(0, m, 2)$-nets with the best minimum distance properties cannot be generated in the classical way using generator matrices.

1 Introduction

Image synthesis can be considered as an integro-approximation problem [Kel06]

$$g(y) = \int_{[0,1)^s} f(x, y)dx \approx \frac{1}{n}\sum_{i=0}^{n-1} f(x_i, y), \tag{1}$$

where g is the image function with pixel coordinates y on the screen. One numerical method to compute approximations is the method of dependent tests, where one set $P_n := \{x_0, \ldots, x_{n-1}\}$ of points from the unit cube $[0, 1)^s$ is used to simultaneously estimate all averages $g(y)$. The accumulation buffer [HA90] for realistic image synthesis in computer graphics is a very popular realization of that scheme. In this application the image plane is tiled by one sampling point set P_n as illustrated in Figure 2.

2 (t, m, s)-Nets in Base 2

For the scope of this paper, only (t, m, s)-nets (for an extensive reference see [Nie92, Ch. 4]) in base 2 up to $s = 3$ are considered. While this may seem like a strong restriction, improving these patterns directly results in considerable performance gains in industrial rendering applications [HA90, CCC87, Kel03]. The considerations in base 2 go back to Sobol's LP_τ-nets and -sequences [Sob67]. This basic concept was generalized by Niederreiter [Nie92], which for base 2 is given by

Definition 1. *For two integers $0 \leq t \leq m$, a finite point set of 2^m s-dimensional points is a (t, m, s)-net in base 2, if every elementary interval of size $\lambda_s(E) = 2^{t-m}$ contains exactly 2^t points.*

The elementary intervals in base 2 are specified in

Definition 2. *For $l_j \in \mathbb{N}_0$ and integers $0 \leq a_j < 2^{l_j}$ the elementary interval is*

$$E := \prod_{j=1}^{s} \left[\frac{a_j}{2^{l_j}}, \frac{a_j + 1}{2^{l_j}} \right) \subseteq [0, 1)^s.$$

The parameter t controls the stratification properties of the net, which are best for $t = 0$, because then every elementary interval contains exactly one point (see Figure 1). Thus the pattern is both a Latin hypercube sample and stratified in the sense of [CPC84]. For base 2, $(0, m, s)$-nets can only exist up to dimension $s = 3$ [Nie92, Cor. 4.21].

2.1 Matrix-Generated (t, m, s)-Nets

The classical way to generate (t, m, s)-nets is the use of generator matrices. In the following we review the efficient generation of these nets and a method for checking whether $t = 0$.

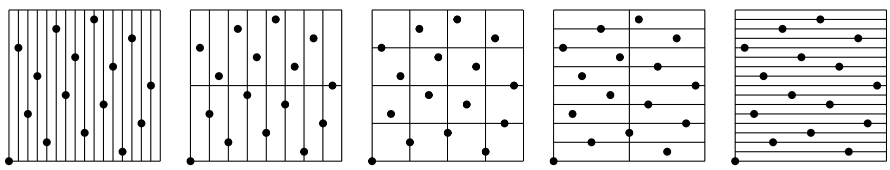


Fig. 1. The quality parameter $t = 0$ ensures extensive stratification: There is exactly one point inside each elementary interval of volume 2^{-4} of this $(0, 4, 2)$-net in base 2. There are exactly $4 + 1$ partitions into 2^4 elementary intervals. The leftmost and rightmost kind of elementary intervals constitute the Latin hypercube property.

Definition 3. *An s-dimensional point set $P_n := \{x_0, \ldots, x_{n-1}\}$ with $n = 2^m$ and $x_i := (x_i^{(1)}, \ldots, x_i^{(s)}) \in [0,1)^s$ is called $C_1, \ldots, C_s$-generated point set in base 2 and dimension s, if $C_1, \ldots, C_s \in \mathbb{F}_2^{m \times m}$ with*

$$x_i^{(j)} = \left(\frac{1}{2} \cdots \frac{1}{2^m}\right) \left[C_j \begin{pmatrix} d_0(i) \\ \vdots \\ d_{m-1}(i) \end{pmatrix} \right], \text{ where } i = \sum_{k=0}^{m-1} d_k(i) 2^k$$

and the matrix-vector product in brackets is performed in $\mathbb{F}_2$. The matrices C_j for $j = 1, \ldots, s$ are called the generators of the point set P_n, more precisely, the matrix C_j is the generator of the j-th coordinate of the point set.

Computations in base 2 allow for exact representation of the coordinates in the IEEE floating point standard, as long as the mantissa holds enough bits (23 in the case of the 4-byte `float`). Interpreting unsigned integers as bit vectors, i.e. elements of $\mathbb{F}_2^{32}$ with $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$, allows one to use standard bitwise operations for vector operations on $\mathbb{F}_2^{32}$. The matrix-vector multiplication $C_j\,(d_0(i), \ldots, d_{m-1}(i))$, which has to be performed in $\mathbb{F}_2$, then becomes very efficient. Exploiting this simple kind of parallelism results in the following algorithm in `C++`, which uses at most $\mathcal{O}(m)$ operations to compute $x_i^{(j)}$. It relies on the fact that addition corresponds to XOR in $\mathbb{F}_2$.

```
double x_j(unsigned int i)
{
  unsigned int result = 0;

  for (unsigned int k = 0; i; i >>= 1, k++)
    if(i & 1)
      // vector addition of (k+1)-th leftmost column of C_j
      result ^= C_j[k];

  return (double) result / (double) (1ULL << m);
}
```

Whether or not a given set of generator matrices produces a net with $t = 0$, can be checked using the following

Theorem 1 (see [Nie92, Thm. 4.28] and [LP01, Def. 1]). *Let P_n be a $C_1, \ldots, C_s$-generated $n = 2^m$-point set in base 2 and dimension s, then P_n is a $(0, m, s)$-net in base 2 if for all $\mathbf{d} = (d_1, \ldots, d_s)^T \in \mathbb{N}_0^s$ with $|\mathbf{d}| = m$ the following holds:*

$$\det\left(C^{\left(\sum_{j=1}^{s} d_j\right)}\right) \neq 0,$$

where

$$C^{\left(\sum_{j=1}^{s} d_j\right)} = \begin{pmatrix} c^1_{1,1} \dots c^1_{1,m} \\ \vdots \\ c^1_{d_1,1} \dots c^1_{d_1,m} \\ \vdots \\ c^s_{1,1} \dots c^s_{1,m} \\ \vdots \\ c^s_{d_s,1} \dots c^s_{d_s,m} \end{pmatrix} \in \mathbb{F}_2^{m \times m},$$

with $C_j = (c^j_{k,l})^m_{k,l=1}$ *for* $j = 1, \dots, s$*. This means* $C^{\left(\sum_{j=1}^{s} d_j\right)}$ *is an* $m \times m$ *matrix consisting of the first* d_j *rows of the generator* C_j *for all* $j = 1, \dots, s$*.*

As a consequence of Theorem 1, all generator matrices $C_1, \dots, C_s$ must be regular. Remark (iv) in [LP01, p. 3] states that the $C_1, \dots, C_s$-generated $(0, m, s)$-net in base 2 and the net generated by $C_1 D, \dots, C_s D$, for any $D \in \mathrm{GL}_m(\mathbb{F}_2)$ contain exactly the same points, where $\mathrm{GL}_m(\mathbb{F}_2)$ denotes the general linear group of matrices over $\mathbb{F}_2$ of dimension $m \times m$. This means that the above mentioned nets are identical except for the numbering of their points. Using this remark we can fix one matrix of a matrix-generated $(0, m, s)$-net in base 2 by choosing

$$C_1 := \begin{pmatrix} 0 & & 1 \\ & \cdot^{\cdot^{\cdot}} & \\ 1 & & 0 \end{pmatrix} \tag{2}$$

which results in the first coordinate $x^{(1)}_i = \frac{i}{2^m}$.

For the special case, where $s = 2$ and C_1 as defined in (2), Theorem 1 then results in the following test for $t = 0$:

$$\det \underbrace{\begin{pmatrix} c^2_{1,1} \dots c^2_{1,k} \\ \vdots \\ c^2_{k,1} \dots c^2_{k,k} \end{pmatrix}}_{=:S_k} \neq 0 \ \forall k \in \{1, \dots, m\} \Rightarrow C_1, C_2 \text{ generate a } (0, m, 2)\text{-net}, \tag{3}$$

because all possible vectors $\mathbf{d}$ as stated in the theorem are considered.

2.2 Permutation-Generated (t, m, s)-Nets

Definition 1 states that a (t, m, s)-net in base 2 is a Latin hypercube sample. This means that for all (t, m, s)-nets in base 2, the integer parts of the coordinates multiplied by 2^m must be a permutation of the numbers given by the integer parts of the first coordinates $x^{(1)}_i$ multiplied by 2^m.

The $(0, m, 2)$-nets in base 2 given by $x_i = \frac{1}{2^m}(i, \sigma(i))$, where σ is a permutation of the numbers $\{0, \dots, 2^m - 1\}$, can be enumerated by using a modified version of the classic backtracking algorithm to solve the *n-Rooks* problem [Rol05] equipped with an additional test. This test checks whether the points fulfill the stratification conditions imposed by the structure of the elementary intervals (see Figure 1). While this seems to be a complicated test, it can in fact be realized with only a few lines of code in C, if the base is 2:

```
for (k = 1; k < m; k++)
{
  // combine k bits of i and m-k bits of j to form index
  idx = (i >> (m - k)) + (j & (0xFFFFFFFF << k));

  if(elementaryInterval[k][idx]++) // already one point there?
    break;  // t > 0 !
}
```

The code fragment tests whether a point given by the coordinates i and $j = \sigma(i)$ falls into an elementary interval that is already taken. In that case the points cannot be a $(0, m, 2)$-net in base 2. There are exactly $(m + 1)$ kinds of each 2^m elementary intervals (see Figure 1). The first and last kind of elementary intervals constitute the Latin hypercube property, which does not need to be checked, because the points are determined by a permutation. Therefore the variable `k` iterates from 1 to $m - 1$. The array `elementaryInterval` counts how many points are in the k-th kind of elementary interval addressed by the index `idx`, which is efficiently computed by using k bits of the first coordinate i and $m - k$ bits of the second coordinate $j = \sigma(i)$.

3 Low Discrepancy and Minimum Distance

Although $(0, m, s)$-nets have exhaustive stratification properties (see Defintion 1) that guarantee low discrepancy [Nie92], these properties do not avoid that points can lie arbitrarily close together across boundaries of elementary intervals.

Maximizing the minimum distance is a basic concept in nature [Yel83] and has been applied extensively in computer graphics [HA90, Gla95]. Minimum distance has been proposed as a measure for uniformity in [Pan04, Kel06], too. In addition it has been observed that scrambling [Owe95] can change minimum distance [Kel04] and that in fact points with maximized minimum distance perform better on the average. We therefore perform an exhaustive computer search for $(0, m, 2)$-nets in base 2 with maximized minimum distance.

The minimum distance

$$d_{\min}(P_n) := \min_{0 \le i < j < n} \|x_i - x_j\|$$

of a point set $P_n = \{x_0, \ldots, x_{n-1}\}$ is the smallest distance between any two distinct points of this set. However, the choice of the norm $\|\cdot\|$ is important.

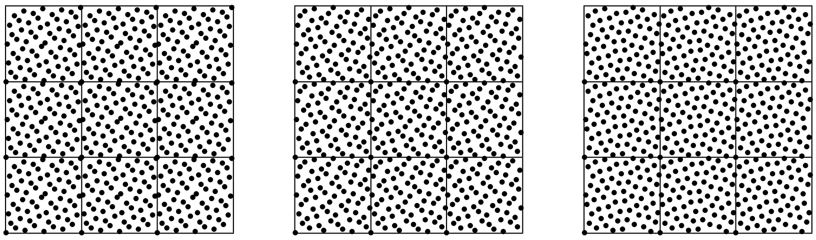


Fig. 2. Three examples of 3×3 periodically tiled $(0, 6, 2)$-nets in base 2.

Choosing the Euclidean distance $\|x - y\| = \sqrt{\sum_{i=1}^{s} |x_i - y_i|^2}$ is not sufficient for our purposes, as for many graphics applications it is beneficial to be able to tile the same net periodically as can be seen in Figure 2 [KH01]. As we do not want the minimum distance to decrease for such a point set consisting of shifted copies of P_n we use the toroidal distance:

Definition 4. *For two points* $x = (x_1, \dots, x_s) \in [0, 1)^s$ *and* $y = (y_1, \dots, y_s) \in [0, 1)^s$ *their toroidal distance is defined as*

$$\|x - y\|_T := \sqrt{\sum_{i=1}^{s} \left(\min\{|x_i - y_i|, 1 - |x_i - y_i|\}\right)^2}.$$

Randomization

Randomized quasi-Monte Carlo point sets can be used to reduce the variance of Monte Carlo estimators and allow for an unbiased error estimate on the class of square-integrable functions [Owe98]. While random scrambling [Owe95, FK02, KK02] does not alter the parameter t, it can decrease the minimum distance of a (t, m, s)-net dramatically [Kel04].

On the contrary a random shift on the unit torus, i.e. a Cranley-Patterson rotation [CP76], preserves minimum distance even for periodically tiled nets, which perfectly matches our optimization goal. While it is often argued that the parameter t can be affected by shifting, it also can be argued that just the integrand is shifted leaving the point set structure untouched.

3.1 Exhaustive Matrix Computer Search

Following Section 2.1, we consider matrix-generated $(0, m, 2)$-nets. As the search space of C_2 generator matrices is exponentially growing in m and the limiting factor for the performance is the minimum distance evaluation, we only want to do these calculations for matrices which fulfill the $t = 0$ property. To efficiently enumerate such matrices, we exploit the conclusions of

Table 1. Minimum distance $d_{\min}$ of $(0, m, 2)$-nets in base 2 for Hammersley, Larcher-Pillichshammer and the new construction.

m	Hammersley	Larcher-Pillichshammer	new construction
2	$\sqrt{2}/2^2 \approx 0.35355339$	$\sqrt{2}/2^2 \approx 0.35355339$	$\sqrt{2}/2^2 \approx 0.35355339$
3	$\sqrt{2}/2^3 \approx 0.17677670$	$\sqrt{5}/2^3 \approx 0.27950850$	$\sqrt{8}/2^3 \approx 0.35355339$
4	$\sqrt{2}/2^4 \approx 0.08838835$	$\sqrt{8}/2^4 \approx 0.17677670$	$\sqrt{13}/2^4 \approx 0.22534695$
5	$\sqrt{2}/2^5 \approx 0.04419417$	$\sqrt{18}/2^5 \approx 0.13258252$	$\sqrt{29}/2^5 \approx 0.16828640$
6	$\sqrt{2}/2^6 \approx 0.02209709$	$\sqrt{32}/2^6 \approx 0.08838835$	$\sqrt{52}/2^6 \approx 0.11267348$
7	$\sqrt{2}/2^7 \approx 0.01104854$	$\sqrt{72}/2^7 \approx 0.06629126$	$\sqrt{100}/2^7 \approx 0.07812500$
8	$\sqrt{2}/2^8 \approx 0.00552427$	$\sqrt{128}/2^8 \approx 0.04419417$	$\sqrt{208}/2^8 \approx 0.05633674$
9	$\sqrt{2}/2^9 \approx 0.00276214$	$\sqrt{265}/2^9 \approx 0.03179457$	$\sqrt{400}/2^9 \approx 0.03906250$
10	$\sqrt{2}/2^{10} \approx 0.00138107$	$\sqrt{512}/2^{10} \approx 0.02209709$	$\sqrt{832}/2^{10} \approx 0.02816837$
11	$\sqrt{2}/2^{11} \approx 0.00069053$	$\sqrt{1060}/2^{11} \approx 0.01589729$	$\sqrt{1600}/2^{11} \approx 0.01953125$
12	$\sqrt{2}/2^{12} \approx 0.00034527$	$\sqrt{2048}/2^{12} \approx 0.01104854$	$\sqrt{3328}/2^{12} \approx 0.01408418$
13	$\sqrt{2}/2^{13} \approx 0.00017263$	$\sqrt{4153}/2^{13} \approx 0.00786667$	$\sqrt{6385}/2^{13} \approx 0.00975417$
14	$\sqrt{2}/2^{14} \approx 0.00008632$	$\sqrt{8192}/2^{14} \approx 0.00552427$	$\sqrt{13312}/2^{14} \approx 0.00704209$
15	$\sqrt{2}/2^{15} \approx 0.00004316$	$\sqrt{16612}/2^{15} \approx 0.00393334$	$\sqrt{25313}/2^{15} \approx 0.00485536$
16	$\sqrt{2}/2^{16} \approx 0.00002158$	$\sqrt{32768}/2^{16} \approx 0.00276214$	$\sqrt{53248}/2^{16} \approx 0.00352105$

Table 2. Maximum obtainable minimum distance $d_{\min}$ for matrix-generated $(t, m, 2)$-nets with $t \geq 0$. Requiring $t = 0$ does not always allow for obtaining the maximum.

t	0	0	0, 1	0, 1	1, 2, 3
m	2	3	4	5	6
$d_{\min}$	$\sqrt{2}/2^2 \approx$ 0.35355339	$\sqrt{8}/2^3 \approx$ 0.35355339	$\sqrt{13}/2^4 \approx$ 0.22534695	$\sqrt{29}/2^5 \approx$ 0.16828640	$\sqrt{65}/2^6 \approx$ 0.12597278

Theorem 1 that are sufficient to ensure only one point per elementary interval. The matrices are enumerated using a backtracking algorithm that first checks, whether $\det(S_k) \neq 0$ (see Equation (3)) and in that case tries to extend the matrix S_k by a right column and a bottom row to form S_{k+1} (see Figure 3). If such an extension cannot be found with $\det(S_{k+1}) \neq 0$, the next S_k will be explored according to the backtracking search principle.

To compute the determinant, we apply the standard Gauss elimination scheme. The implementation exploits that permuting matrix rows does not change the determinant in $\mathbb{F}_2$. It exits on the first resulting zero on the diagonal

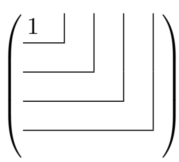


Fig. 3. Iterative construction of C_2 for a $(0, m, 2)$-net by sequences of S_k.

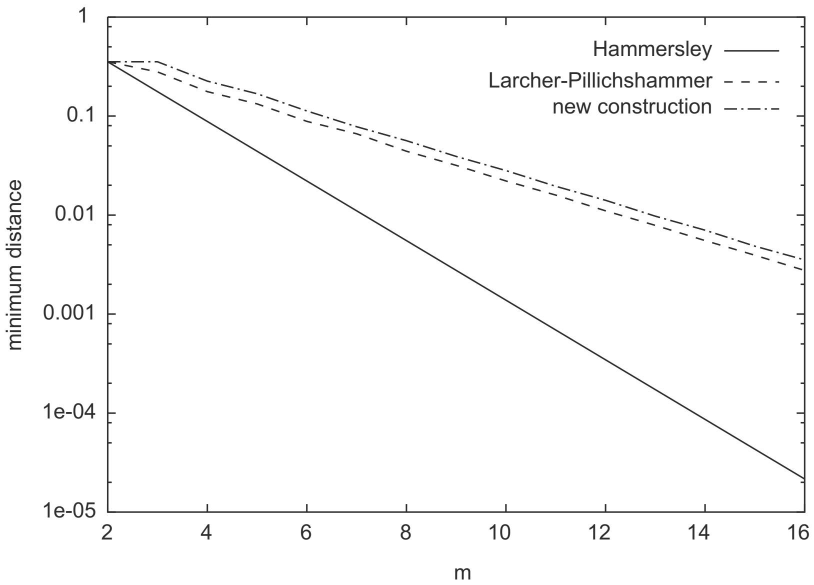
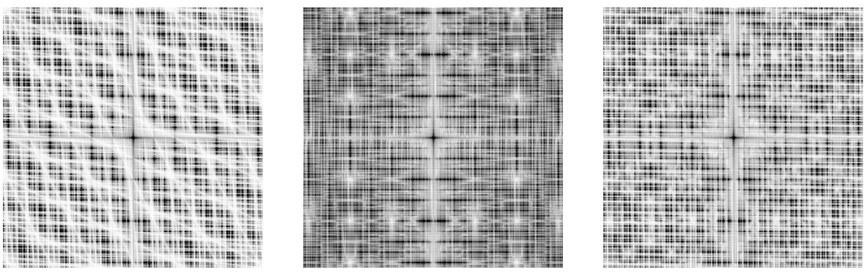


Fig. 4. Minimum distance $d_{\min}$ of a Hammersley-net, a Larcher-Pillichshammer-net and the new construction.



Hammersley Larcher-Pillichshammer new construction

Fig. 5. Squared amplitudes of the Fourier transformation of the Hammersley-, Larcher-Pillichshammer- and the newly constructed $(0, 10, 2)$-net in base 2 (inverted, higher values are darker). A larger region of low frequencies gets attenuated as the minimum distance increases (similar to the blue-noise spectrum [Yel83]). This region is largest for the new construction and means that low frequencies are reproduced more precisely. Like the Larcher-Pillichshammer-net, the new construction is much more isotropic as compared to the Hammersley-net.

signaling a zero determinant. Using 32-bit integers together with the bitwise XOR operation to perform vector addition in $\mathbb{F}_2^{32}$, the algorithm runs in $\mathcal{O}(k^2)$. To further improve performance, we found it beneficial to use vector operations of modern computers (i.e. the SSE2 instruction set) for some values of m. A non-zero determinant implies $t = 0$.

Next, to calculate the minimum distance all points are enumerated according to the gray code numbering [PTVF92]. Computing square distances multiplied by 2^m allows for using integer arithmetic and guarantees to avoid any floating point arithmetic problem. The omission of the division by 2^m in the first code fragment (see Section 2.1) then results in integer coordinates.

For the efficient minimum distance computation the resulting points are sorted into a regular grid. This can be done very efficiently if m is even, because then one kind of elementary intervals must be squares forming a regular grid (see the middle plot in Figure 1). As mentioned before exactly one point will fall into each grid cell. Assigning a grid cell to a point is done by simply omitting the least $\lceil \frac{m}{2} \rceil$ significant bits of the point coordinates. To find the minimum distance of all points to one fixed point it is then sufficient to examine the eight points in the neighboring cells. If m is odd, we use a square regular grid with cells twice the area of the elementary intervals. So in this case there are two points per cell which gives the complexity of the algorithm a worse constant than in the even case. It still runs in $\mathcal{O}(n)$ where $n = 2^m$ is the number of points.

Search Results

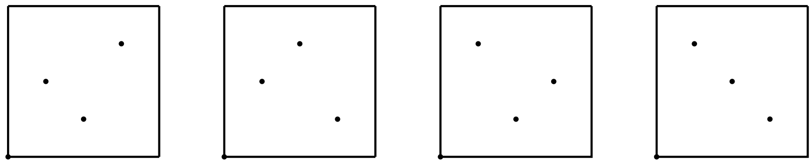
For $m \leq 6$ Figures 6–10 show the $(0, m, 2)$-nets in base 2 along with the generator matrix C_2 for the second coordinate, where the minimum distance between the points on the torus is maximal. Note that C_1 always is the flipped unit matrix as defined in Equation (2). Figure 11 considers the case $m = 7$, where only a fraction of the search space could be explored and thus there could exist generator matrices resulting in a larger minimum distance.

By abandoning the $t = 0$ constraint, the stratification properties of $(0, m, s)$-nets are lost, but the minimum distance of resulting nets can be increased even further (see Table 2). Instead of testing for $t = 0$, now the quality parameter t is computed following the algorithm outlined in [PS01] after determining the minimum distance of the point set. The size $\mathcal{O}\left(2^{m^2}\right)$ of the search space is even larger, but a complete search is still feasible up to $m \leq 6$.

3.2 Restricted Computer Search

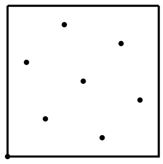
Our exhaustive matrix search is only computationally feasible up to $m \leq 6$. However, the search results allow one to infer a submatrix structure:

$$\text{for even } m: \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \rightsquigarrow \left(\begin{array}{c|cc|c} 1 & 0 & 1 & 1 \\ \cline{2-3} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ \cline{2-3} 0 & 0 & 0 & 1 \end{array}\right) \rightsquigarrow \left(\begin{array}{c|cccc|c} 1 & 0 & 0 & 0 & 0 & 0 \\ \cline{2-5} 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ \cline{2-5} 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \rightsquigarrow \dots, \qquad (4)$$



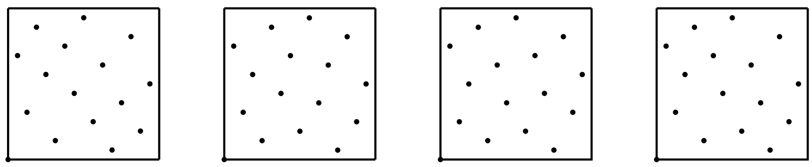
$$\begin{pmatrix}1&0\\0&1\end{pmatrix} \quad \begin{pmatrix}1&1\\0&1\end{pmatrix} \quad \begin{pmatrix}1&0\\1&1\end{pmatrix} \quad \begin{pmatrix}1&1\\1&0\end{pmatrix}$$

Fig. 6. All matrix-generated $(0, 2, 2)$-nets in base 2 with maximal minimum distance $d_{\min} = \sqrt{2}/2^2 \approx 0.35355339$.



$$\begin{pmatrix}1&0&1\\0&1&0\\1&0&0\end{pmatrix}$$

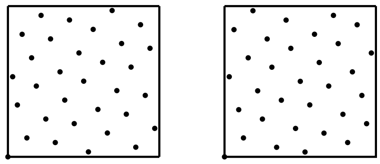
Fig. 7. All matrix-generated $(0, 3, 2)$-nets in base 2 with maximal minimum distance $d_{\min} = \sqrt{8}/2^3 \approx 0.35355339$.



$$\begin{pmatrix}1&0&1&1\\0&1&0&1\\1&0&0&1\\1&1&1&1\end{pmatrix} \quad \begin{pmatrix}1&0&0&0\\1&1&0&0\\0&0&1&1\\0&1&0&1\end{pmatrix} \quad \begin{pmatrix}1&0&1&1\\1&1&1&1\\0&0&1&1\\0&0&0&1\end{pmatrix} \quad \begin{pmatrix}1&0&1&1\\1&1&1&1\\0&0&1&1\\0&1&0&1\end{pmatrix}$$

Fig. 8. All matrix-generated $(0, 4, 2)$-nets in base 2 with maximal minimum distance $d_{\min} = \sqrt{13}/2^4 \approx 0.22534695$.

$$\text{for odd } m : \begin{pmatrix}1&0&1\\0&1&0\\1&0&0\end{pmatrix} \rightsquigarrow \left(\begin{array}{c|ccc|c}1&0&0&0&1\\ \hline 0&1&0&1&0\\0&0&1&0&0\\0&1&0&0&0\\ \hline 1&1&0&0&0\end{array}\right) \rightsquigarrow \dots \,. \tag{5}$$



$$\begin{pmatrix} 1&0&0&0&1 \\ 0&1&0&1&0 \\ 0&0&1&0&0 \\ 0&1&0&0&0 \\ 1&1&0&0&0 \end{pmatrix} \qquad \begin{pmatrix} 1&1&0&0&1 \\ 0&1&0&1&0 \\ 0&0&1&0&0 \\ 0&1&0&0&0 \\ 1&1&0&0&0 \end{pmatrix}$$

Fig. 9. All matrix-generated $(0, 5, 2)$-nets in base 2 with maximal minimum distance $d_{\min} = \sqrt{29}/2^5 \approx 0.16828640$.

This observation motivated our approach to restrict the search to reduce the growth of the search space to order $\mathcal{O}(2^m)$. For this purpose, we define the following matrix structure.

We write a matrix $C \in \mathbb{F}_2^{m \times m}$ with $m \geq 4$ as matrix $\mathcal{I} = (i_{k,l})_{k,l=1}^{m-2}$ and vectors $\mathbf{u} = (u_1 \cdots u_m)$, $\mathbf{r} = (r_1, \ldots, r_{m-2}), \mathbf{b} = (b_1 \cdots b_m)$ and $\mathbf{l} = (l_1, \ldots, l_{m-2})$ in the following way:

$$C = \left(\begin{array}{c|cccc|c} u_1 & u_2 & \ldots & & u_{m-1} & u_m \\ \hline l_1 & i_{1,1} & \ldots & & i_{1,m-2} & r_1 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ l_{m-2} & i_{m-2,1} & \ldots & & i_{m-2,m-2} & r_{m-2} \\ \hline b_1 & b_2 & \ldots & & b_{m-1} & b_m \end{array} \right) = \begin{bmatrix} & \mathbf{u} & \\ \mathbf{l} & \mathcal{I} & \mathbf{r} \\ & \mathbf{b} & \end{bmatrix}.$$

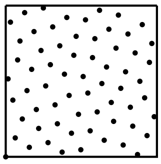
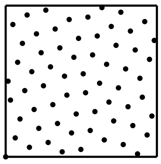
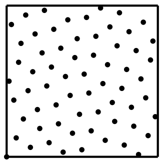
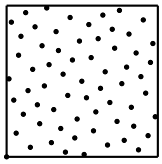
This way we are able to continue the matrix expansion scheme (4) and (5) as follows:

$$C_2^{(m)} = \begin{bmatrix} & \mathbf{u} & \\ \mathbf{l} & C_2^{(m-2)} & \mathbf{r} \\ & \mathbf{b} & \end{bmatrix}.$$

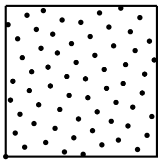
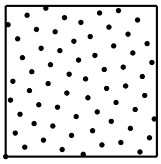
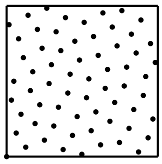
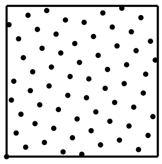
Search Results

Our iterative search algorithm takes an $(m-2) \times (m-2)$-matrix and seeks for the best $\mathbf{u}, \mathbf{r}, \mathbf{b}, \mathbf{l}$ vectors by maximizing the minimum distance of the resulting $(0, m, 2)$-net. The resulting generator matrices for $m = 7, 8, 9$ are given in Figure 12.

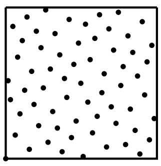
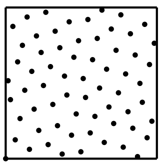
However, this iterative approach does not yield generator matrices that obtain the maximal possible minimum distance, which already becomes apparent for $m = 7$. The largest minimum distance of the generator matrix that was found by the iterative search is $\sqrt{98}/2^7$ whereas the incomplete (see previous section) full matrix search already revealed a matrix with minimum distance $\sqrt{100}/2^7$ (see Figure 11).

$$\begin{pmatrix}1&1&0&1&1&0\\0&1&0&1&1&0\\0&1&1&1&1&0\\0&0&0&1&1&0\\0&0&0&0&1&1\\1&1&0&1&1&1\end{pmatrix} \quad \begin{pmatrix}1&0&0&0&0&0\\0&1&0&1&1&0\\0&1&1&1&1&0\\0&0&0&1&1&0\\0&0&0&0&1&1\\0&0&0&0&0&1\end{pmatrix} \quad \begin{pmatrix}1&1&0&1&1&0\\0&1&0&1&1&0\\0&1&1&1&1&0\\0&0&0&1&1&0\\0&0&0&0&1&1\\0&0&0&0&0&1\end{pmatrix} \quad \begin{pmatrix}1&1&0&1&0&1\\0&1&0&1&1&0\\0&1&1&1&1&0\\0&0&0&1&1&0\\0&0&1&0&1&0\\1&1&0&1&0&0\end{pmatrix}$$

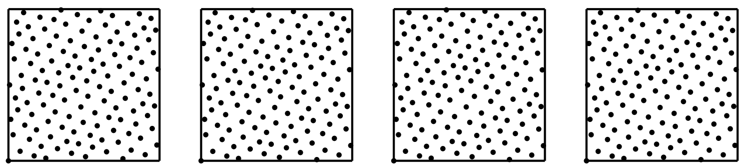





$$\begin{pmatrix}1&0&0&0&1&1\\1&1&0&0&1&1\\1&1&1&0&1&1\\0&0&0&1&1&0\\0&0&1&0&1&0\\0&0&0&0&0&1\end{pmatrix} \quad \begin{pmatrix}1&0&0&0&0&1\\1&1&0&0&0&1\\1&1&1&0&0&1\\0&0&0&1&1&0\\0&0&1&0&1&0\\0&0&0&0&1&1\end{pmatrix} \quad \begin{pmatrix}1&0&0&0&0&0\\1&1&0&0&0&0\\1&1&1&0&0&0\\0&0&0&1&1&0\\0&0&1&0&1&0\\0&0&0&0&1&1\end{pmatrix} \quad \underbrace{\begin{pmatrix}1&0&0&0&0&0\\1&1&0&0&0&0\\1&1&1&0&0&0\\0&0&0&1&1&0\\0&0&1&0&1&0\\0&0&0&0&0&1\end{pmatrix}}_{C_2^{(6)}}$$

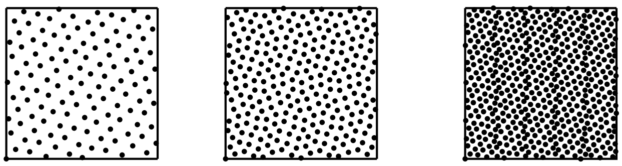
$$\begin{pmatrix}1&0&0&0&0&1\\0&1&0&1&1&0\\0&1&1&1&1&0\\0&0&0&1&1&0\\0&0&1&0&1&0\\1&1&0&1&0&0\end{pmatrix} \quad \begin{pmatrix}1&0&0&0&0&0\\0&1&0&1&1&0\\0&1&1&1&1&0\\0&0&0&1&1&0\\0&0&0&0&1&1\\1&1&0&1&1&1\end{pmatrix}$$

Fig. 10. All matrix-generated $(0, 6, 2)$-nets in base 2 with maximal minimum distance $d_{\min} = \sqrt{52}/2^6 \approx 0.11267348$.



$$\begin{pmatrix}1&0&0&0&0&0&0\\0&1&0&1&1&0&1\\0&0&1&0&1&0&0\\0&0&0&1&0&0&0\\0&0&1&0&0&0&0\\0&1&1&1&0&1&0\\0&1&0&1&1&0&0\end{pmatrix}\quad\begin{pmatrix}1&1&0&1&1&0&0\\0&1&0&1&1&0&1\\0&0&1&0&1&0&0\\0&0&0&1&0&0&0\\0&0&1&0&0&0&0\\0&1&1&1&0&1&0\\0&1&0&1&1&0&0\end{pmatrix}\quad\begin{pmatrix}1&0&0&0&0&0&1\\0&1&0&1&1&0&1\\0&0&1&0&1&0&0\\0&0&0&1&0&0&0\\0&0&1&0&0&0&0\\0&1&1&1&0&1&0\\0&1&0&1&1&0&0\end{pmatrix}\quad\underbrace{\begin{pmatrix}1&1&0&1&1&0&1\\0&1&0&1&1&0&1\\0&0&1&0&1&0&0\\0&0&0&1&0&0&0\\0&0&1&0&0&0&0\\0&1&1&1&0&1&0\\0&1&0&1&1&0&0\end{pmatrix}}_{C_2^{(7)}}$$

Fig. 11. $(0, 7, 2)$-nets in base 2 with minimum distance $d_{\min} = \sqrt{100}/2^7 = 0.078125$ (possibly not the maximum).



$m = 7$	$m = 8$	$m = 9$
$d_{\min} = \sqrt{98}/2^7$	$d_{\min} = \sqrt{208}/2^8$	$d_{\min} = \sqrt{392}/2^9$
≈ 0.07733980	≈ 0.05633674	≈ 0.03866990

$$\begin{pmatrix}1&0&0&0&0&0&1\\0&1&0&0&0&1&0\\0&0&1&0&1&0&0\\0&0&0&1&0&0&0\\0&0&1&0&0&0&0\\0&1&1&0&0&0&0\\1&0&0&0&0&0&0\end{pmatrix}\quad\begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&1&1&0&0&0&0&0\\0&1&1&1&0&0&0&0\\0&0&0&0&1&1&0&0\\0&0&0&1&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\end{pmatrix}\quad\begin{pmatrix}1&0&0&0&0&0&0&0&1\\1&1&0&0&0&0&0&1&0\\0&0&1&0&0&0&1&0&0\\0&0&0&1&0&1&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&1&0&0&0&0&0\\0&0&1&1&0&0&0&0&0\\0&1&0&0&0&0&0&0&0\\1&0&0&0&0&0&0&0&0\end{pmatrix}$$

Fig. 12. Partial results of the restricted search for $(0, m, 2)$-nets in base 2.

3.3 Construction Inferred from Computer Search

For $m \geq 8$ we suggest the generator matrices

$$\begin{pmatrix} 1 & 0 & & \cdots & & 0 \\ 0 & \ddots & & & & \\ & & 1 & & & \\ \vdots & & & \boxed{C_2^{(6)}} & & \vdots \\ & & & & 1 & \\ & & & & \ddots & 0 \\ 0 & & & \cdots & 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 & & \cdots & & 0 \\ 0 & \ddots & & & & \\ & & 1 & & & \\ \vdots & & & \boxed{C_2^{(7)}} & & \vdots \\ & & & & 1 & \\ & & & & \ddots & 0 \\ 0 & & & \cdots & 0 & 1 \end{pmatrix}$$

for even $m \geq 8$ for odd $m \geq 9$,

where $C_2^{(6)}$ and $C_2^{(7)}$ are the matrices for $m = 6, 7$ given in Figure 10 and Figure 11. The matrices extended by ones on the diagonals still fulfill the conditions of Theorem 1. Hence the resulting nets are $(0, m, 2)$-nets in base 2 for $m \geq 8$.

We would like to note that in general there are many matrices of size $(m-2) \times (m-2)$ with nets having the same minimum distance. However, when using these matrices for the inferred construction the resulting nets might have different minimum distances. The restricted search from the previous Section 3.2 was only computationally feasible up to $m = 9$, but did not find better results.

Resulting New Construction and Numerical Results

Combining the findings of the previous sections, we propose to use the results from

- the full matrix search for $m \leq 7$,
- and the inferred construction for $m \geq 8$.

The points generated by these matrices were compared to the Larcher-Pillichshammer- (for an implementation see [KK02]) and the Hammersley-net (see Table 1 and Figure 4). It turned out that the new construction performed better than the Larcher-Pillichshammer-net with respect to the toroidal minimum distance. In fact the Hammersley-net always generates the worst possible minimum distance of $\sqrt{2}/2^m$ for our constructions. This distance is found between the first and the last point of the net. The spectral properties of these nets can be visually compared in Figure 5.

3.4 Exhaustive Permutation Search

In order to verify the results, we searched the space of the permutation-generated $(0, m, 2)$-nets in base 2. For $m < 5$ the search produced nets with the same minimum distance as the new construction (see Table 1). However, for $m = 5$ points were found which cannot be generated using matrices. These also exhibited a better minimum distance ($\sqrt{32}/2^5 \approx 0.17677670$) compared to all matrix-generated nets (the maximum is $\sqrt{29}/2^5 \approx 0.16828640$ for the new construction). Unfortunately, for $m > 5$ this space is way too large for an exhaustive search.

4 Conclusion

By maximizing the minimum distance of a point set, we removed a degree of freedom from the definition of (t, m, s)-nets. Although an exhaustive computer search is infeasible, two interesting facts could be revealed by examples:

1. Nets with a quality parameter $t > 0$ can obtain a minimum distance larger than those with $t = 0$.
2. Permutation-generated nets can obtain a larger minimum distance than matrix-generated nets.

We will continue our research in this direction and search for $(0, m, 3)$-nets in base 2 with maximized minimum distance. We plan to investigate the practical benefits of the new concepts in the setting of computer graphics, namely the realistic simulation of light transport along the lines of [CCC87, KK02, WK07].

Acknowledgements

The first two authors have been funded by the project *information at your fingertips - Interaktive Visualisierung für Gigapixel Displays*, Forschungsverbund im Rahmen des Förderprogramms Informationstechnik in Baden-Württemberg (BW-FIT). Our special thanks go to Sun Microsystems GmbH who kindly supported our research with computing equipment.

References

[CCC87] R. Cook, L. Carpenter, and E. Catmull. The REYES Image Rendering Architecture. In *Computer Graphics (SIGGRAPH '87 Conference Proceedings)*, pages 95–102, July 1987.

[CP76] R. Cranley and T. Patterson. Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis*, 13:904–914, 1976.

[CPC84] R. Cook, T. Porter, and L. Carpenter. Distributed Ray Tracing. In *Computer Graphics (SIGGRAPH '84 Conference Proceedings)*, pages 137–145, 1984.

[FK02] I. Friedel and A. Keller. Fast Generation of Randomized Low-Discrepancy Point Sets. In H. Niederreiter, K. Fang, and F. Hickernell, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 257–273. Springer, 2002.

[Gla95] A. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, 1995.

[HA90] P. Haeberli and K. Akeley. The Accumulation Buffer: Hardware Support for High-Quality Rendering. In *Computer Graphics (SIGGRAPH '90 Conference Proceedings)*, pages 309–318, 1990.

[Kel03] A. Keller. Strictly Deterministic Sampling Methods in Computer Graphics. *SIGGRAPH 2003 Course Notes, Course #44: Monte Carlo Ray Tracing*, 2003.

[Kel04] A. Keller. Trajectory Splitting by Restricted Replication. *Monte Carlo Methods and Applications*, 10(3-4):321–329, 2004.

[Kel06] A. Keller. Myths of Computer Graphics. In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 217–243. Springer, 2006.

[KH01] A. Keller and W. Heidrich. Interleaved Sampling. In K. Myszkowski and S. Gortler, editors, *Rendering Techniques 2001 (Proc. 12th Eurographics Workshop on Rendering)*, pages 269–276. Springer, 2001.

[KK02] T. Kollig and A. Keller. Efficient Multidimensional Sampling. *Computer Graphics Forum*, 21(3):557–563, September 2002.

[LP01] G. Larcher and F. Pillichshammer. Walsh Series Analysis of the L_2-Discrepancy of Symmetrisized Point Sets. *Monatsh. Math.*, 132:1–18, 2001.

[Nie92] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, 1992.

[Owe95] A. Owen. Randomly Permuted (t, m, s)-Nets and (t, s)-Sequences. In H. Niederreiter and P. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, volume 106 of *Lecture Notes in Statistics*, pages 299–315. Springer, 1995.

[Owe98] A. Owen. Monte Carlo Extension of Quasi-Monte Carlo. In *Winter Simulation Conference*, pages 571–577. IEEE Press, 1998.

[Pan04] F. Panneton. *Construction d'ensembles de points basé sur une récurrence linéaire dans un corps fini de caractéristique 2 pour la simulation Monte Carlo et l'intégration quasi-Monte Carlo*. PhD thesis, Université de Montréal, 2004.

[PS01] G. Pirsic and W. Ch. Schmid. Calculation of the quality parameter of digital nets and application to their construction. *J. Complexity*, 17(4):827–839, 2001.

[PTVF92] H. Press, S. Teukolsky, T. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.

[Rol05] T. Rolfe. Optimal Queens - A classical problem solved by backtracking. *Dr. Dobb's Journal*, 30(372), May 2005.

[Sob67] I. Sobol'. On the Distribution of Points in a Cube and the approximate Evaluation of Integrals. *Zh. vychisl. Mat. mat. Fiz.*, 7(4):784–802, 1967.

[WK07] C. Wächter and A. Keller. Efficient Simultaneous Simulation of Markov Chains. In A. Keller, S. Heinrich, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2006*, in this volume. Springer, 2007.

[Yel83] J. Yellot. Spectral Consequences of Photoreceptor Sampling in the Rhesus Retina. *Science*, 221:382–385, 1983.