Holographic Representations of Images

Alfred M. Bruckstein, Robert J. Holt, and Arun N. Netravali, *Fellow, IEEE*

***Abstract*— We discuss a new type of holographic image representations that have advantages in a "distributed" world. We call these representations *holographic*. Arbitrary portions of a holographic representation enable reconstruction of the whole image, with distortions that decrease gradually with the increase in the size of the portions available. Holographic representations enable progressive refinement in image communication or retrieval tasks, with no restrictions on the order in which the data fragments (sections of the representation) are accessed or become available.**

***Index Terms*—Holographic representations, progressive refinement, pseudorandom uniform samplings, random phase Fourier transforms.**

I. Introduction

FROM arbitrary portions of optical holograms that encode a scene, the entire scene can be reconstructed. The quality of the view recovered depends on the size of the hologram portion used, but is independent of the place from where it was cropped. A small portion of the hologram provides a blurred scene reconstruction, and as the area of the portion grows, the scene is reconstructed with more and more detail. The information on the scene is homogeneously distributed in its holographic representation, in such a way that even small portions of the hologram contain a global view, with details missing. Hence, an optical hologram of a three-dimensional (3-D) scene is a representation that makes possible a successive refinement scheme for information retrieval and transmission, insensitive to the order in which the portions of the representation become available.

In this paper we discuss two ways of representing an image holographically. These representations do not simulate the process of optical holography, yet they achieve the property of distributing the image information "uniformly" in a way that enables successive refinement insensitive to the order in which the data portions are made available to the user.

The first method for holographic image representation is based on a sampling scheme that is systematic yet simulates the properties of random sampling of the image plane. This representation maps the image into a sequence of sample pixels, so that any portion of this sequence contains image samples that are spread over the image plane as uniformly as possible. We propose a method to achieve such a uniform sampling derived using number theoretic considerations. Once we have a set of uniformly spread samples available, the image can be approximately reconstructed in many possible ways. We have tested reconstruction methods based on pixel replication and on interpolations using 1/(distance) and 1/(distance)^2 weighting filters that yielded better results. We point out, however, that an in-depth analysis of image estimation from the sparse samples is beyond the scope of this paper.

The second method for holographic representation was motivated by our image watermarking scheme [3] that modifies the spectrum of an image by a real multiplicative mask of gains that either slightly enhance or slightly depress frequency components over various regions in the Fourier domain. The so-embedded watermark is "holographic" in the sense that even arbitrarily cropped portions of the image will enable recovery of the entire watermark by comparison with the original image. The watermark mask (an image itself, encoding the spectral modifications made) is thus represented in the image in a distributed fashion and this is due to the uncorrelated image phase that is effectively added to it. Hence, our second idea for holographic representation is based on regarding the image as the frequency domain magnitude of a random phase image in the "representation domain."

Encoding an image after multiplication by a random phase diffuser is a common practice in optical holography, the aim of this trick being the equalization of the hologram magnitude to be recorded in media with rather restricted linear dynamic ranges. The diffuser indeed spreads the amplitude information evenly in the Fourier domain. However, the idea presented herein differs from the optical holography process and has never been proposed as a method for distributed image representation.

II. Holographic Sampling of Images

In this section, we explore the idea of sampling the image sequentially in a way that will yield a holographic representation. The original two-dimensional (2-D) array of image pixels is thereby stacked into a sequence—a long one-dimensional (1-D) array—so that any portion of this sequence will correspond to a sampling of the image, with samples spatially distributed as uniformly as possible over the array. In order to know the exact location of the samples, the portion of the sequence has to be specified, and we assume that this issue is indeed solved by some indexing method specifying the 2-D location of some of the pixel values (with some memory overhead!).

An ideal situation would occur if there was an ordering of the pixels of a $2^N \times 2^N$ image such that every $2^M \times 2^M$ square from the original image would be represented by any

Manuscript received January 23, 1997; revised January 27, 1998. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Fabrice Heitz.

A. M. Bruckstein is with Technion—Israel Institute of Technology, 32000 Haifa, Israel (e-mail: freddy@cs.technion.ac.il).

R. J. Holt and A. N. Netravali are with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: rjh@research.bell-labs.com; ann@research.bell-labs.com).

Publisher Item Identifier S 1057-7149(98)07794-X.

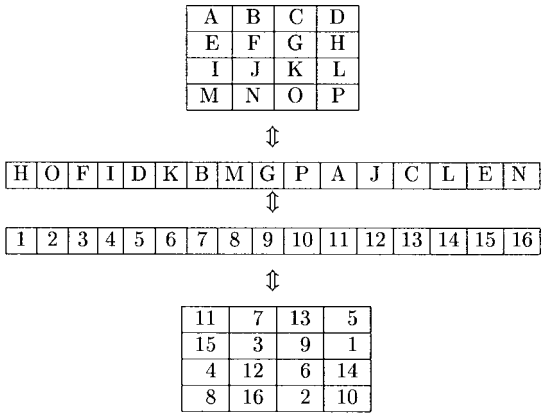


Fig. 1. Possible pixel ordering for a 4 × 4 image and the corresponding pixel sequence.

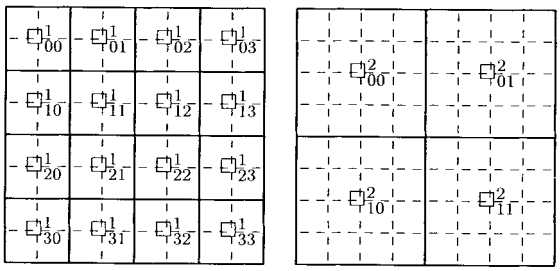


Fig. 2. Decomposing the image into disjoint blocks.

sequence of 4^{N-M} consecutive pixels. For example, consider the ordering shown in Fig. 1. The bottom square array in Fig. 1 shows the order in which the pixels of the top square array are chosen to make up the pixel sequence. Hereinafter, square arrays of integers are used as a compact way to represent the pixel ordering. In this ordering, the sequence 1-2-3-4, or H-O-F-I, covers all nine 2 × 2 squares in Fig. 1. However, the sequence 2-3-4-5 misses one 2 × 2 square, namely that denoted by pixels GHKL.

Unfortunately, the ideal ordering turns out to be impossible, and this is shown in Appendix A. Since the ideal ordering turns out not to be possible, we develop an ordering for which only certain nonoverlapping $2^M \times 2^M$ blocks of squares from the original image are guaranteed to be represented by any sequence of 4^{N-M} consecutively numbered pixels. These blocks will be denoted by $\square_{kl}^M$, where $\square_{kl}^M$ represents the $2^M \times 2^M$ square array of pixels whose upper left corner is at row $2^M k$ and column $2^M l$. Examples for $N = 3$ and $M = 1, 2$ are illustrated in Fig. 2. Thus $\square_{kl}^M$ comprises the pixels with row index in $\{2^M k, \cdots, 2^M(k+1) - 1\}$ and column index in $\{2^M l, \cdots, 2^M(l+1) - 1\}$. Also, the entire image may be broken into the 4^{N-M} disjoint blocks $\square_{00}^M, \square_{01}^M, \cdots, \square_{0(2^{N-M}-1)}^M, \square_{10}^M, \cdots, \square_{(2^{N-M}-1)(2^{N-M}-1)}^M$.

A basic sampling pattern for square images is illustrated in Fig. 3. Given four pixels in a 2 × 2 square, we put them in the order upper left, lower right, upper right, lower left. There is not a lot of choice involved here, but it is slightly better to have as many diagonally opposite pixels numbered consecutively as possible.

0	2
3	1

Fig. 3. Basic pattern for pixel ordering.

0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

0	32	8	40	2	34	10	42
48	16	56	24	50	18	58	26
12	44	4	36	14	46	6	38
60	28	52	20	62	30	54	22
3	35	11	43	1	33	9	41
51	19	59	27	49	17	57	25
15	47	7	39	13	45	5	37
63	31	55	23	61	29	53	21

Fig. 4. Regular holographic sampling for a 4 × 4 and an 8 × 8 image.

One possibility to generate sampling schemes with "holographic" properties in the sense described above is to iterate this pattern for larger arrays. These sampling schemes have the property that sample subsequences of the same length yield images of similar quality, and the image distortion decreases regularly as the length of the sample increases. A 4 × 4 grid is split into four 2 × 2 squares. Now the four upper left corners of the 2 × 2 squares are filled with 0 to 3 following the pattern of Fig. 3. Second, the four lower right corners are filled with 4 to 7, again following the same pattern. Third, the four upper right corners are numbered 8 to 11, and finally the four lower left corners are numbered 12 to 15. The result is shown in Fig. 4. For an 8 × 8 array, we can split the grid into sixteen 2 × 2 squares. First, the 16 upper left corners of the 2 × 2 squares are filled in with 0 to 15 in the pattern of Fig. 4. Then the 16 lower right corners, the 16 upper right corners, and the 16 lower left corners, are filled in, as shown in Fig. 4. This ordering has some similarities to the "farthest point sampling" espoused in [7]. After the first four pixels are selected, the next four chosen pixels are located in the four positions that are farthest away from the original four. The next eight pixels are chosen to be at the eight locations farthest away from the first eight, and so on. There are also similarities to the "well-spaced labelings" recently introduced by Lagarias in [8], where the goal is to separate all sequences of consecutive integers as much as possible when they are placed in a 2-D memory array. The motivation of the work in [8] is the storage of 1-D signals (music, speech, etc.) in memory arrays in such a way that if a contiguous blob of the memory cells fails, the signal will still be recoverable with sufficient perceptibility. Our objective here includes the constraint that each square $\square_{kl}^M$ is visited periodically, so the goals here and in [8] are somewhat different. However, a contiguous blob loss is equivalent to having contiguous strings intact, so the regular holographic sampling method is useful under these circumstances.

The above enumeration of the array elements guarantees that if a $2^N \times 2^N$ array is divided into the 4^{N-M} disjoint $2^M \times 2^M$ squares $\square_{kl}^M$ for $0 \le M \le N$, then any sequence of 4^{N-M} consecutive integers will have one term in each of the $\square_{kl}^M$. The sequence may "wrap around," so that 0 is considered to follow $4^N - 1$. A consequence of this ordering is that the pixels corresponding to any string of at least four consecutive

0	128	32	160	8	136	40	168	**2**	130	34	162	10	138	42	170
192	64	**224**	96	**200**	72	**232**	104	194	66	**226**	98	**202**	74	**234**	106
48	176	16	144	56	184	24	152	50	178	18	146	58	186	26	154
240	112	**208**	80	**248**	120	**216**	88	**242**	114	**210**	82	**250**	122	**218**	90
12	140	44	172	**4**	132	36	164	14	142	46	174	6	134	38	166
204	76	**236**	108	196	68	**228**	100	**206**	78	**238**	110	**198**	70	**230**	102
60	188	28	156	52	180	20	148	62	190	30	158	54	182	22	150
252	124	**220**	92	**244**	116	**212**	84	**254**	126	**222**	94	**246**	118	**214**	86
3	131	35	163	11	139	43	171	**1**	129	33	161	9	137	41	169
195	67	**227**	99	**203**	75	**235**	107	193	65	**225**	97	**201**	73	**233**	105
51	179	19	147	59	187	27	155	49	177	17	145	57	185	25	153
243	115	**211**	83	**251**	123	**219**	91	**241**	113	**209**	81	**249**	121	**217**	89
15	143	47	175	7	135	39	167	13	141	45	173	5	133	37	165
207	79	**239**	111	**199**	71	**231**	103	**205**	77	**237**	109	**197**	69	**229**	101
63	191	31	159	55	183	23	151	61	189	29	157	53	181	21	149
255	127	**223**	95	**247**	119	**215**	87	**253**	125	**221**	93	**245**	117	**213**	85

Fig. 5. Sixty-four consecutive integers, the boldface entries 197–255 and 0–4, represent each of the 64 2 × 2 squares (regular holographic sampling).

integers are reasonably uniformly distributed about the entire image. Specifically, since any sequence of length 4^{N-M} hits each $\square_{kl}^{M}$, every pixel in the image is at distance at most the length of the diagonal of one of these squares, namely $2^{-(N-M-1/2)}$ times the length of the edge of the image, from one of the pixels in the sequence. An example with $N = 4$ and $M = 1$ is shown in Fig. 5.

The above construction has the nice property that every sequence of 4^M consecutive integers in $[0, 4^N - 1]$ has a representative in each of the 4^{N-M} squares $\square_{kl}^{M}$, $k, l \in \{0, \cdots, 2^M - 1\}$. However, this rather uniform, or "regular holographic," sampling frequently leads to aliasing and/or Moiré effects, particularly when in the sampled image there are patterns of high spatial frequency [4]–[7], [9].

Fig. 6 shows the images we used in a series of experiments with holographic encoding of images. Figures 6(a), (b) are two 256 × 256 synthetic images: 6(a) is a three-gray-level image, and 6(b) is the sampling of a radial function of the form

$$I(\rho, \phi) = 128\rho\left[1 + \cos\left(2\pi \frac{a}{b\frac{|\phi - \pi|}{\pi} + c}\right)\right]$$

over $[0,1]\times[0,1]$, where $\rho^2 = x^2 + y^2$, $\tan\phi = y/x$, and a, b, c are chosen so that $a/(b + c)$ and a/c are integers. Fig. 6(c), (d), and (e) show 512 × 512 images used in experiments. Fig. 6(c) is a three-gray-level image similar to Fig. 6(a), 6(d) is a synthetic continuous-tone image based on the same formula as Fig. 6(b), and 6(e) is a portion of an old engraving that was scanned in. The engraving example is a particularly difficult test case since it has a fine structure consisting of a spiral line whose modulation was apparently used by the artist to produce the gray scale. Fig. 6(f) and (g) are the standard Barbara and Goldhill images.

Using Fig. 6(e) as the original image, the Moiré effects are seen in Figs. 9(a), 10(a), and 11(a), exhibiting images reconstructed from holographic samplings using the regular strategy described above. Better results were obtained when uniform samples were "jittered randomly." Here, we do not want a truly random pattern, since we need to be able to reconstruct the sampling pattern at will, so we simulate a random pattern (a pseudorandom sequence) by a carefully chosen pixel ordering described below.

The indices of the four pixels of the 2 × 2 array in Fig. 3 are $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$ when one starts counting at zero. These four ordered pairs may be regarded as binary numbers and associated with the (decimal) integers 0, 1, 2 and 3. Thus we may say that the integers 0, 1, 2, 3 were placed in the 0, 3, 1, 2 positions, respectively, in the grid in Fig. 3. In Fig. 4, the integers 0, 1, 2, 3 were all placed in the "0" position in squares $\square_{00}^{1}$, $\square_{11}^{1}$, $\square_{01}^{1}$, $\square_{10}^{1}$, respectively. A placement that appears more random would have these four integers in different positions in those four squares, say in the 0, 2, 3, 1, positions, respectively. Now 4, 5, 6, 7 have to be placed at different locations inside $\square_{00}^{1}$, $\square_{11}^{1}$, $\square_{01}^{1}$, $\square_{10}^{1}$, so we shift the sequence of positions 0, 2, 3, 1 to 1, 3, 0, 2. Then 8, 9, 10, 11 are placed in positions 2, 0, 1, 3 in $\square_{00}^{1}$, $\square_{11}^{1}$, $\square_{01}^{1}$, $\square_{10}^{1}$, and 12, 13, 14, 15 are place at positions 3, 1, 2, 0. The final "pseudorandom" sampling pattern result is exhibited in Fig. 7.

For an 8 × 8 array, we can split the grid into the sixteen 2 × 2 squares $\square_{kl}^{1}$, $k, l \in \{0, 1, 2, 3\}$. We choose another permutation, say 0, 1, 2, 3 this time, for the positions of integers 0, 1, 2, 3 within $\square_{00}^{1}$, $\square_{32}^{1}$, $\square_{13}^{1}$, $\square_{21}^{1}$, the squares corresponding to those integers' placements in Fig. 7. Then we shift the permutation to 2, 3, 0, 1 for the locations of 4, 5, 6, 7 within $\square_{00}^{1}$, $\square_{32}^{1}$, $\square_{13}^{1}$, $\square_{21}^{1}$. Continuing in this manner, the locations of 8, 9, 10, 11 are 1, 2, 3, 0 and those of 12, 13, 14, 15, are 3, 0, 1, 2. The results appear more random, and the resulting image reconstructions show weaker aliasing effects, with these positionings for 4, 8, and 12, than when these three integers are given positions 1, 2, and 3, and that pattern is continued.

To take care of the next 16 integers, the sequence of positions for 0 through 15, which is 0, 1, 2, 3, 2, 3, 0, 1, 1, 2, 3, 0, 3, 0, 1, 2, is shifted to 1, 2, 3, 0, 3, 0, 1, 2, 2, 3, 0, 1, 0, 1, 2, 3 for 16 through 31. The integers 32 through 63 are filled in after two similar shifts, with 32 and 48 going into the 2 and 3 positions in $\square_{00}^{1}$, respectively. The final result is shown in Fig. 7.

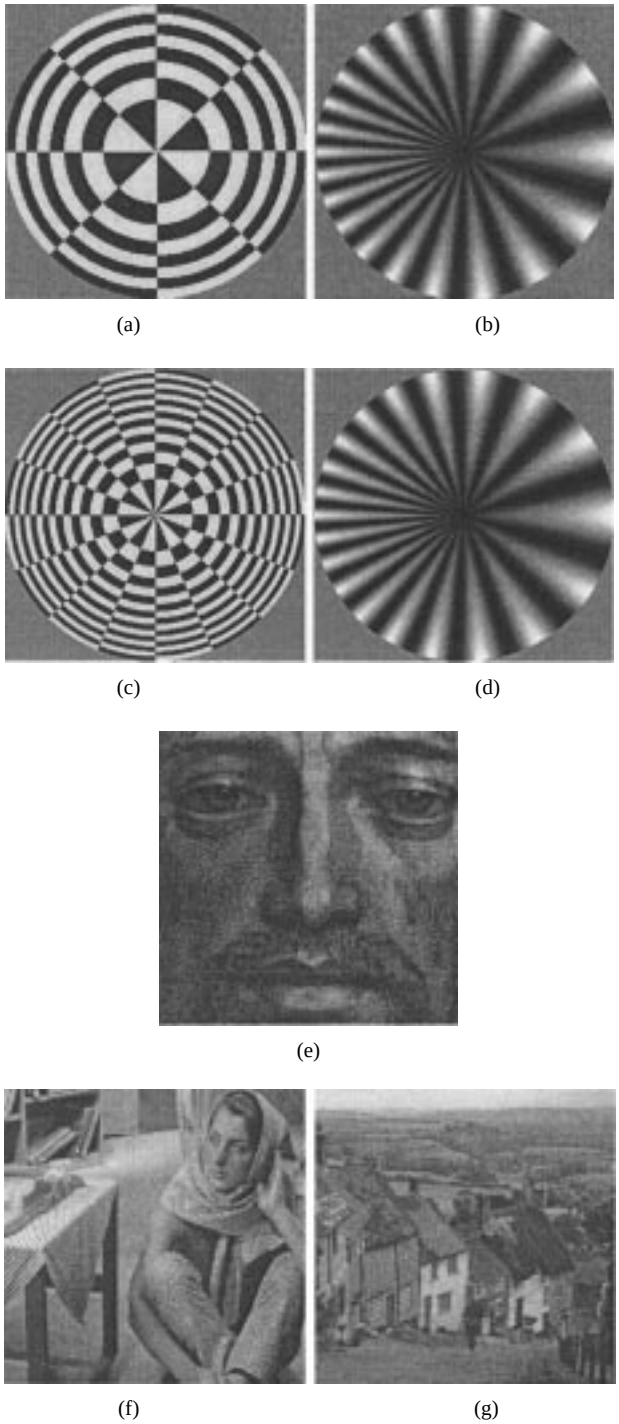


Fig. 6. (a)–(b) Original test images of size 256 × 256. (c)–(g) Test images of size 512 × 512. (a)–(d) are synthetic, with (a) and (c) having three distinct gray levels, while (b) and (d) have continuous tones. (e) is a scanned engraving. (f) is the Barbara image. (g) is the Goldhill image.

In general, we want to mix up the various permutations of $\{0,1,2,3\}$ as much as possible to avoid the Moiré effects mentioned above. Since we are allowing integer sequences to wrap around 4^N, there is no loss in generality in assigning the integer zero to position 0 at each stage. If some other integer was assigned to position 0, we could get an equivalent ordering by subtracting (modulo 4^N) that integer from each integer in the grid.

0	4	6	10
8	12	14	2
15	3	9	13
7	11	1	5

0	16	36	52	6	22	26	42
32	48	4	20	38	54	58	10
56	8	28	44	62	14	34	50
24	40	60	12	30	46	2	18
47	63	19	35	41	57	13	29
15	31	51	3	9	25	45	61
55	7	11	27	49	1	21	37
23	39	43	59	17	33	53	5

Fig. 7. Pixel ordering for a 4 × 4 image and an 8 × 8 image.

We now give a precise description of where each integer goes into a $2^N \times 2^N$ grid. Define the permutation functions on $\{0,1,2,3\}$ as follows:

$$\begin{array}{llll} h_{6i+1}(0)=0 & h_{6i+1}(1)=1 & h_{6i+1}(2)=2 & h_{6i+1}(3)=3 \\ h_{6i+2}(0)=0 & h_{6i+2}(1)=2 & h_{6i+2}(2)=1 & h_{6i+2}(3)=3 \\ h_{6i+3}(0)=0 & h_{6i+3}(1)=2 & h_{6i+3}(2)=3 & h_{6i+3}(3)=1 \\ h_{6i+4}(0)=0 & h_{6i+4}(1)=1 & h_{6i+4}(2)=3 & h_{6i+4}(3)=2 \\ h_{6i+5}(0)=0 & h_{6i+5}(1)=3 & h_{6i+5}(2)=1 & h_{6i+5}(3)=2 \\ h_{6i+6}(0)=0 & h_{6i+6}(1)=3 & h_{6i+6}(2)=2 & h_{6i+6}(3)=1 \end{array} \tag{1}$$

and $h_i(4k+m) = h_i(m)$ for all nonnegative integers i, k, and m. Also let

$$\begin{array}{lll} g_{6i+1}(n)=h_5(n) & g_{6i+2}(n)=h_3(n) & g_{6i+3}(n)=h_1(n) \\ g_{6i+4}(n)=h_6(n) & g_{6i+5}(n)=h_2(n) & g_{6i+6}(n)=h_4(n) \end{array}$$

for all nonnegative integers i. Next, let

$$f(n,a) = \left[g_{a+1}(n) + \sum_{b=1}^{a} h_{a-b+1}\left(\left\lfloor \frac{n}{4^b} \right\rfloor \right) \right] \pmod 4 \tag{2}$$

where $\lfloor \cdot \rfloor$ is the floor function ($\lfloor x \rfloor$ is the greatest integer less than or equal to x), and define

$$\begin{aligned} r(n,a) &= \begin{cases} 0, & \text{if } f(n,a) \in \{0,1\} \\ 1, & \text{if } f(n,a) \in \{2,3\} \end{cases} \\ c(n,a) &= \begin{cases} 0, & \text{if } f(n,a) \in \{0,2\} \\ 1, & \text{if } f(n,a) \in \{1,3\} \end{cases}. \end{aligned} \tag{3}$$

Then the integer n is located at pixel (i,j), where

$$i = \sum_{a=0}^{N-1} 2^{N-a-1} r(n,a) \quad j = \sum_{a=0}^{N-1} 2^{N-a-1} c(n,a). \tag{4}$$

This ordering is more easily understood with an example. Consider the integer 45 in the second array of Fig. 7. The formulas above, with $N=3$, give

$$\begin{aligned} f(45,0) &= g_1(45) = h_5(1) = 3 \\ f(45,1) &= g_2(45) + h_1(11) \\ &= h_3(1) + h_1(3) = 2 + 3 \pmod 4 = 1 \\ f(45,2) &= g_3(45) + h_2(11) + h_1(2) \\ &= h_1(1) + h_2(3) + h_1(2) \\ &= 1 + 3 + 2 \pmod 4 = 2. \end{aligned}$$

Thus, by (4) the row index for 45 is $2^2 1 + 2^1 0 + 2^0 1 = 5$ and the column index is $2^2 1 + 2^1 1 + 2^0 0 = 6$.

Considering (1)–(4), and in particular the binary representations expressed in (4), if two integers n_1 and n_2 are located at the same pixel, then $r(n_1,a) = r(n_2,a)$ and $c(n_1,a) =$

0	64	144	**208**	100	164	**244**	52	6	70	150	**214**	90	154	170	**234**
128	192	16	80	**228**	36	116	180	134	**198**	22	86	**218**	26	42	106
224	32	112	176	132	196	20	84	**230**	38	118	182	58	122	**202**	10
96	160	**240**	48	**4**	68	148	**212**	102	166	**246**	54	186	**250**	74	138
184	**248**	72	136	92	156	172	**236**	190	**254**	78	142	98	162	**242**	50
56	120	**200**	8	**220**	28	44	108	62	126	**206**	14	**226**	34	114	178
216	24	40	104	60	124	**204**	12	**222**	30	46	110	130	194	18	82
88	152	168	**232**	188	**252**	76	140	94	158	174	**238**	**2**	66	146	**210**
111	175	**255**	63	83	147	163	**227**	105	169	**249**	57	13	77	157	**221**
239	47	127	191	**211**	19	35	99	**233**	41	121	185	141	**205**	29	93
143	**207**	31	95	51	115	195	**3**	137	**201**	25	89	**237**	45	125	189
15	79	159	**223**	179	**243**	67	131	9	73	153	**217**	109	173	**253**	61
183	**247**	71	135	11	75	155	**219**	177	**241**	65	129	85	149	165	**229**
55	119	**199**	7	139	**203**	27	91	49	113	193	**1**	**213**	21	37	101
215	23	39	103	**235**	43	123	187	**209**	17	33	97	53	117	**197**	5
87	151	167	**231**	107	171	**251**	59	81	145	161	**225**	181	**245**	69	133

Fig. 8. Sixty-four consecutive integers, the boldface entries 197–255 and 0–4, represent each of the sixty-four 2×2 squares.

$c(n_2, a)$ for $a = 0, \cdots, N-1$. Then $f(n_1, a) = f(n_2, a)$ for $a = 0, \cdots, N-1$. Now from (2) with $a = 0$, we get $g_1(n_1) = g_1(n_2)$, so $n_1 \equiv n_2 \,(\mathrm{mod}\, 4)$. Next, from (2) with $a = 1$, we get $g_2(n_1) + h_1(\lfloor n_1/4 \rfloor) \equiv g_2(n_2) + h_1(\lfloor n_2/4 \rfloor) \,(\mathrm{mod}\, 4)$. Since $n_1 \equiv n_2 \,(\mathrm{mod}\, 4)$, we have $g_2(n_1) = g_2(n_2)$, and thus $h_1(\lfloor n_1/4 \rfloor) \equiv h_1(\lfloor n_2/4 \rfloor) \,(\mathrm{mod}\, 4)$, which implies $\lfloor n_1/4 \rfloor \equiv \lfloor n_2/4 \rfloor \,(\mathrm{mod}\, 4)$, and thus $n_1 \equiv n_2 \,(\mathrm{mod}\, 4^2)$ as $n_1 \equiv n_2 \,(\mathrm{mod}\, 4)$. Continuing in this manner, at the last step, (2) with $a = N-1$, we get $n_1 \equiv n_2 \,(\mathrm{mod}\, 4^N)$. Therefore, each integer n in $[0, 4^N - 1]$ is located at a distinct pixel.

This enumeration of the array elements has the same properties as the regular sampling method described above in that each sequence of consecutive integers of length 4^{N-M} has one representative in each $\square_{kl}^M$. An example with $N = 4$ and $M = 1$ is shown in Fig. 8.

The proof of the following is given in Appendix B.

Theorem 1: Let a $2^N \times 2^N$ array for any $N \geq 0$ be numbered according to the rules expressed by (1), (2), (3), and (4). For M such that $0 \leq M \leq N$, divide the array into the 4^{N-M} disjoint $2^M \times 2^M$ squares $\square_{kl}^M, k, l \in \{0, \cdots, 2^{N-M} - 1\}$. Then exactly one element of any sequence of 4^{N-M} consecutive integers in $[0, 4^N - 1]$ appears in each of the squares $\square_{kl}^M$. The sequence may wrap around, with 0 following $4^N - 1$.

We now return to the regular holographic sampling that was first described in this section. Using the same notation as (1), we find that (3) and (4) hold in this case when in place of (2), $f(n, a)$ is simply defined as

$$f(n, a) = h_5\Big(\Big\lfloor \frac{n}{4^a} \Big\rfloor\Big). \tag{5}$$

As an example, consider the integer 27 in Fig. 5. Formulas (1), (5), (3), (4) yield

$$\begin{aligned} f(27, 0) &= h_5(27) = h_5(3) = 2 \\ f(27, 1) &= h_5(6) = h_5(2) = 1 \\ f(27, 2) &= h_5(1) = 3 \\ f(27, 3) &= h_5(0) = 0. \end{aligned}$$

Thus by (4) the row index for 27 is $2^3 1 + 2^2 0 + 2^1 1 + 2^0 0 = 10$ and the column index is $2^3 0 + 2^2 1 + 2^1 1 + 2^0 0 = 6$.

For this regular sampling we have the following theorem, whose proof is similar to that of Theorem 1 (given in Appendix B) and will not be repeated.

Theorem 2: Let a $2^N \times 2^N$ array for any $N \geq 0$ be numbered according to (1, 5, 3, 4). For M such that $0 \leq M \leq N$, divide the array into the 4^{N-M} disjoint $2^M \times 2^M$ squares $\square_{kl}^M, k, l \in \{0, \cdots, 2^{N-M} - 1\}$. Then exactly one element of any sequence of 4^{N-M} consecutive integers in $[0, 4^N - 1]$ appears in each of the squares $\square_{kl}^M$. The sequence may wrap around, with 0 following $4^N - 1$.

After a certain quantity of pixels, say 4^{N-M} out of 4^N with $0 < M < N$, have been selected, there remains the problem of how to fill in the remaining pixels of the reconstructed image. A simple method is to color all the pixels of $\square_{kl}^M$ with the same shade of gray as that of the representative pixel of that square. This technique was used to construct the images in Figs. 9(b), 10(b), and 11(b). In the latter two, the uniformly colored squares are quite noticeable, and the resulting figures are not aesthetically pleasing. A better method is to use some type of smoothing, where each unfilled-in pixel is assigned a weighted of average of its nearest neighbors that are available. To accomplish this we need to use a sliding window large enough to guarantee that at least one pixel in the window will be filled in for all positions of the window center. For the orderings described above, where each $2^M \times 2^M$ square is guaranteed to be represented by a sequence of length 4^{N-M}, a square window of side $2^{M+1} - 1$ centered about an unselected pixel is the smallest window which must contain at least one pixel from the sequence (for $M < N$). When more than one selected pixel is in such a window, their intensities are weighted by some function of their distance d, such as $1/d$ or $1/d^2$, to the pixel in the center of the window, and averaged. Some examples of $1/d$ and $1/d^2$ smoothing are shown in Figs. 9(c) and (d), 10(c) and (d), and 11(c) and (d).

To determine which type of smoothing is better in general, the root mean square error (RMSE) was computed for many images. This error is defined for a $2^N \times 2^N$ image $I_1(\cdot, \cdot)$

compared to the original image $I_0(\cdot,\cdot)$ as

$$\mathrm{RMSE}(I_0, I_1) = \frac{1}{4^N}\left\{\sum_{i=1}^{2^N}\sum_{j=1}^{2^N}\left[\frac{I_1(i,j) - I_0(i,j)}{I_{\max}}\right]^2\right\}^{1/2}.$$

(The individual value $I(i,j)$ is the gray-level value at pixel (i,j), on a scale of 0 to $I_{\max} = 255$). It was found that both types of smoothing were a significant improvement over the coarse pixel replications, namely the regular holographic sampling method and the "random" sampling method without smoothing, as indicated in Figs. 9–11. In the majority of cases, the $1/d$ smoothing had a smaller RMSE than the $1/d^2$ smoothing, so $1/d$ smoothing is used in the remaining examples. The RMSE generally increases close to linearly with the logarithm of the ratio of the sample length to the image size until it levels off at a point where the distorted image is unrecognizable.

To illustrate the claim that the reconstructed images obtained by different integer sequences are similar, we ran our program on the data that generated Fig. 10(c) using several different starting points. Specifically, the pixels were numbered from 0 to $2^{18} - 1 = 262\,143$ in accordance with (1), (2), (3), and (4), and the results shown in Figs. 10(c) and 12 were those obtained with sampling sequences of length 128×128 (2^{14}), the starting pixel of the sequence being 83298, 27561, 133476, 193002, and 241 785. While tiny differences may be observed upon close inspection, the general overall images are quite similar to the naked eye. Furthermore, the RMSE's are all close to each other, ranging from 0.112 to 0.113.

While in these examples we considered sampling sequence lengths that were powers of four, that is definitely not a restriction of the sampling scheme described. The reconstructed images for sequences of general length are, as expected, of increasing quality as the length of the sequence of pixels increases from one power of four to the next when the smoothing operation above is utilized. As an example, we have Fig. 13, where the number of sample pixels is shown as a percentage of the number of pixels in the original image. The "random" sampling does have the feature for that any sequence of length $2^{2(N-M)-1}$ for $0 \le M < N$, each of the $2^{M+1} \times 2^M$ rectangles obtained by joining $\square^M_{(2k),l}$ with $\square^M_{(2k+1),l}$ is covered exactly once, so a certain uniformity is still present in this case. The transpose of this ordering could be used if for some reason it is desired that this property should apply to "horizontal" $2^M \times 2^{M+1}$ rectangles instead of "vertical" $2^{M+1} \times 2^M$ ones.

Figs. 14 and 15 illustrate the degradation of the quality of the standard Barbara and Goldhill images as the pixel length sequence is halved at each step. The figures remain recognizable for sequence lengths as short as 1/32 or 1/64 of the number of pixels in the images.

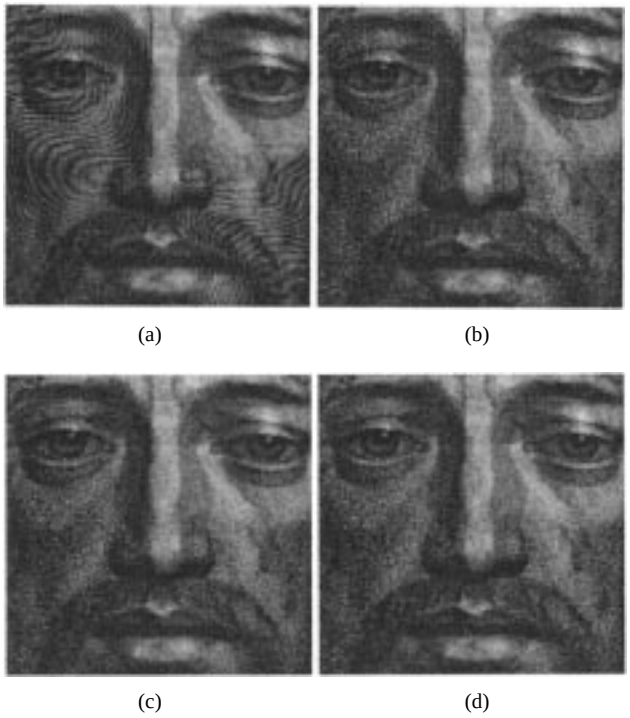


Fig. 9. (a) 256×256 reconstruction using regular holographic sampling. (b) 256×256 reconstruction using "random" sampling.(c) 256×256 reconstruction using $1/d$ smoothing.(d) 256×256 reconstruction using $1/d^2$ smoothing. The RMSE's are 0.107, 0.104, 0.084, and 0.084, respectively.

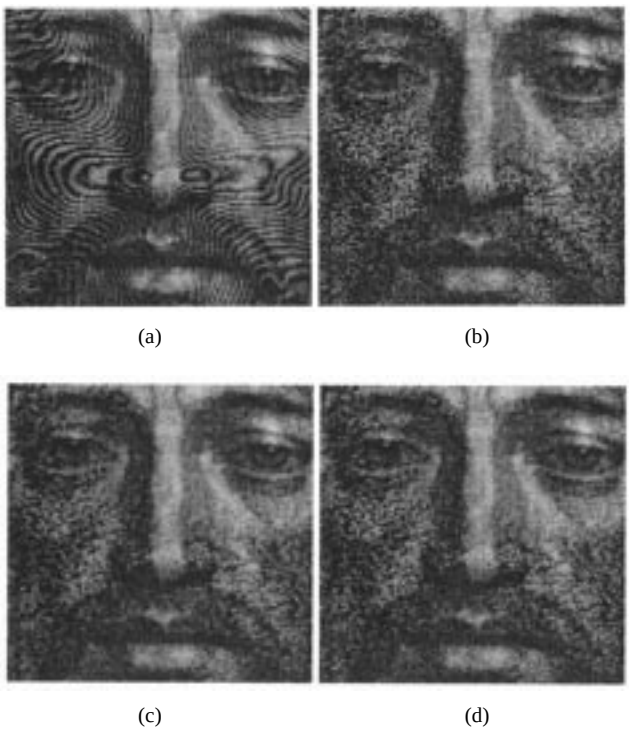
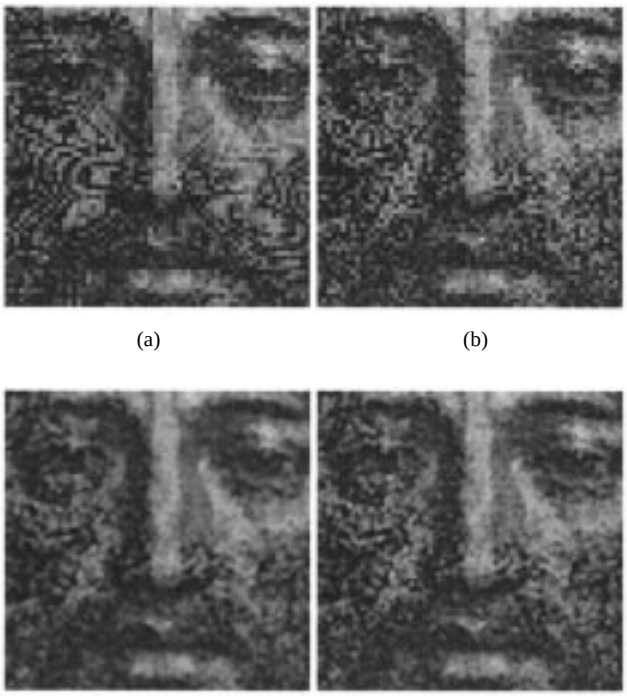


Fig. 10. (a) 128×128 reconstruction using regular holographic sampling.(b) 128×128 reconstruction using "random" sampling. (c) 128×128 reconstruction using $1/d$ smoothing and initial pixel 83 298. (d) 128×128 reconstruction using $1/d^2$ smoothing. The RMSE's are 0.135, 0.134, 0.112, and 0.115, respectively.

III. Holographic Fourier Representations

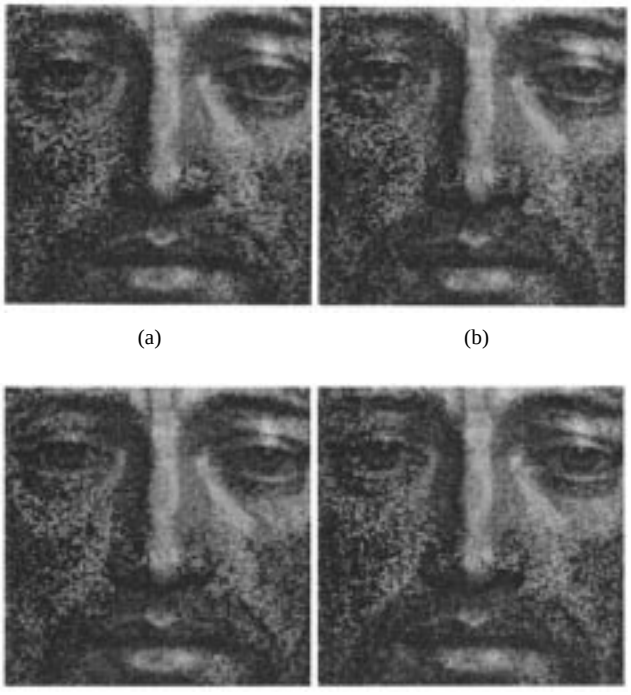
So far we have discussed a method for sequentially sampling the pixels of an image so that any subsequence of the ordered samples will be as uniformly spread over the image as possible. This property yields holographic representations of images as sequences of pixels, and the reconstruction of the original image proceeds by some sort of interpolation between the



(a) (b)

(c) (d)

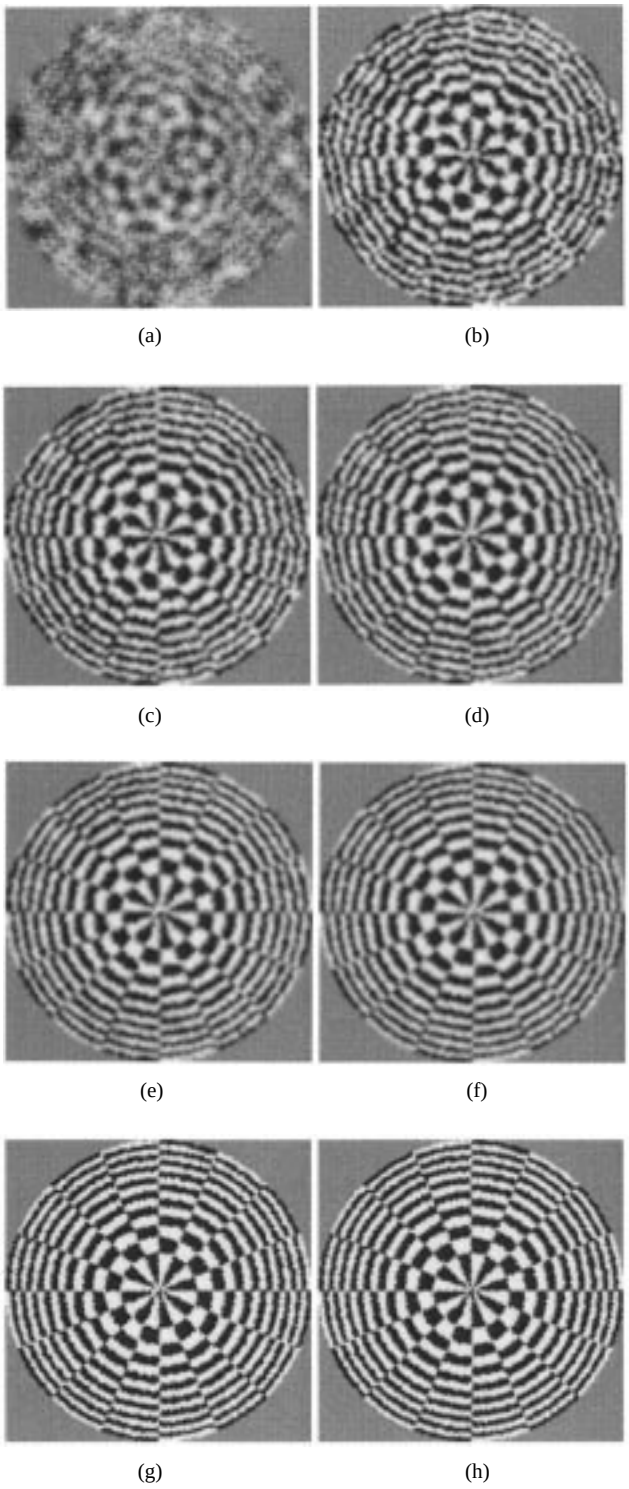
Fig. 11. (a) 64 × 64 reconstruction using regular holographic sampling. (b) 64 × 64 reconstruction using "random" sampling. (c) 64 × 64 reconstruction using $1/d$ smoothing. (d) 64 × 64 reconstruction using $1/d^2$ smoothing. The RMSE's are 0.143, 0.142, 0.115, and 0.120, respectively.



(a) (b)

(c) (d)

Fig. 12. (a) 128 × 128 reconstruction using $1/d$ smoothing and initial pixel 27 561. (b) 128 × 128 reconstruction using $1/d$ smoothing and initial pixel 133 476. (c) 128 × 128 reconstruction using $1/d$ smoothing and initial pixel 193 002. (d) 128 × 128 reconstruction using $1/d$ smoothing and initial pixel 241 785. The RMSE's are 0.113, 0.112, 0.112, and 0.112, which is indicative of how similar the distorted images are regardless of the starting point of the pixel sequence chosen.



(a) (b)

(c) (d)

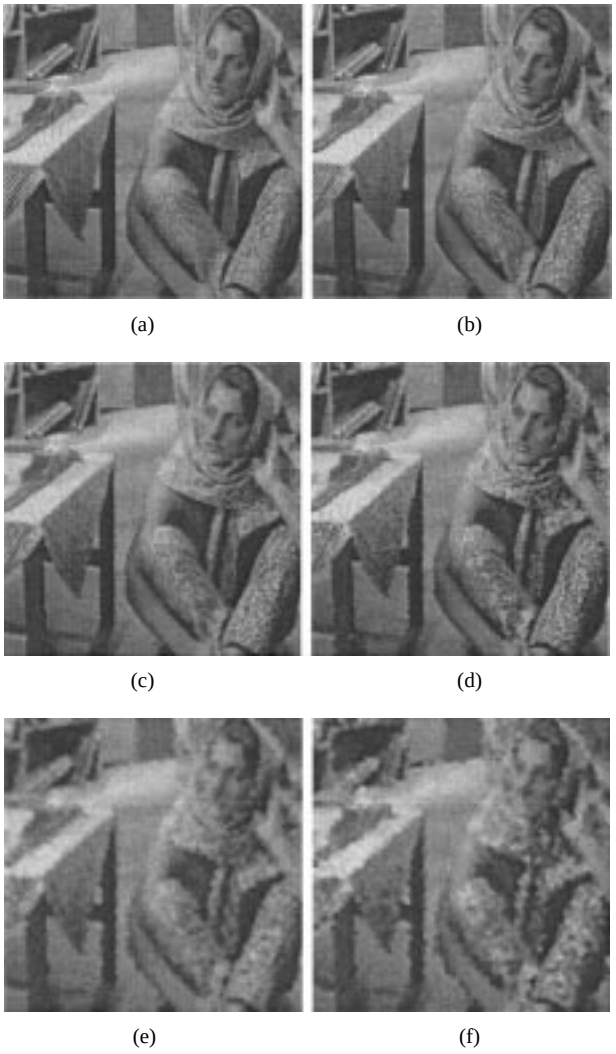
(e) (f)

(g) (h)

Fig. 13. These images are obtained from using pixel sequences of length equal to (a) 1%, (b) 2%, (c) 3%, (d) 4%, (e) 5%, (f) 6%, (g) 7%, (h) 8%, of the number of pixels in the original 512 × 512 image. $1/d$ smoothing is used in all cases. The RMSE's are 0.323,0.248, 0.231, 0.225, 0.218, 0.213, 0.180, and 0.172, respectively.

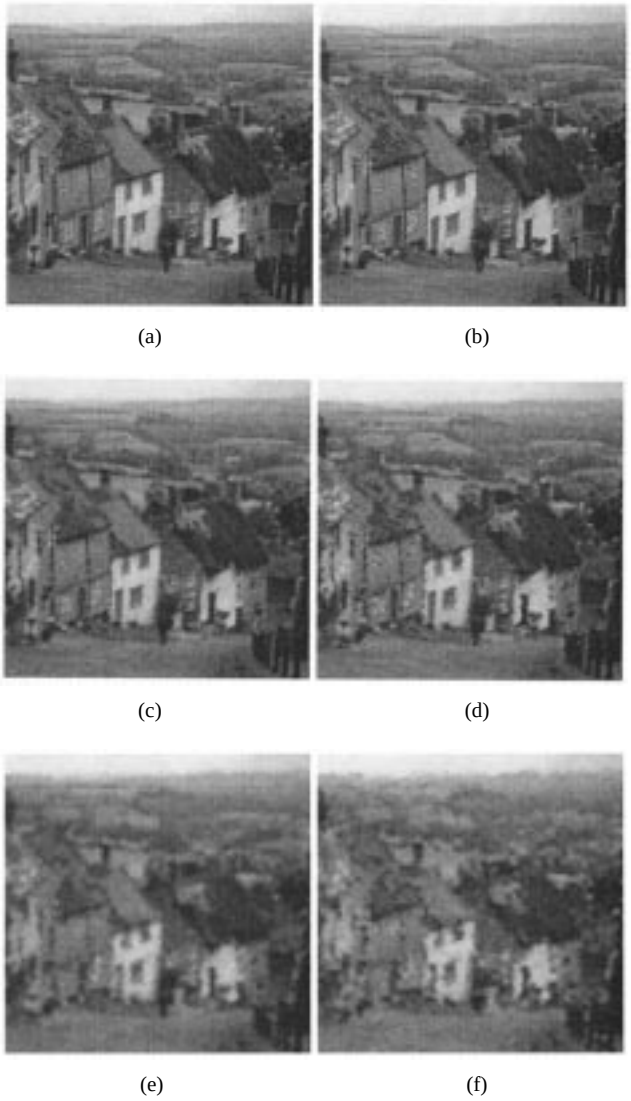
samples available. The next idea that we shall investigate in this paper is based on the Fourier transform.

Suppose $I(x, y)$ is the image to be holographically represented. The idea is to regard this image as the amplitude of the Fourier transform of its holographic representation $H(u, v)$,



(a) (b) (c) (d) (e) (f)

Fig. 14. These images are obtained from using pixel sequences of length equal to (a) $1/2$, (b) $1/4$, (c) $1/8$, (d) $1/16$, (e) $1/32$, and (f) $1/64$, of the number of pixels in the original 512×512 image. The RMSE's are 0.046, 0.061, 0.070, 0.080, 0.086, and 0.097, respectively.



(a) (b) (c) (d) (e) (f)

Fig. 15. These images are obtained from using pixel sequences of length equal to (a) 1/2, (b) 1/4, (c) 1/8, (d) 1/16, (e) 1/32, and (f) 1/64 of the number of pixels in the original 512×512 image. The RMSE's are 0.022, 0.033, 0.043, 0.051, 0.060, and 0.068, respectively.

defined to yield a spatially uncorrelated random phase. Hence,

$$H(u,v) = \mathcal{FT}^{-1}\{I(x,y)e^{jP(x,y)}\}$$

where $P(x,y)$ is a "random phase" image so that $E[P(x,y)P(\tilde{x},\tilde{y})] = 0$ for $(x,y) \neq (\tilde{x},\tilde{y})$ and $P(x,y)$ is a random variable uniformly distributed over $[-\pi,\pi]$. Let us analyze why $H(u,v)$ is expected to be a holographic representation of $I(x,y)$. Here by holographic representation we mean that from an image portion $H^c(u,v)$ cropped from the complex image $H(u,v)$ we can, by 2-D Fourier transformation, get a version of $I(x,y)$ so that the degradation is proportional to the size of $H^c(u,v)$. Clearly, if $H(u,v)$ is available we get back $I(x,y)$ as the amplitude of its 2-D Fourier transform, and the same image recovery process will be applied to cropped parts of $H(u,v)$.

In order to see why we expect this idea to work, let us look at the 1-D version of the above proposed holographic representation method. The discrete 1-D signal $\{I(k)\}$ for $k = 0, 1, \cdots, M-1$ (where M is usually 2^n for some positive integer n) is transformed to

$$H(u) = \sum_{k=0}^{M-1} I(k)e^{j2\pi P(k)} \cdot \frac{1}{\sqrt{M}} e^{j\frac{2\pi}{M}uk}$$
$$= \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} I(k)e^{j\frac{2\pi}{M}[uk+MP(k)]}$$

where $\{P(k)\}$ is a set of independent and identically distributed (i.i.d.) random numbers uniformly distributed over $[0,1]$. We shall represent the process of cropping a portion of $H(u)$ by multiplying it with a window function $W(u)$ so that

$$W(u) = \begin{cases} 1 & \text{for } u \in [a, a+(L-1)] \\ 0 & \text{for } u \notin [a, a+(L-1)] \end{cases}$$

where $a \in \{0, 1, \cdots, M-L\}$ for simplicity.

Now we ask the question: what can be recovered from $H^c(u) = H(u) \cdot W(u)$ by the 1-D Fourier transform? Using

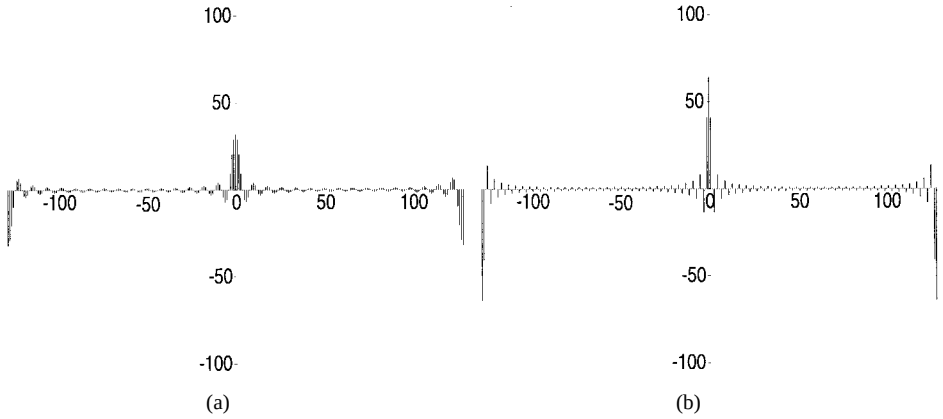


Fig. 16. (a) Coefficients for $M = 128$, $L = 32$. (b) Coefficients for $M = 128$, $L = 64$.

straightforward algebra, we can obtain

$$I_W(r) = \sum_{u=0}^{M-1} H(u)W(u)\frac{1}{\sqrt{M}}e^{-j\frac{2\pi}{M}ur}$$
$$= \frac{1}{M}e^{-j\frac{2\pi}{M}(a+\frac{L-1}{2})r} \times \sum_{k=0}^{M-1} I(k)e^{j\frac{2\pi}{M}[MP(k)+(a+\frac{L-1}{2})k]}\frac{\sin\frac{L\pi}{M}(r-k)}{\sin\frac{\pi}{M}(r-k)},$$

where at the end we have used the identity

$$\sum_{u=0}^{L-1} e^{-j2\pi xu} = e^{-j(L-1)\pi x}\frac{\sin L\pi x}{\sin\pi x}.$$

This equation shows that $I_W(r)$ will be, essentially, a lowpass filtered version of $I(k)e^{j2\pi[P(k)+(a+L-1)k/M]}$. If the random phase factors were not present here the $I_W(r)$ would be simply a bandpass filtered version of $I(k)$ and the holographic property would be lost, since the key low-frequency components would usually be missing in this case.

The role of the random phase $P(k)$ added to $I(k)$ when generating $H(u)$ is therefore clear: since for any a, the sequences $\{\tilde{P}_a(k)\} = \{[P(k) + \frac{k}{m}(a + \frac{L-1}{2})] \bmod 1\}$ are i.i.d. uniformly over $[0,1]$ due to the fact that the $P(k)$ are uniformly i.i.d. there, all portions of $H(u)$ of length L will yield essentially the "same" type information on the whole sequence $I(k)$.

Explicitly, we have

$$I_W(r) = e^{-j\frac{2\pi}{M}(a+L-1)r}\frac{1}{M}\sum_{k=0}^{M-1} I(k)e^{j2\pi\tilde{P}_a(k)}\frac{\sin\frac{L\pi}{M}(r-k)}{\sin\frac{\pi}{M}(r-k)}$$

and we should consider the behavior of the coefficients

$$\phi_L(r-k) = \frac{1}{M}\frac{\sin\frac{L\pi}{M}(r-k)}{\sin\frac{\pi}{M}(r-k)}$$

for $k = 0, \pm1, \pm2, \cdots$, to see what type of an average of $I(k)$ the value of $I_W(r)$ is. First of all we see that when $L = M$ we have

$$\phi_L(r-k) = \begin{cases}(-1)^{(M-1)[(r-k)/M]}, & \text{if } r \equiv k \bmod M\\ 0, & \text{if } r \not\equiv k \bmod M\end{cases}$$

so that $I_W(r) = I(r)e^{j2\pi P(r)}$ as expected. Fig. 16 shows the function $M\phi_L(x)$ for $M = 128$ and $L = 32, 64$, for $x \in [-128, 128]$. As L decreases from M we see from this figure that $I_W(r)$ will be a local average of $I(k)$, involving k's around r with various random phase factors affecting the averaging.

Suppose $I(k) = I_0$ is constant (for all k, or a suitably large neighborhood of r). Then we have

$$I_W(r) = I_0\sum_{k=0}^{M-1} e^{j2\pi\tilde{P}_a(k-r)}\phi_L(r-k)$$

and

$$|I_W(r)|^2 = I_0^2\Bigg\{\sum_{k=0}^{M-1}\phi_L^2(r-k) + \sum_{k\neq l}\sum_{k,l=0}^{M-1} e^{j2\pi[\tilde{P}_a(k-r)-\tilde{P}_a(l-r)]} \times \phi_L(r-k)\phi_L(r-l)\Bigg\}.$$

The question we have to address is: how good an estimate is $|I_W(r)|$ for the value I_0. To answer this question we rely on the following result, whose proof is given in Appendix C.

Lemma: Let $\theta_0, \theta_1, \cdots, \theta_{M-1}$ be i.i.d. random numbers uniformly chosen from $[0,1]$, and define the random variable

$$V = \sum_{k=0}^{M-1} e^{j2\pi\theta_k}\varphi(k),$$

where $\{\varphi(k)\}$ is a sequence of real numbers (weights). For V we have the following statistics:

$$E(V) = 0, \quad E(|V|^2) = \sum_{k=0}^{M-1}\varphi^2(k),$$

and

$$\sigma(|V|^2) = \sqrt{\sum_{k\neq l}\sum_{k,l=0}^{M-1}\varphi^2(k)\varphi^2(l)}.$$

From the lemma it follows that

$$E[|I_W(r)|^2] = I_0^2 \sum_{k=0}^{M-1} \phi_L^2(r-k)$$

and

$$\sigma[|I_W(r)|^2] = I_0^2 \sqrt{\sum_{k\neq l} \sum_{k,l=0}^{M-1} \phi_L^2(r-k)\phi_L^2(r-l)}.$$

If $L = M$ we get $E[|I_W(r)|^2] = I_0^2$, $\sigma[|I_W(r)|^2] = 0$, and if $L < M$, we get

$$E[|I_W(r)|^2] = \frac{L}{M} I_0^2.$$

If $L \leq (M+1)/2$,

$$\sigma[|I_W(r)|^2] = \frac{L}{M}\sqrt{1 - \frac{2L^2+1}{3LM}} I_0^2,$$

and if $1 \ll L \leq (M+1)/2$ we can write

$$\sigma[|I_W(r)|^2] \approx \frac{L}{M}\sqrt{1 - \frac{2L}{3M}} I_0^2,$$

while if $(M+1)/2 \leq L \leq M$, we have

$$\sigma[|I_W(r)|^2] = \frac{1}{M}\sqrt{\frac{(M-L)(M^2-5ML+10L^2-1)}{3M}} I_0^2.$$

Therefore, we have that for L small in comparison with M, $(M/L)^{1/2}|I_W(r)|$ is a relatively noisy estimate of I_0 with standard deviation proportional to I_0, as in an exponential distribution. However, we have the possibility to further improve our estimators by local spatial averaging, and it is exactly this factor that intervenes crucially in image representation and recovery.

So far we have assumed that $H^c(u) = H(u) \cdot W(u)$ is used to recover the holographically encoded signal. However we might have also considered recovering the signal $\{I(k)\}$ from $H(u)$ for $u \in [a, a+(L-1)]$ by computing the 1-D Fourier transform of the sequence $\{H(a), H(a+1), \cdots, H(a+(L-1))\}$ directly. In this case we can use the so-called "packing theorem" for Fourier transforms [1, p. 369]. If

$$\mathcal{FT}\{H(a)\,H(a+1)\,\cdots\,H(a+(L-1))\} = I_T(r)|_{r=0,1,\cdots,L-1}$$

and

$$\mathcal{FT}\{H(a)H(a+1)\cdots H(a+(L-1))\underbrace{0\;0\;\cdots\;0}_{M-L}\} = I_{WS}(\nu)\,|_{\nu=0,1,\cdots,M}$$

then we have

$$I_{WS}(\nu) = \sqrt{\frac{L}{M}} I_T\left(\frac{\nu L}{M}\right)$$

for $\nu = 0, L/M, 2L/M, \ldots, (L-1)L/M$. But we know that

$$I_{WS}(\nu) = I_W(\nu) \cdot e^{-j\frac{2\pi}{M}a\nu}$$

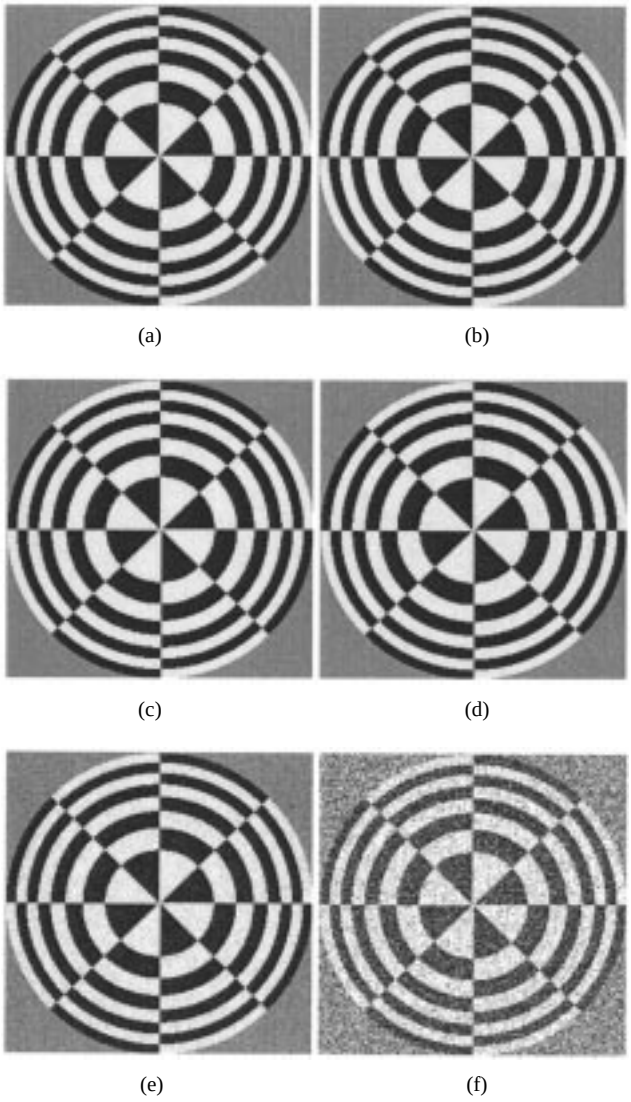


Fig. 17. Effects of quantization in the holographic representation domain. (a) Original image. (b) Image recovered with double precision. (c)–(f) Images recovered using 8 + 8, 6 + 6, 4 + 4, and 2 + 2 b/pixel, respectively. The RMSE's of (b)–(f) are 0.003, 0.005, 0.016, 0.060, and 0.231.

since $I_{WS}(\nu)$ is a (cyclic) shift of $H^c(u) = H(u) \cdot W(u)$ by a spaces to the right. Therefore we have that

$$I_T\left(\nu\frac{L}{M}\right) = \sqrt{\frac{M}{L}} I_W(\nu) e^{-j\frac{2\pi}{M}a\nu}.$$

Hence

$$I_T(\tilde{r}) = \sqrt{\frac{M}{L}} I_W\left(\tilde{r}\frac{M}{L}\right) e^{-j\frac{2\pi}{L}a\tilde{r}}$$

and thus

$$|I_T(\tilde{r})| = \sqrt{\frac{M}{L}}\left|I_W\left(\tilde{r}\frac{M}{L}\right)\right| \quad \text{for } \tilde{r} = 0, 1, 2, \cdots, (L-1).$$

Hence, $|I_T(\tilde{r})|$ is a subsampling of $(M/L)^{1/2}|I_W(r)|$ at $r = 0, M/L, 2M/L, \ldots, (L-1)M/L$. Since $(M/L)^{1/2}\,|I_W(r)|$ is an estimator of I_0 (in case of constant $\{I(k)\}$), we have that $|I_T(\tilde{r})|$ is readily an estimate of I_0 for $\tilde{r} = 0, 1, 2, \cdots, (L-1)$. Therefore, we can proceed either way: consider that the

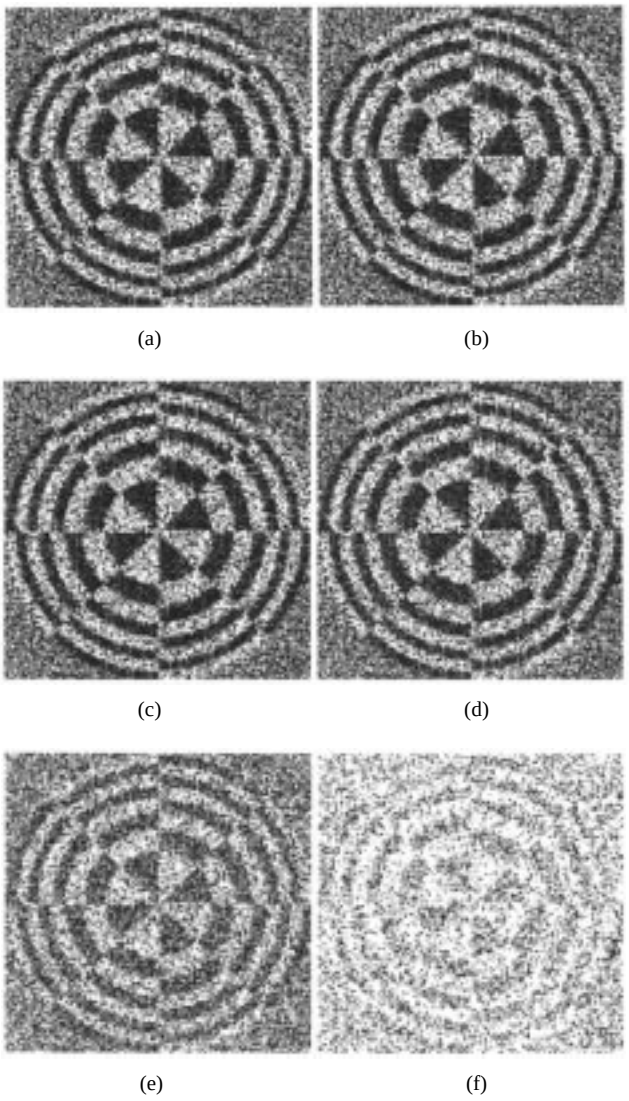


Fig. 18. Recovery of image from an arbitrary 128 × 128 portion of $H(u,v)$ using varying precision in the holographic representation. (a) Double precision. (b)–(f) 8 + 8, 6 + 6, 4 + 4, 2 + 2, and 1 + 1 b/pixel, respectively. The RMSE's are 0.289, 0.289, 0.289, 0.292, 0.338, and 0.465.

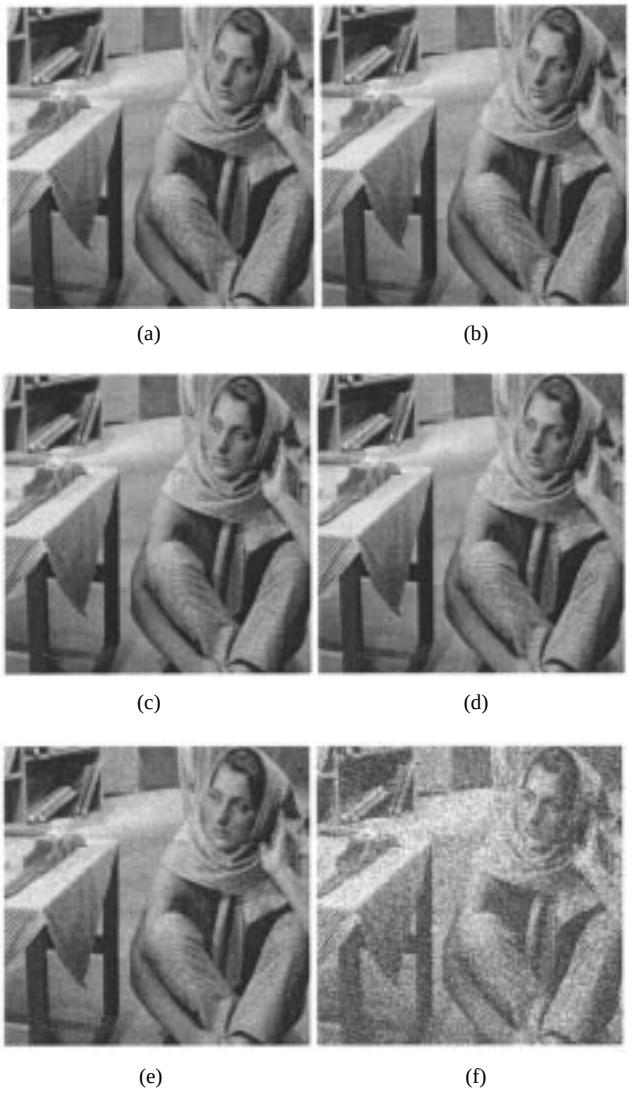


Fig. 19. Effects of quantization in the holographic representation domain. (a) Original Barbara image. (b) Image recovered with double precision. (c)–(f) Images recovered using 8 + 8, 6 + 6, 4 + 4, and 2 + 2 b/pixel, respectively. The RMSE's of (b)–(f) are 0.003, 0.004, 0.016, 0.061, and 0.234.

data provided by $\{H(u)\}$ for $u \in [a, a+(L-1)]$ is a sequence of length L and compute the magnitude of its Fourier transform or pad $H(u)$ with the appropriate number of zeros (to get a sequence of length M) and compute the magnitude of its Fourier transform and then multiply the resulting sequence by $(M/L)^{1/2}$ to get an estimate of $I(k)$. The second approach has the apparent advantage of automatically generating the interpolation (from the sparse data) for full-sized image reconstruction, at the expense of doing larger size transform computations; however, as we shall see, the interpolation provided by this method is not visually superior to pixel replication (see Fig. 20).

So far we have seen that $H(u,v)$, the complex image associated with $I(x,y)$ via the inverse Fourier transform $\mathcal{FT}^{-1}\{I(x,y)e^{jP(x,y)}\}$, is a holographic representation of $I(x,y)$. For simplicity, we have made the arguments to prove this in the 1-D case; however they extend in a straightforward manner to 2-D images. As opposed to the holographic sampling idea discussed in the previous section where a 1-D stream of samples is generated, this representation maps an image matrix $I(x,y)$ encoded usually by 8 b/pixel with a pair of matrices, the real and imaginary parts of $H(u,v)$, that must be quantized with sufficient precision to yield "good quality" glimpses of $I(x,y)$ from arbitrary portions of $H(u,v)$ and full recovery of $I(x,y)$ from the complete $H(u,v)$. It is interesting to note that the fact that the amplitude information of a random phase image is distributed "evenly" in the Fourier domain was previously pointed out in the context of optical holography and SAR imaging (see, e.g., [10], and references therein); however this idea was never proposed as an effective way to achieve distributed image representation.

We have first tested this idea of holographic encoding of $I(x,y)$ using the computer's double precision floating point representation for the real and imaginary parts of $H(u,v)$. For this case the precision of the recovery of $I(x,y)$ from the entire $H(u,v)$ via the Fourier transform was down to the least significant bit of the 8-b representation of $I(x,y)$. A 256 × 256

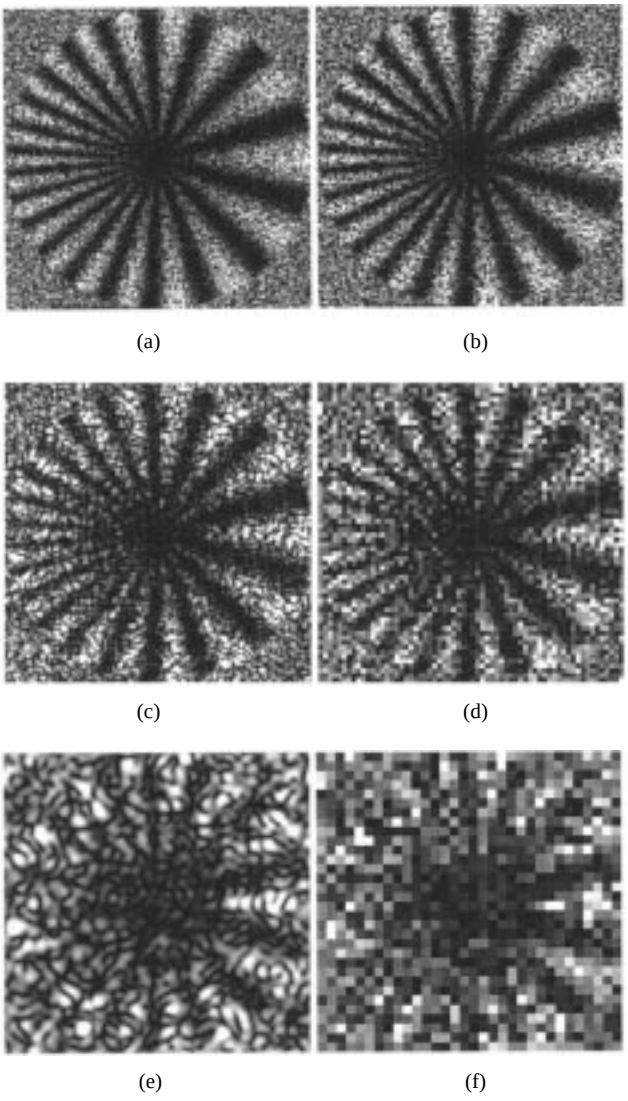


Fig. 20. (a), (c), (e) Comparison of image recovery by zero padding and (b), (d), (f) by using Fourier transform directly and expanding the result by pixel replication. (a)–(b) 128 × 128 portions. (c)–(d) 64 × 64 portions. (e)–(f) 32 × 32 portions. The RMSE's of (a)–(f) are 0.201, 0.203, 0.210, 0.224, 0.233, and 0.264.

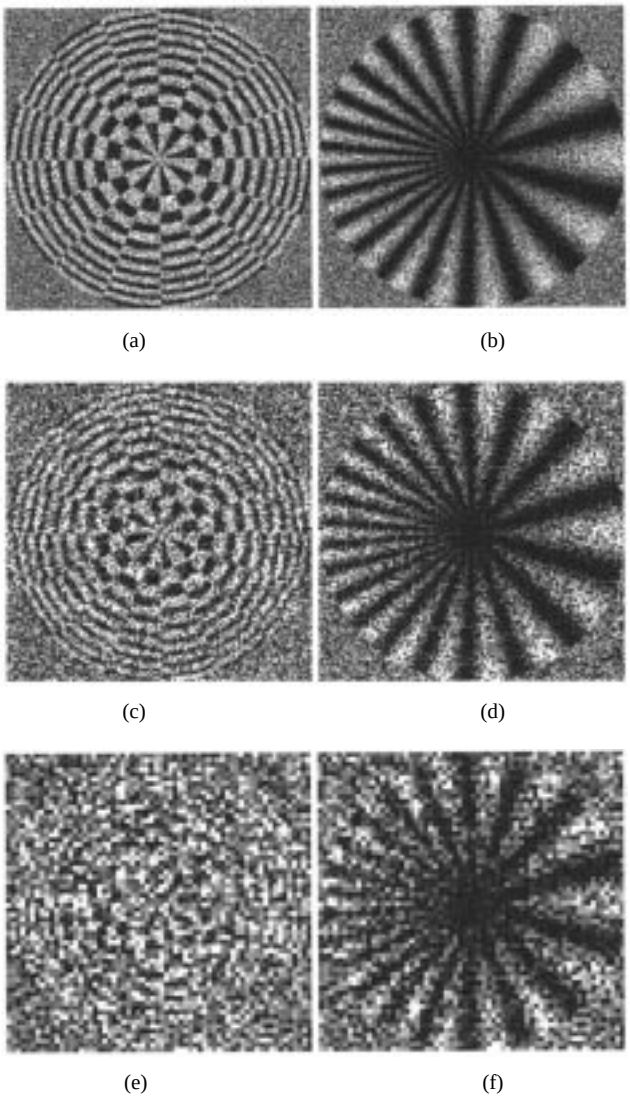


Fig. 21. Reconstruction of the 512 × 512 synthetic images of Fig. 6 from arbitrary portions of $H(u,v)$ of various sizes. (a)–(b) 256 × 256. (c)–(d) 128 × 128. (e)–(f) 64 × 64. The RMSE's of (a)–(f) are 0.288, 0.196, 0.337, 0.204, 0.401, and 0.226.

image with three gray levels (27–127–227) [see Fig. 17(a)] was recovered as having six gray levels (26, 27), (126, 127), (226, 227). When a simple uniform quantization was used to represent $H(u,v)$ with a total of $4+4$ b/pixel (4 b for each the real and imaginary parts of $H(u,v)$), the variance increased to about 20 levels about each of these three gray levels. With 6 + 6 b/pixel the variance was about ten levels and the resulting reconstruction of $I(x,y)$ was visually almost indistinguishable from the original. At 8 + 8 b/pixel the variance was about four levels and the reconstruction looked perfect. (Only when histogram equalization was done could one see the variability in each region having "constant" gray levels.) At each of these quantization levels we also computed the entropy of the resulting histograms for the images $\Re H(u,v)$ and $\Im H(u,v)$. The histograms had a Gaussian-like appearance and the entropies were, for example, 5 b (for the 6-b representations), 7 b (for the 8-b representations), 9 b (for the 10-b representations), and 15 b (for a 20-b representations). In view of these and other experimental findings we concluded that, with optimal quantization in the (u,v) domain, the $\Re H(u,v)$ and $\Im H(u,v)$ images can in fact be as reliably represented with a total of 8 b/pixel as their straightforward uniform quantization to 8 + 8 b.

Fig. 17(a)–(f) shows an original 256 × 256 three-gray-level image and its recovery from the entire (256 × 256) $H(u,v)$ representation when the pixels of $\Re H(u,v)$ and $\Im H(u,v)$ are each encoded with double precision of the machine (Sun Sparc 20), or quantized to 8 + 8, 6 + 6, 4 + 4, and 2 + 2 b/pixel. Fig. 18(a)–(f) shows the recovery of the same image from an arbitrary 128 × 128 portion of $H(u,v)$ using double precision, or 8 + 8, 6 + 6, 4 + 4, 2 + 2, and 1 + 1 b pixel. Fig. 19(a)–(f) shows the 512 × 512 Barbara image using double precision, and then 8 + 8, 6 + 6, 4 + 4, and 2 + 2 b/pixel.

All subsequent experimental results are presented using the straightforward quantization of the real and imaginary parts if $H(u,v)$ with 8 b each.

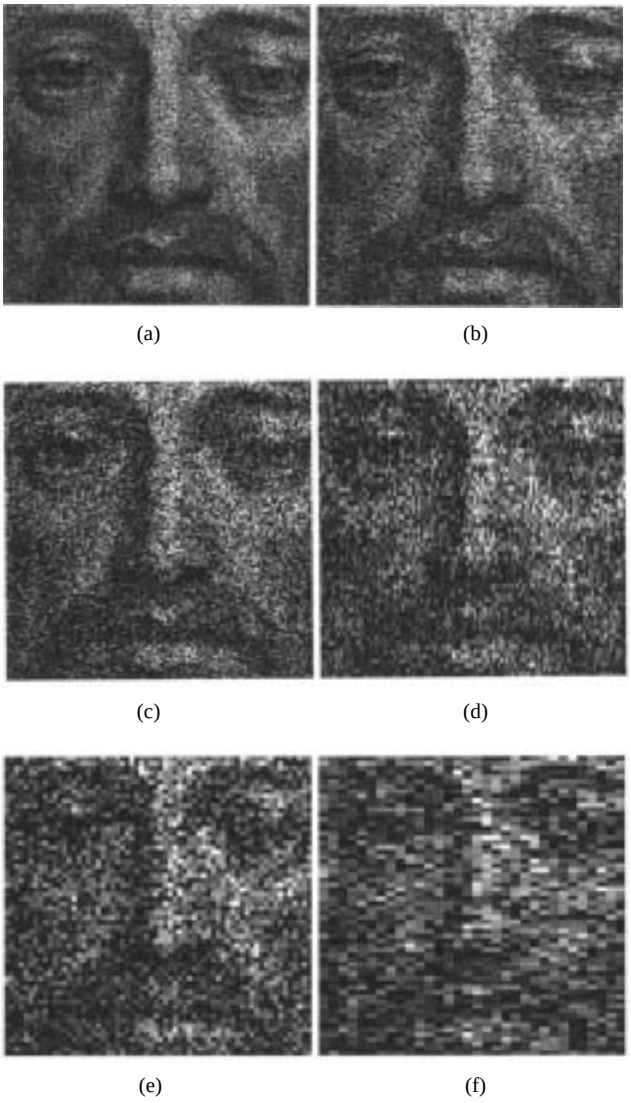


Fig. 22. Reconstruction of the scanned engraving of Fig. 6(e) from arbitrary portions of $H(u,v)$ of various sizes. (a) 256 × 256. (b) 128 × 256. (c) 128 × 128. (d) 128 × 64. (e) 64 × 64. (f) 32 × 64. The RMSE's are 0.178, 0.184, 0.187, 0.190, 0.191, and 0.194, respectively.

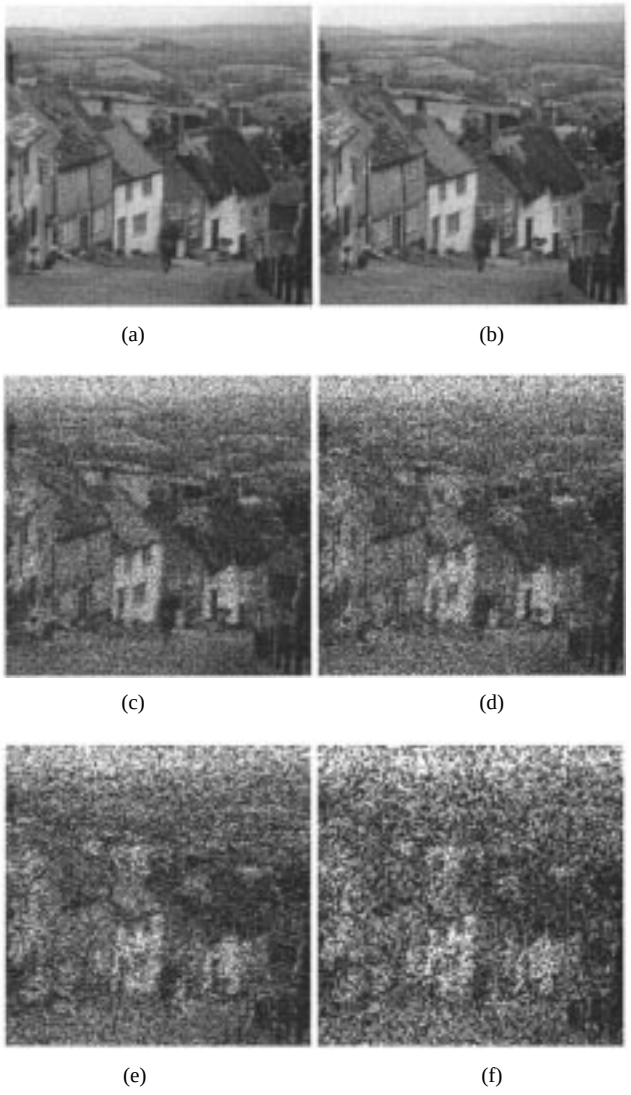


Fig. 23. Reconstruction of the Goldhill image from arbitrary portions of $H(u,v)$ of various sizes: (a) Original image. (b) 512 × 512. (c) 256 × 512. (d) 256 × 256. (e) 128 × 256. (f) 128 × 128. The RMSE's of (b)–(f) are 0.003, 0.196, 0.211, 0.216, and 0.220, respectively.

Fig. 20 compares the recovery of the 256 × 256 test image shown in Fig. 6(b) from portions of $H(u,v)$ of various sizes using either zero padding of $H^c(u,v)$ to an array of original size followed by taking its Fourier transform or the direct application of the Fourier transform followed by expansion to 256 × 256 via straightforward pixel replication. These results show that recovery via direct application of the Fourier transform followed by pixel replication is quite good and the interpolations implicitly carried out via the zero padding approach clearly do not justify the added computational effort. Subsequent experiments with 512 × 512 images hence always used image recovery by direct Fourier transform and pixel replication.

Fig. 21(a)–(f) shows reconstruction of the synthetic images of Fig. 6(c) and (d) from arbitrary portions of $H(u,v)$ of sizes 256 × 256 (1/4 of original), 128 × 128 (1/16 of original), and 64 × 64 (1/64 of original).

Fig. 22(a)–(f) shows reconstruction of the scanned engraving original of Fig. 6(e) from arbitrarily selected portions of $H(u,v)$ of sizes 256 × 256, 128 × 256, 128 × 128, 128 × 64, 64 × 64, and 64 × 32. Fig. 23(a)–(f) shows reconstructions of the 512 × 512 Goldhill image from all of $H(u,v)$, and arbitrary portions of sizes 256 × 512, 256 × 256, 128 × 256, and 128 × 128. These sequences of images clearly exhibit the progressive improvement that is obtained as bigger portions of the holographic representation become available.

IV. Conclusion

There exist many well-established types of hierarchical representations of images; however, it seems to us that the holographic representation as discussed herein and others that will probably be developed in the future do have some advantages in a "distributed" world.

We have presented here two ideas for holographic image representations. It is possible to recover a blurred version of the

A	B	C	D
E	4	7	H
I	10	13	L
M	N	O	P

A	B	C	D
E	4	7	H
I	13	10	L
M	N	O	P

A	B	C	D
E	4	10	H
I	13	7	L
M	N	O	P

Fig. 24. Arrangements with {4, 7, 10, 13} in the center 2 × 2 square.

whole image from an arbitrary portion of these representations, the degree of blurriness decreasing as the size of the portion increases. Holographic representations enable progressive refinement in image communication tasks, with no restrictions whatsoever on the order in which the data are received.

As we have seen from the experimental results, the pseudorandom holographic sampling representation yields better image quality, especially when small portions of the data are available. For the Fourier domain representation however, the recovery of blurred originals is possible even when the location of the cropped portion is unavailable. Both methods are, in our opinion, viable if the aim is progressive refinement for preview purposes and the full image recovery will be carried out before actually making further use of the images. We point out that holographic representations are by no means compression schemes: their use can be in distributing the information evenly among many locations in a computer network or in situations when the image is to be broken into packets of data that may arrive to a user in a random order. For these situations the user can have a low resolution preview of the image from whatever piece of information is first available. Such holographic representation could also be used for sharing image data in a way that will make it necessary for all the parties involved (that own pieces of the holographic representation) to agree and collaborate in order to recover the highest quality version of the image.

One might be tempted to regard the Fourier domain representation discussed in Section III as a digitally implemented hologram. However, it is clear from the literature that computer-generated holograms are, in spite of the use of the Fourier transform in their generation (see e.g. [11] and the references therein), quite different from the representations introduced in Section III.

Appendix A
Proof of Assertion in Section II

It is impossible for every sequence of length four to cover all nine 2 × 2 squares in Fig. 1. Indeed, consider the four corner squares ABEF, CDGH, IJMN, and KLOP. One each of {1, 2, 3, 4} must appear in exactly one of those four squares. Now 5 must appear in the same square as 1 so that 2-3-4-5 covers all four squares. Continuing in this manner, each of {ABEF, CDGH, IJMN, KLOP}, is seen to hold numbers that are congruent to each other modulo 4. Now considering FGJK, the only way to take four numbers in $[1, 16]$, all of which are unequal modulo 4, such that each sequence of length 4 is represented, is to choose {4, 7, 10, 13}. There cannot be a gap of length 5 between nearest integers in this set, or in any of the 2 × 2 squares. Also, there cannot be a gap of length 4 in FGJK as each element is different modulo 4.

There are three ways to place the numbers {4, 7, 10, 13} in {F, G, J, K}, counting rotations and reflections as equivalent, as shown in Fig. 24. In the first two of these patterns, consider JKNO. In order for this square to cover each of the sequences 1-2-3-4, 2-3-4-5, 3-4-5-6, 6-7-8-9, $\{\mathrm{N}, \mathrm{O}\}$ must contain an element from each of $\{1, 2\}$, $\{2, 5\}$, $\{5, 6\}$, and $\{6, 9\}$, as neither N nor O may be 3, 4, 7, or 8. The only this can happen is if $\{\mathrm{N}, \mathrm{O}\} = \{2, 6\}$, but this is impossible as O must be congruent to 1 mod 4. In the third pattern, consider GHKL. In order for this square to cover both of 1-2-3-4 and 3-4-5-6, $\{\mathrm{H}, \mathrm{L}\}$ must contain an element from each of $\{2, 3\}$, and $\{3, 6\}$, as neither H nor L may be 1, 4, or 5. This could only happen if $\mathrm{L} = 3$, as H cannot be simultaneously 2 and 6. Now consider JKNO. In order for this square to cover both of 1-2-3-4 and 2-3-4-5, $\{\mathrm{N}, \mathrm{O}\}$ must contain an element from each of $\{1, 3\}$, and $\{3, 5\}$, as neither N nor O may be 2 or 4. This could only happen if $\mathrm{O} = 3$, as N cannot simultaneously be 1 and 5. However, both L and O cannot be 3. Therefore in all three cases it is impossible for every 2 × 2 square to contain at least one element from every sequence of length 4.

Appendix B
Proof of Theorem 1

Proof: This result will be proved by showing that all the integers contained in any square $\square_{kl}^{M}$ are congruent modulo 4^{N-M}. Equivalently, any two integers congruent to each other modulo 4^{N-M} must lie in the same square $\square_{kl}^{M}$. Suppose $n_1 \equiv n_2 \,(\mathrm{mod}\, 4^{N-M})$. Then $n_1/4^{N-M-1} \equiv n_2/4^{N-M-1} \,(\mathrm{mod}\, 4)$, and by (2), $f(n_1, a) = f(n_2, a)$ for $a = 0, \cdots, N-M-1$. Then by (3), $r(n_1, a) = r(n_2, a)$ and $c(n_1, a) = c(n_2, a)$ for $a = 0, \cdots, N-M-1$, and by (4), the first $N-M-1$ binary digits of i_1 and i_2 are identical, as are those for j_1 and j_2, where (i_1, j_1) and (i_2, j_2) are the pixels at which n_1 and n_2 are located. This implies that (i_1, j_1) and (i_2, j_2) are both located in $\square_{kl}^{M}$, where $k = \sum_{a=0}^{N-M-1} 2^{N-M-1-a} r(n, a)$, $l = \sum_{a=0}^{N-M-1} 2^{N-M-1-a} c(n, a)$, and n is either n_1 or n_2.

Thus, for each k and l, $\square_{kl}^{M}$ contains 4^M integers in $[0, 4^N - 1]$ that are congruent to each other. Since there are 4^{N-M} such squares, there is a one-to-one correspondence between these squares and the sets of 4^M integers in $[0, 4^N - 1]$ which are congruent to each other modulo 4^{N-M}. Therefore every integer sequence of length 4^{N-M} (including wrap-around sequences) has exactly one element in each $\square_{kl}^{M}$. □

Appendix C
Proof of Lemma in Section III

Proof: Using straightforward algebra, we obtain

$$E(|V|^2) = E \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} e^{j2\pi(\theta_k - \theta_l)} \varphi(k)\varphi(l) = \sum_{k=0}^{M-1} \varphi^2(k)$$

and

$$E[(|V|^2)^2] = E\left\{ \left[\sum_{k=0}^{M-1} \varphi^2(k) + \sum_{k \neq l} \sum_{k,l=0}^{M-1} e^{j2\pi(\theta_k - \theta_l)} \varphi(k)\varphi(l) \right] \right.$$

$$\cdot \left[\sum_{k'=0}^{M-1} \varphi^2(k') + \sum_{k' \neq l'} \sum_{k',l'=0}^{M-1} e^{j2\pi(\theta_{k'} - \theta_{l'})} \varphi(k')\varphi(l') \right] \Bigg\}$$

$$= \left[\sum_{k=0}^{M-1} \varphi^2(k) \right]^2 + \sum_{k \neq l} \sum_{k,l=0}^{M-1} \varphi^2(k)\varphi^2(l).$$

Thus

$$\sigma^2(|V|^2) = \mathrm{Var}(|V|^2) = E[(|V|^2)^2] - [E(|V|^2)]^2$$
$$= \sum_{k \neq l} \sum_{k,l=0}^{M-1} \varphi^2(k)\varphi^2(l). \qquad \square$$

ACKNOWLEDGMENT

The authors thank J. Lagarias for bringing his recent work with N. Jayant on storing signals in 2-D arrays to their attention while this paper was being written, and T. Marzetta for bringing [10] to their attention following a lecture on holographic image representation based on [2]. The editor and the referees are also acknowledged for several helpful suggestions.

REFERENCES

[1] R. N. Bracewell, *The Fourier Transform and Its Applications*, 2nd ed. New York: McGraw-Hill, 1978.
[2] A. M. Bruckstein, R. J. Holt, and A. N. Netravali, "Holographic image representations," Tech. Rep. BL0113380-961010-09TM, Bell Labs., Murray Hill, NJ, Oct. 1996.
[3] A. M. Bruckstein and T. J. Richardson, "A holographic transform domain image watermarking method," Tech. Rep. BL011338-961205-11TM, Bell Labs. Murray Hill, NJ, Dec. 1996.
[4] R. L. Cook, "Stochastic sampling in computer graphics," *ACM Trans. Graph.*, vol. 5, pp. 51–72, 1986.
[5] F. C. Crow, "The aliasing problem in computer-generated shaded images," *Commun. ACM*, vol. 20, pp. 799–805, 1977.
[6] A. Z. Dippe and E. H. Wold, "Antialiasing through stochastic sampling," *Comput. Graph.*, vol. 19, pp. 69–78, 1985.
[7] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Trans. Image Processing*, vol. 6, pp. 1305–1315, 1997.
[8] J. Lagarias, "Well-spaced labelings of points in rectangular grids," AT&T Tech. Memo. 11218-960130.01TM, 1996.
[9] D. P. Mitchell, "Generating antialiased images at low sample densities," *Comput. Graph.*, vol. 21, pp. 65–72, 1987.
[10] D. C. Munson and J. L. C. Sanz, "Image reconstruction from frequency-offset Fourier data," *Proc. IEEE*, vol. 72, pp. 661–669, 1984.
[11] S. Trester, "Computer simulated holography and computer generated holograms," *Amer. J. Phys.*, vol. 64, pp. 472–478, 1996.

Alfred M. Bruckstein received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion—Israel Institute of Technology, Haifa, in 1977 and 1980, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1984.

Since October 1984, he has been with the Technion, where he is Professor of Computer Science. He is a frequent visitor to Bell Laboratories, Murray Hill, NJ. His present research interests include image analysis, image processing pattern recognition, ants, robotics, and computer graphics. He also has done work in estimation theory, signal processing, algorithmic aspects of inverse scattering, point processes and mathematical models in neurophysiology.

Prof. Bruckstein is a member of SIAM, MAA, and AMS.

Robert J. Holt received the B.S. degree in mathematics from Stanford University, Stanford, CA, in 1982, and the Ph.D. degree in mathematics from the Massachusetts Institute of Technology, Cambridge, MA, in 1986.

He was Assistant Professor of Mathematics at Northwestern University, Evanston, IL, from 1986 to 1988. Since then he has worked at Bell Laboratories, Murray Hill, NJ, first part of AT&T until 1996 and then part of Lucent Technologies thereafter. He is currently in the Video Communications Research Department of the Multimedia Communications Research Laboratory. His research interests include computer vision, pattern recognition, and image processing.

Arun N. Netravali (S'67–SM'78–F'85) received the B.Tech degree from the Indian Institute of Technology, Bombay, India, and the M.S. and Ph.D. degrees from Rice University, Houston, TX, all in electrical engineering. He is also the recipient of an honorary doctorate from the Ecole Polytechnique Federale, Lausanne, Switzerland.

He is Executive Vice President of Lucent Technologies, Bell Labs Research, Murray Hill, NJ. Dr. Netravali has authored more than 135 technical papers and co-authored three books, and he holds more than 60 patents in the areas of computer networks, human interfaces to machines, image processing, and digital TV.

Dr. Netravali is a Fellow of the American Academy of Arts and Sciences and a member of the U.S. Academy of Engineering. Among many awards, he recently received the Japanese C&C Computers and Communications Prize in 1997.