Optical implementation of matching pursuit for image representation

Tien-Hsin Chao and Brian Lau

Jet Propulsion Laboratory

California Institute of California

Pasadena CA 91109

and

William J. Miceli

Office of Naval Research

Arlington VA, 22217

ABSTRACT

We have developed a technique for image analysis, representation, and decomposition. This technique was motivated by Stephane Mallat's matching-pursuit algorithm. We've altered and simplified the mechanics of his algorithm to enable an extremely fast implementation via optical processing. Initial computer simulations show that our algorithm is capable of decomposing and representing a 2-D image as a linear combination of basis images with both high speed and high fidelity.

Key Words: Matching Pursuit; Adaptive Wavelet Transform; Image Representation; Pattern Recognition; optical Processing.

I. introduction

The matching-pursuit algorithm developed by Stephane Mall at [1] expresses any signal as a linear sum of waveforms from a highly redundant dictionary of functions. Because this dictionary is highly redundant, expansions in terms of this basis aren't unique. The waveforms in the dictionary are chosen to best match the signal structures. Because, the matching pursuit is a greedy algorithm, basis elements with very large correlations to the data have the largest effect on the expansions. This tends to spread out noise (and other signal elements differing from the dictionary elements) among several basis elements, diluting its impact on the representation.

Mallat demonstrates this favorable effect on noise with a matching pursuit representation (using a dictionary made with the Gabor family of wavelets) of a sound signal to which Gaussian white noise has been added with a signal to noise ratio of 1 db. The white noise is spread throughout the time-frequency display of the representation and this display retains most of the features of the representation of the noiseless signal.

As a final example, Mallat builds a wavepacket dictionary with the Daubechies 6 quadrature mirror filters. He compares a matching pursuit with the best basis algorithm of Coifman and Wikerhauser that selects an "optimal" orthonormal basis within the wavepacket dictionary. Coifman and Wikerhauser's algorithm finds an orthonormal basis $\{g_{\gamma_n} : n = 1 \ldots N\}$ within the dictionary that minimizes the entropy

$$\sum_{n=1}^{N} |\langle f, g_{\gamma_n} \rangle|^2 \log_2 \left(|\langle f, g_{\gamma_n} \rangle|^2 \right)$$

minimizing over all signal components in the dictionary.

Both algorithms are applied to a sound signal composed of chirps and a variety of waveforms of different time-frequency Idealizations. The time-frequency display of the representation found by the matching pursuit displays clear similarities to that of the original sound signal. The time-frequency display of the representation found by Coifman and Wikerhauser's algorithm shows little resemblance to that of the original signal. The global optimization of the Coifman and Wikerhauser algorithm is not well adapted to the wide range of local structures appearing in the signal. For such signals, a matching pursuit is superior to global optimization of basis.

Before an overview of Mallat's matching pursuit algorithm, we give some basic notation: The space $L^2(R^2)$ is the Hilbert space of complex valued functions such that

$$\|f\|^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |f(x,y)|^2 dxdy < \infty \quad (1)$$

The inner product of f and g in $L^2(R^2)$ is defined by

$$\langle f, g \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)\overline{g}(x,y)dxdy \quad (2)$$

where $\overline{g}$ is the complex conjugate of g.

Basically, a matching pursuit works like this: Dictionary elements with unit norm are chosen, To approximate a function f in L^2 (R^2), the dictionary element g_1 with the greatest amplitude inner product with f *is* found. The orthogonal projection of this dictionary element onto f *is* subtracted from f to give a residue *RI(f), where*

$$R_1(f) = f - \langle f, g_1 \rangle g_1 \quad (3)$$

The algorithm continues by decomposing the. residue, *RI(f)* to get

$$R_2(f) = f - \langle f, g_1 \rangle g_1 - \langle f, g_2 \rangle g_2 \quad (4)$$

and, in general,

$$R_n(f) = f - \sum_{i=1}^{n} \langle f, g_i \rangle g_i \quad (5)$$

As the residues' norm approaches O, the sum of the orthogonal projections comes closer and closer to representing the input function exactly.

In reality, there area number of complications. First, the dictionary will, in practice, be so large that computing the inner product of *f with* every dictionary element is infeasible or impossible. Instead, a subdictionary is selected and inner products with each element in the subdictionary are computed. Newton's method is used to get, from this data, an approximation to the dictionary element with the largest inner product with *f*.

Second, after we find the dictionary element having the largest inner product with *f* Mallat's algorithm uses what he calls a ``backprojection'' to refine the coefficients in the linear sum thus far. This makes the matching pursuit converge faster.

II. A matching pursuit algorithm for image representation

In the interests of speedy optical computation, neither of these complications used by Mallat were utilized. Our problem differs from those Mallat attacks in several ways. We wish to study images, not waveforms. Thus, our targets will not, in general, be well localized in frequency. We wish to recognize certain target shapes in our images, not frequencies. Moreover, our images will be positive functions representing light intensity, not zero-mean waveforms. This makes a wavelet basis or Fourier basis not as well suited to representing our images as are functions more like the targets we wish to recognize. We wish to choose dictionary elements similar to our targets, and dissimilar to our decoys, if any. Here, by similar and dissimilar we mean having, respectively, large and small inner products with our dictionary elements.

Let's assume we are trying to discriminate or recognize targets where we know, to a good approximation, their shapes (for example, printed letters of the alphabet, or airplane silloulettes, or, as in our example, geometrical shapes). We form a dictionary, D, of the targets we might find. Let g(x, y) be one such target. (In our example, it is an isosceles triangle.) We expand our dictionary by including in it, all translations, rotations, and scales of our target set. More precisely, For each scale s, angular orientation θ, and translation (x_0, y_0), we denote $\gamma = (s, \theta, x_0, y_0)$. We expand our dictionary, D, so that, if it includes $g(x,y)$ then it includes all it's scale, orientation, and translation variations:

$$g_\gamma(x,y) = \frac{1}{\sqrt{s}} g\left(\frac{x\cos\theta - y\sin\theta - x_0}{s}, \frac{y\cos\theta - x\sin\theta - y_0}{s} \right) \tag{6}$$

This dictionary is extremely redundant. We wish to represent any image f as a linear sum

$$f = \sum_\gamma a_\gamma g_\gamma \tag{7}$$

Selecting only a finite number of indices in our sum gives an approximation to $f \approx \sum_i a_{\gamma_i} g_{\gamma_i}$. Denote the residue after n iterations as $R,(f) = f - \sum_{i=1}^{n} a_{\gamma_i} g_{\gamma_i}$. We wish to select our indices to approximate f efficiently. As with the matching pursuit, we do so by selecting the g_{γ_i} one at a time, choosing, at the $(n+1)'th$ iteration, that g_γ most closely representing some part of the residue R.(f) *(i.e.* this is a greedy algorithm). The details appear below.

Even if we reduce the size of our dictionary to include only a few scales s and angular orientations O, the expansions we form will still be meaningful. They will give information on the location of target-like pieces in the image, *f,* and, if our dictionary isn't too small, still do a fairly good job of approximating *f.* Since our goal is target recognition

and image analysis rather than approximating images, wc accordingly reduce our dictionary size in the interests of enabling fast, optical, computation of our results. Optically, we will be able to do the calculations we need in parallel, using negligible amounts of time. Mallat uses what he calls a backprojection to refine his approximations at every step (essentially, approximating the residue as a linear sum of the dictionary elements in use so far, and incorporating this sum into his approximation of f). The time to do this is prohibitive compared to the rest of the computation (all done optically). Since the backprojection gains us only a slight increase in accuracy for all this computation, we omit this calculation.

For our test of the method, we used a dictionary composed of only one target type (an isosceles triangle with fixed apex angle) at 36 angular orientations (10 degree increments), and arbitrary position. Our test image contains similar triangles (slightly different in size than our dictionary elements), and, a circle.

Since our dictionary elements are positive functions, choosing the dictionary element with largest inner product with f (as in the matching pursuit) merely selects the largest brightest object in our image as the position, (x_o, y_0), of the chosen dictionary element. We'd like to chose the position and angular orientation of the dictionary elements we use in our sum based upon the resemblance of the dictionary elements to parts of our image. For this purpose, we chose the dictionary elements in our expansions in the following way.

For each scale, angular orientation, and target type in our dictionary, chose the $g_\gamma(x,y) \in D$, with centroid at the origin. Denote these by g_n n = 1,2, ... Our test example has 36 g_n. Now create the functions

$$G_n(x,y) = \left(2g_n(x,y) - g_n\left(\frac{x}{\sqrt{2}}, \frac{y}{\sqrt{2}}\right)\right) \Big/ \|g_n\|^2 \qquad (8)$$

Then

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G_n(x,y)dxdy = 0 \qquad (9)$$

If g_n is a binary function, then the inner product of a binary image against G_n will have maximum value if the image matches g_n exactly. Any scaling or rotation of the target image will reduce this inner product.

To chose the next g_γ in our expansion of the image, we correlate the residue against all the G_n formed as above. These correlations can be done instantly and in parallel using optics.' If G_n gives us the largest (in absolute value) correlation peak, then the corresponding g_N translated to the position of the peak gives us the next g_γ in the expansion. The amplitude of the peak is the coefficient, a_γ of g_γ in the expansion.

At some point, the L^2-norm of the residue will increase after an iteration (unless our dictionary is large enough to represent f exactly). We halt the expansion when the norm of the residue becomes small or when norm of the residue increases (else we might have the norm oscillate. forever without converging to O).

Our results show an amazingly close representation of our original image, especially considering how far from complete is our dictionary and how little the dictionary elements (triangles) resemble one of the objects (a circle). Pieces of our image that resemble our dictionary elements are easily found by the large coefficients in the expansion obtained, illustrating the usefulness of this technique for automatic target recognition.

III. Feasibility demonstration via computer simulation

To evaluate the performance of the matching pursuit image representation algorithm, we have performed a computer simulation. Successful decomposition of an input image into a linear combination of elements from a limited dictionary has been completed. The results are shown in Figure 1. Figure 1 a shows an input image that consists many triangles with varied orientations and positions and a circle. The progress results in decomposing, this input into a sum of selected dictionary elements are shown in Figures 1 b through 1j. The dictionary elements are triangles that are very close in size and shape to those appearing in the input image. Figures 1 b and 1 c show one of these dictionary elements and the corresponding wavelet-like counterpart, respectively. There are a total of 36 such triangles contained in the dictionary, each one is of the same size with 10 degree increments of orientation. After each iteration of our algorithm, our linear sum (of triangles) contains one more term and the residue left after subtraction of this sum from our input image has smaller norm. Figures 1d through 1 i show the residue after the first, second, seventh, tenth, fourteenth, and the twenty-second iterations. A table of data, as shown in Table I, represents the progress of our algorithm in decomposing our example image as a linear sum of dictionary elements which take the form of isosceles triangles (as shown in figures 1 b and lc). It gives the orientation, spatial address, and coefficient in the linear sum for the dictionary elements used to represent the example image. The original image's amplitude is 255 and its size is 512 x 512 pixels. Location is in pixels (across, down) from the upper left corner. Norm is the L-squared norm of the error in the approximation. We stop at the 22nd iteration because the 23rd increases the norm. Note that the coefficients associated with the first nine dictionary elements chosen are very large, close to the brightness of the input image (i.e. 255). These are the coefficients for the terms representing the triangle in the example image. The rest of the iterations of the algorithm are used to approximate the circle or to refine the approximations to the triangles in the input (since the dictionary triangles are slightly smaller than the input image triangles). One can see that the coefficients associated with these dictionary elements are much smaller than those

associated with the first nine, illustrating the way signal elements not resembling the dictionary elements have their information diluted among several dictionary elements. The last image (Figure 1j) shows the linear sum of dictionary elements representing our input image. Though close to the input, the representation could be further improved by utilizing a larger set of redundant (or even a complete) dictionary.

III. Optical Implementation

The matching pursuit algorithm for image representation described in Section 2 requires iterative use of several basic operations: 1) 2-D adaptive wavelet transforms; 2) peak detection for those transforms; and 3) intensity modulations of the dictionary element images and their subsequent subtractions from the input. Optical processing systems can perform all three operations, greatly reducing processing time over computer implementations.

We have developed an innovative optical processing architecture, as shown in Figure 2, that is capable of performing multichannel parallel wavelet transforms and peak detections, This system consists of an spatial light modulator, e-beam replication optics, a Fourier transform lens, a holographic wavelet filter array, an inverse Fourier transform lenslet array, and an array of interconnected 2-D peak-detection optoelectronic photodetector chips. In order to identify the dictionary element wavelet that would produce the maximum wavelet transform peak against the input image, an exhaustive search has to be conducted. First, all wavelet elements in the dictionary are arranged in a 2-D Fourier hologram array of filters. Second, the input image is replicated by e-beam optics and each copy is then Fourier transformed to form a 2-D Fourier spectrum array which is passed through the array of wavelet filters. A lenslet array performs inverse Fourier transforms of the outputs from each of the wavelet filters, producing the corresponding wavelet transforms. These wavelet

transforms are spatially separated each other and input to an array of 2-D peak-detection chips, one chip for each transform. A common thresholding signal is sent to each of these chips though interconnecting bus circuitry. Each peak detector outputs the locations of all pixels in its input image with intensities above the threshold level, The outputs of all of the peak detector chips is simultaneously monitored. The threshold level increases until the brightest of the peaks found by the photodetectors is identified. This peak's value, location, and corresponding wavelet are recorded for use in the subsequent operations.

The dictionary element image, associated with the identified wavelet, is then multiplied by the wavelet transform peak value (i.e. intensity modulation), recentered at the address identified by the peak detection chip, and subtracted from the input image . The remaining residue image is then fed back into the parallel optical wavelet processor for the next iteration. The iterations continue until a zero output (i.e. residue image) is obtained or until an iteration of the process increases the energy of the residue image (this increase in energy represents a decrease in the accuracy of the representation).

Intensity modulation and image subtraction can be implemented through optical processing. However implementing these operations with optical processing instead of digital processing gains a much lower increase in speed than that gained through optical implementations of the computationally expensive wavelet transforms and peak detections.

We have developed a multichannel-correlator-based neocognitron-type optical neural network and have demonstrated its applicability to multiclass target recognition and tracking [2]. We also have recently shown this architecture to be particularly suitable for parallel wavelet transform processing [3]. During the course of our research, a 32x 32 and a 64 x 64 thresholding photodetector chip has been designed, built, and tested at JPL [4]. Figure 3 shows the system block diagram of this chip. Each pixel of this photodetector

array consists of a phototransistor detector and electronic processing circuitry that enables direct comparison of the detected light level with a preset threshold level. The dectector outputs the addresses of all those pixels with intensities exceeding the threshold level. This JPL photodetector chip has a 500 frame/see speed, more than an order of magnitude faster than commercially available CCD arrays.

We have also performed experiments to investigate a technique for synthesizing a ternary-valued wavelet filter by using a liquid crystal television spatial light modulator (LCTVSLM). Our results are shown in Figure 4. A ternary-valued triangular wavelet, similar to those used in our computer simulation, is shown in figure 4a. The white, black, and gray areas have -I-1, -1, and O transmission respective] y. The optical Fourier transform of such a wavelet, written into a LCTVSLM operated in a ternary-value mode, is shown in Figure 4b. Computer simulated results are shown in Figure 4C for comparison. The accuracy of this optical synthesis technique is demonstrated by the close resemblance of the results in figures 4b and 4c. The opaque center of the Fourier spectrum of this wavelet demonstrates the wavelet's zero-mean characteristics.

IV. Summary

We have presented an innovative matching pursuit algorithm, for image decomposition and representation. This algorithm is motivated by that described by Mallat [1] with several major modifications to enable the its extension from 1-D time-frequency representations to 2-D image representations. We have demonstrated the effectiveness of this algorithm with a computer simulation. We have proposed an optical implementation that will increase processing speed by several orders of magnitude through massive parallel processing. We have also presented some preliminary optical experimental results. Full-scale laboratory investigation is currently underway at JPL. The resulting system, upon completion, could

prove very useful for real-time image representation as well as for pattern recognition applications.

The research described in this paper was performed by the Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the Ballistic Missile Defense Organization/In novative Science and Technology Office, through an agreement with the National Aeronautics and Space Administration.

References

1. Stephane Mallat and Zhifeng Zhang, "Matching Pursuits with Time-Frequency Dictionaries," to be published in IEEE Transactions in Signal Processing.

2. Tien-Hsin Chao and William W. Stoner, " Optical Implementation of a Feature-based Neural Network with Application to Automatic Target Recognition," Appl. Opt. vol. 32 (8) P. 1359, 1993.

3. Tien-Hsin Chao, Eric Hegblom, Brian Lau, and William J. Miceli, "Optoelectronically Implemented Neural Network with a Wavelet Preprocessor," Proceedings of SPIE Vol. 2026, P.472-482, San Diego, CA., 1993.

4. Harry Langenbacher, Tien-Hsin Chao, Tim Shaw, and Jeffrey Yu, "64 x 64 Thresholding Photodetector Array for Optical Pattern Recognition," Proceedings of SPIE, Vol. 1959, p. 350-358, Orlando, FI .., 1993.

Figure Captions:

Figure 1.

Computer simulation results demonstrating image representation using a matching pursuit algorithm.

(a) input image; (b) an example trinagular-shaped dictionary element; (c) the corresponding wavelet element; (d) residue image after the first iteration; (e) residue image after the second iteration; (f) residue image after the seventh iteration; (g) residue image after the tenth iteration; (h) residue image after the fourteenth iteration; (i) residue image after the twenty-second iteration; and (j) the representattion of the input image as a linear sum of the triangular dictionary element images after the twenty-two iterations - demonstrating close resemblance to that of the input shown in 1a).

Figure 2,

A multichannel optical wavelet processor for the computation of image representation using a matching pursuit algorithm.

Figure 3.

System functional diagram of a JPL developed 32 x 32 photodetector array for peak detection.

Figure 4.

Experimental results showing optical synthesis of ternary-valued wavelet.

(a) A ternary-valued triangular wavelet: the white, black, and gray areas possess +1, -1, and O transmissions respectively;

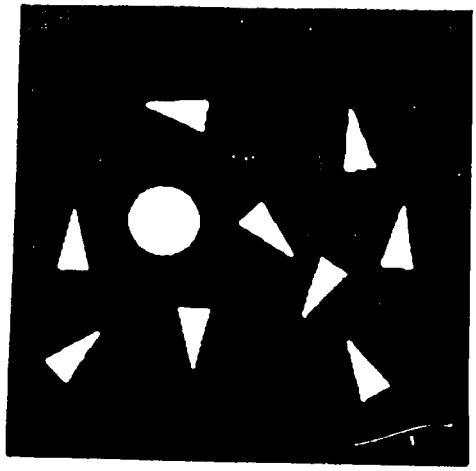
(b) Optical Fourier transform of the input wavelet synthesized with a LCTVSLM;

(c) Computed Fourier transform of the input wavelet.

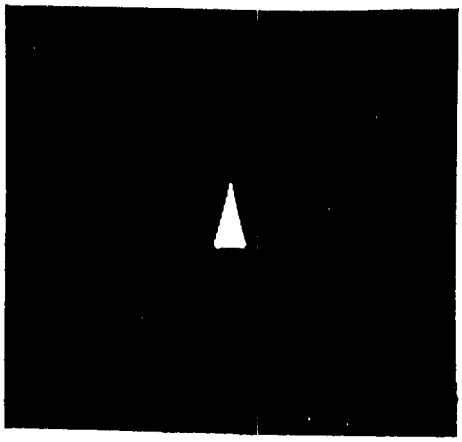
‘I’able I.

data showing progress of the matching pursuit algoritm in image decomposition

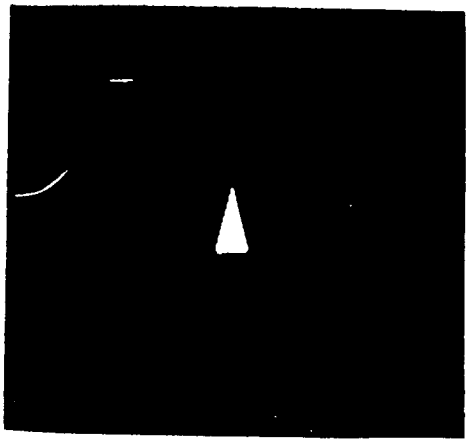
Iteation #	Angular Orientation	Coefficients	Spatial Address	Norm
1	30	238.4	(396, 410)	29558
2	0	226.3	(69, 276)	28539
3	180	217.2.	(204, 365)	27494
4	80	234.7	(188, 119)	26222
5	150	235.1	(344, 309)	24880
6	10	230.7	(379, 157)	23487
7	0	208.3	(430, 263)	22155
8	320	198.2	(71,403)	20820
9	2 2 0	196.3	(280, 248)	19407
10	190	131.5	(1 42, 240)	18298
11	190	127.7	(167, 230)	16691
12	180	120.9	(1 94, 239)	15282
13	20	128.8	(161, 261)	13967
14	40	113.2	(181 , 252)	12871
15	220	91.8	(191, 232)	12210
16	350	90.8	(139, 249)	11873
17	90	72.8	(166, 270)	11574
18	310	73.0	(70, 401)	11373
19	230	73,9	(280, 251)	11140
20	270	68.9	(1 57, 250)	10986
21	310	70.9	(161, 218)	10860
2’2	350	70.5	(430, 265)	10705
23	180	57.8	(206, 368)	10705



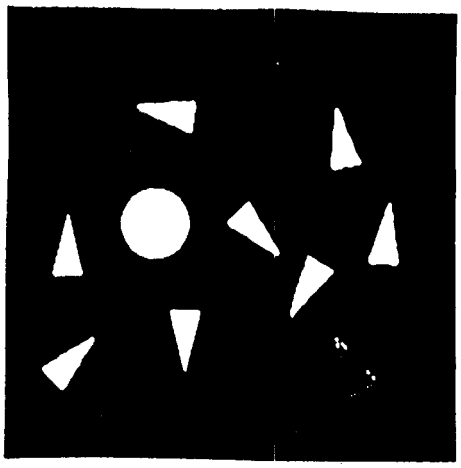
(a)



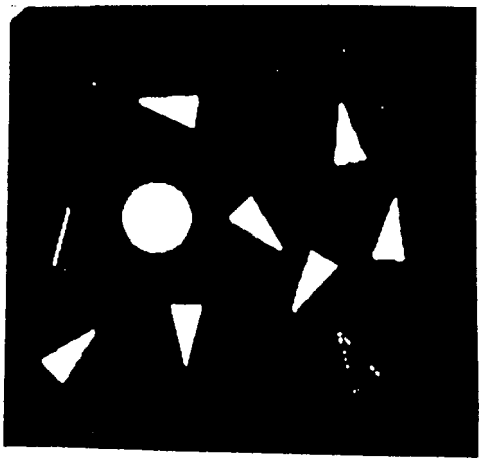
(b)



(c)

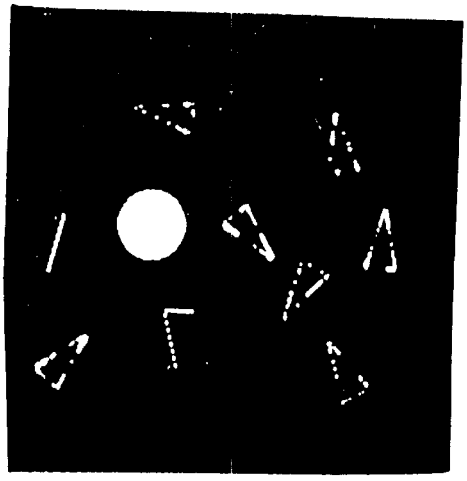


(d)

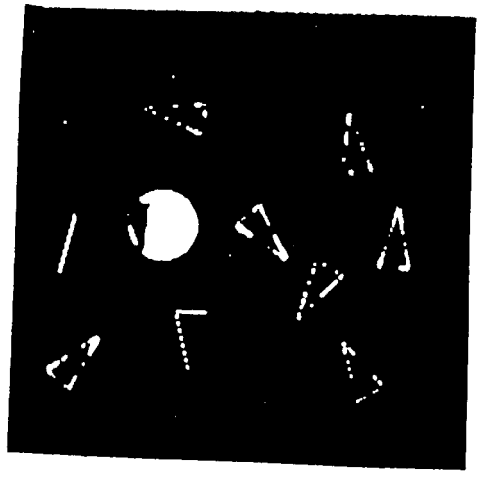


(e.)

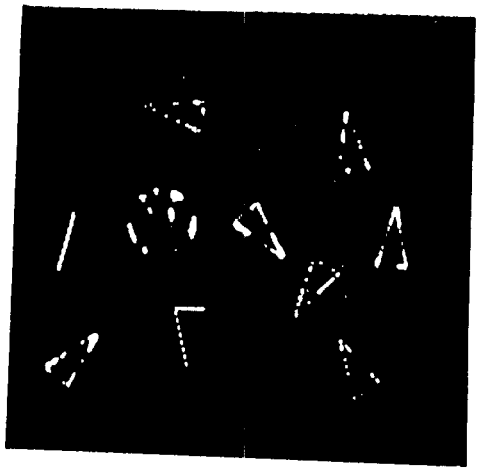
Fig. 1



(f)

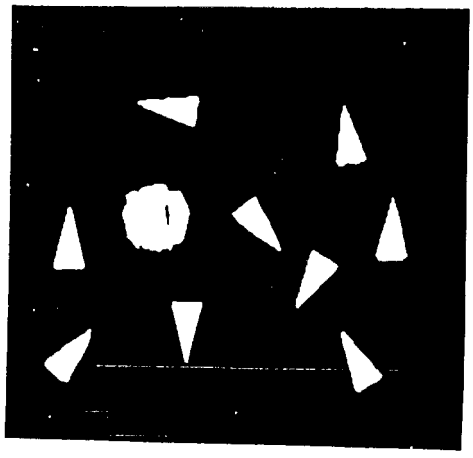


(g)



(h)

(i)



(j)

Fig. 1

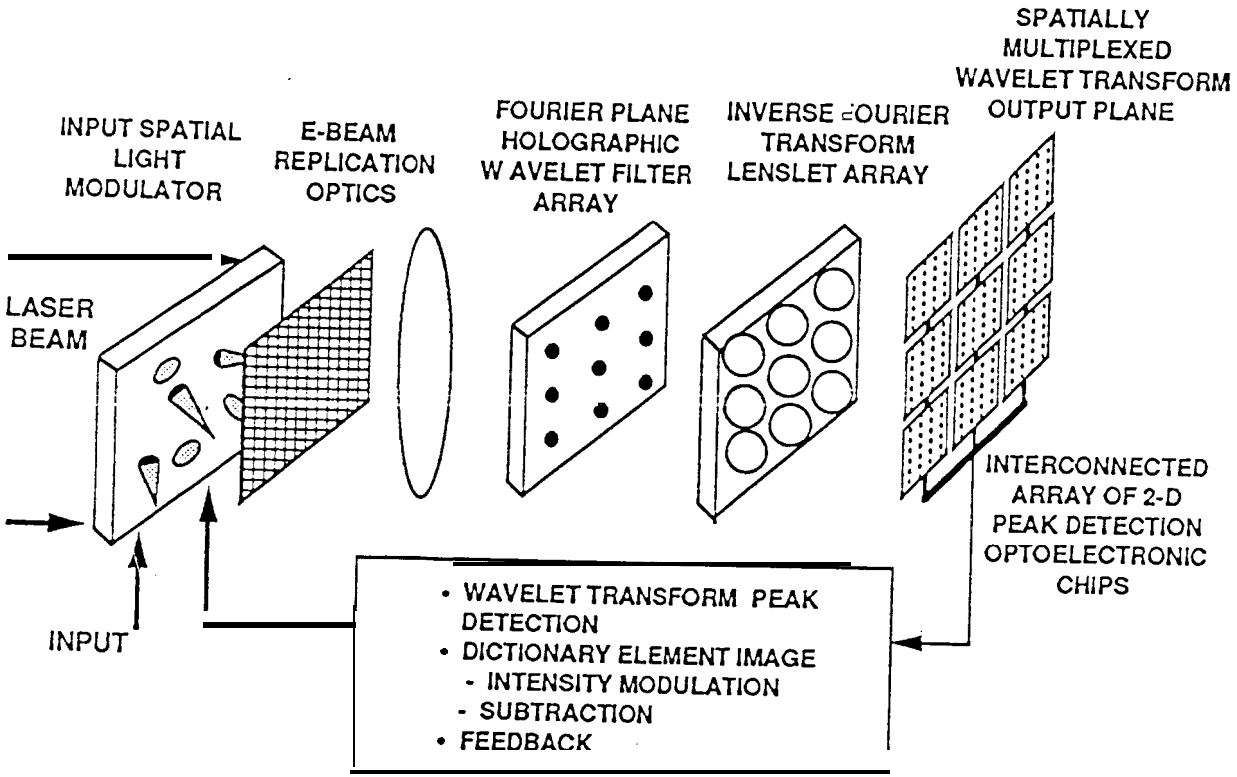

INPUT SPATIAL LIGHT MODULATOR
E-BEAM REPLICATION OPTICS
FOURIER PLANE HOLOGRAPHIC W AVELET FILTER ARRAY
INVERSE FOURIER TRANSFORM LENSLET ARRAY
SPATIALLY MULTIPLEXED WAVELET TRANSFORM OUTPUT PLANE
LASER BEAM
INTERCONNECTED ARRAY OF 2-D PEAK DETECTION OPTOELECTRONIC CHIPS
INPUT
• WAVELET TRANSFORM PEAK DETECTION
• DICTIONARY ELEMENT IMAGE
- INTENSITY MODULATION
- SUBTRACTION
• FEEDBACK

Fig 2

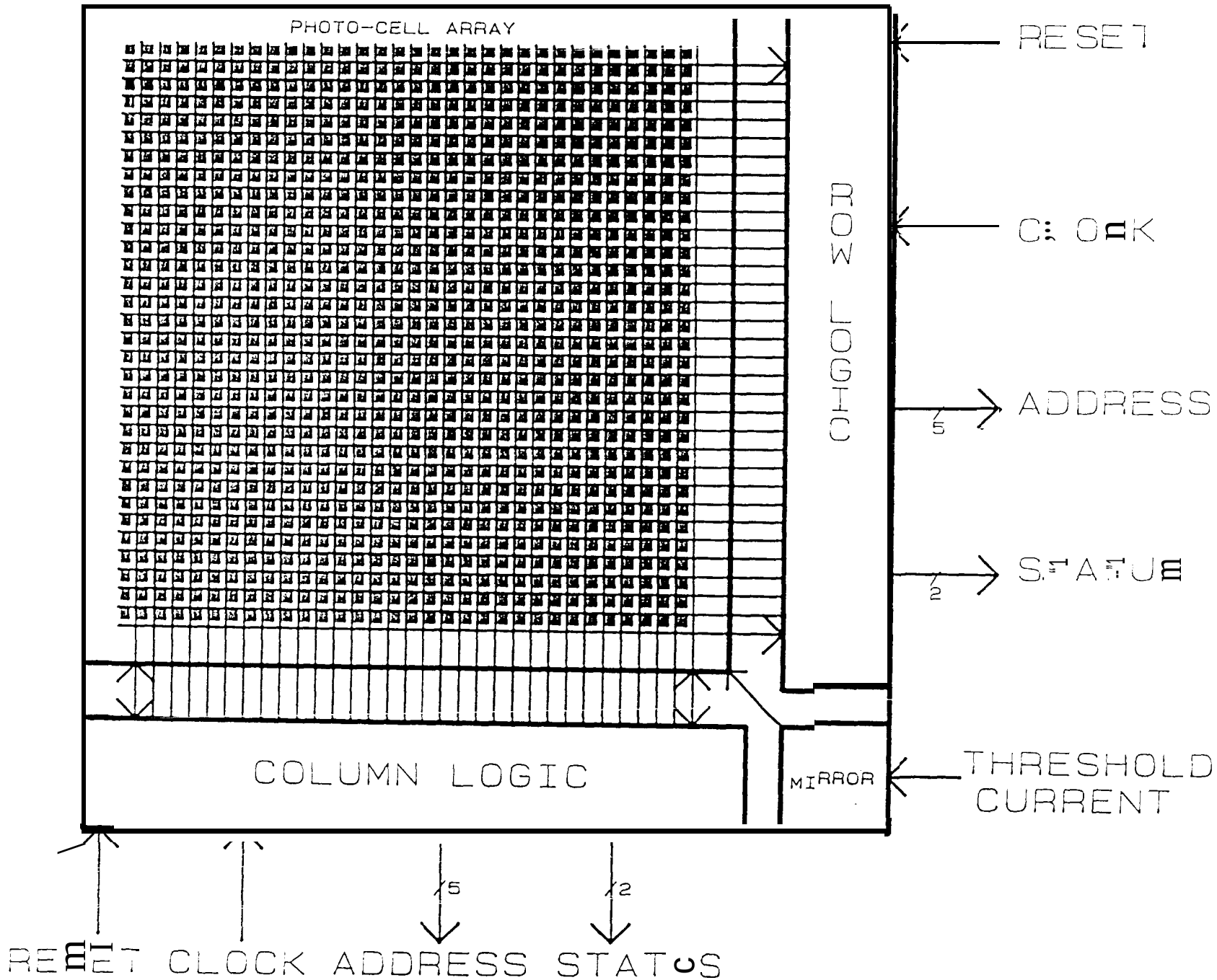


PHOTO-CELL ARRAY
ROW LOGIC
RESET
CLOCK
ADDRESS
5
STATUS
2
COLUMN LOGIC
MIRROR
THRESHOLD CURRENT
RESET CLOCK ADDRESS STATUS
5
2

Fig 3

(a)

(b)

(c)

Fig. 4