SOLVING IMAGE PROCESSING PROBLEMS BY USING NONSTANDARD REGULARIZATION

M.Sc. Thesis

Tolga Acar, B.Sc.

Control and Computer Engineering
Institute of Science and Technology
İstanbul Technical University
80626 Maslak - İstanbul

June 1994

Regularization

An energy functional with constraints is used in regularization. In this thesis, the membrane functional is used with the smoothness constraint.

The String and Membrane

The energy associated with string under tension is given by

$$E_{s1} = \int_{\Omega_s} (u(x) - d(x))^2 dx + \lambda^2 \int_{\Omega_s} u_x^2(x) dx. \quad (1)$$

where

$d(x)$: the input function,

$u(x)$: the reconstructed function,

$u_x(x)$: the first order derivative with respect to x

λ : the regularization parameter.

The functional associated with membrane may be thought as the 2D version of this functional, and is given by

$$E_{m1} = \int\int_{\Omega} (u(x,y) - d(x,y))^2 dxdy + \lambda^2 \int\int_{\Omega} (u_x^2(x,y) + u_y^2(x,y)) dxdy. \quad (2)$$

Weak String and Membrane

The weak form of the string functional is given by,

$$E_s = \sum_i (u_i - d_i)^2 + \lambda^2 \sum_i (u_i - u_{i-1})^2 (1 - \ell_i) + \alpha \sum_i \ell_i. \tag{3}$$

where ℓ is called the *line process.*

The weak membrane has two sorts of line processes in its energy functional,

$$\begin{aligned} E &= D + S + P && (4) \\ D &= \sum_i \sum_j (u_{i,j} - d_{i,j})^2 && (5) \\ S &= \lambda^2 \sum_i \sum_j (u_{i,j} - u_{i-1,j})^2 (1 - m_{i,j}) + (u_{i,j} - u_{i,j-1})^2 (1 - \ell_{i,j}) && (6) \\ P &= \alpha \sum_i \sum_j (\ell_{i,j} + m_{i,j}) && (7) \end{aligned}$$

where $\ell_{i,j}$ and $m_{i,j}$ denote horizontal and vertical line processes respectively.

This functional lacks the mathematical property of convexity, thus cannot be minimized by a regular method.

Modified Functional

To minimize this non-convex functional, Graduated Non-Convexity algorithm is used. To accomplish this, the S and P terms are merged into a single term and the modified functional is obtained as

$$\begin{aligned} E &= \sum_{i,j}(u_{i,j} - d_{i,j})^2 + \\ &\lambda^2 \sum_{i,j} g_{\alpha,\lambda}(u_{i,j} - u_{i-1,j}) + \lambda^2 \sum_{i,j} g_{\alpha,\lambda}(u_{i,j} - u_{i,j-1}) \end{aligned} \quad (8)$$

where $g_{\alpha,\lambda}$ is called the neighbourhood interaction function and given by

$$g_{\alpha,\lambda}(t) = \begin{cases} \lambda^2 t^2, & \text{if } |t| < \sqrt{\alpha}/\lambda \qquad (\ell = 0) \\ \alpha, & \text{otherwise} \qquad (\ell = 1), \end{cases} \quad (9)$$

and the line processes may be recovered at any time by

$$\ell(t) = \begin{cases} 1 & |t| > \sqrt{\alpha}/\lambda \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

The Interaction Function

The interaction function is modified for GNC as

$$g_{\alpha,\lambda}^{(p)} = \begin{cases} \lambda^2 t^2 & |t| < q \\ \alpha - c(|t| - r)^2/2 & q \leq |t| < r \\ \alpha & |t| \geq r, \end{cases} \tag{11}$$

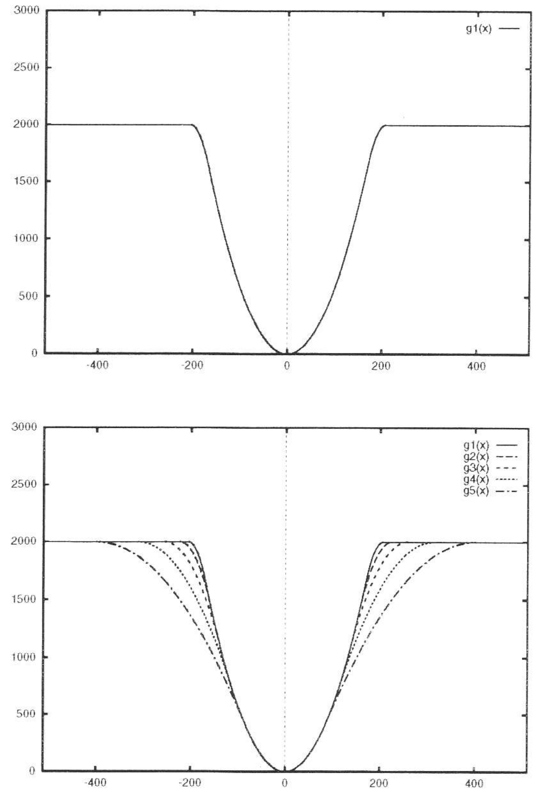


Figure 1: The local interaction function used in GNC. The graph on the left side is the original function. The graph on the right side shows how $g_{\alpha,\lambda}$ is changed by p. g1(x) denotes the unmodified neighbourhood function when $p = 1$. The value of p is 1 for g1(x), 1/2 for g2(x), 1/4 for g3(x), 1/8 for g4(x) and 1/16 for g5(x). λ is 0.24 and α is 2000.

Adaptive Smoothing

In adaptive smoothing, it's aimed to keep discontinuities between regions while performing smoothing operation in the regions.

The adaptive smoothing is given by,

$$I^{(t+1)}(x,y) = \frac{1}{N(x,y)^{(t)}} \sum_{i=-1}^{+1} \sum_{j=-1}^{+1} I^{(t)}(x+i,y+j) \cdot w^{(t)}(x+i,y+j) \quad (12)$$

$I^{(t)}(x,y)$ the signal at (t)th iteration

$w^{(t)}(x,y)$ the filter at (t)th iteration

$I^{(t+1)}(x,y)$ smoothed signal at $(t+1)$th iteration

and

$$N(x,y)^{(t)} = \sum_{i=-1}^{+1} \sum_{j=-1}^{+1} w^{(t)}(x+i,y+j). \quad (13)$$

The suggested filter is given by

$$w^{(t)}(x,y) = f(d^{(t)}(x,y)) = e^{-\frac{|d^{(t)}(x,y)|^2}{2k^2}} \quad (14)$$

$$d^{(t)}(x,y) = \sqrt{G_x^2 + G_y^2}. \quad (15)$$

The Comparison

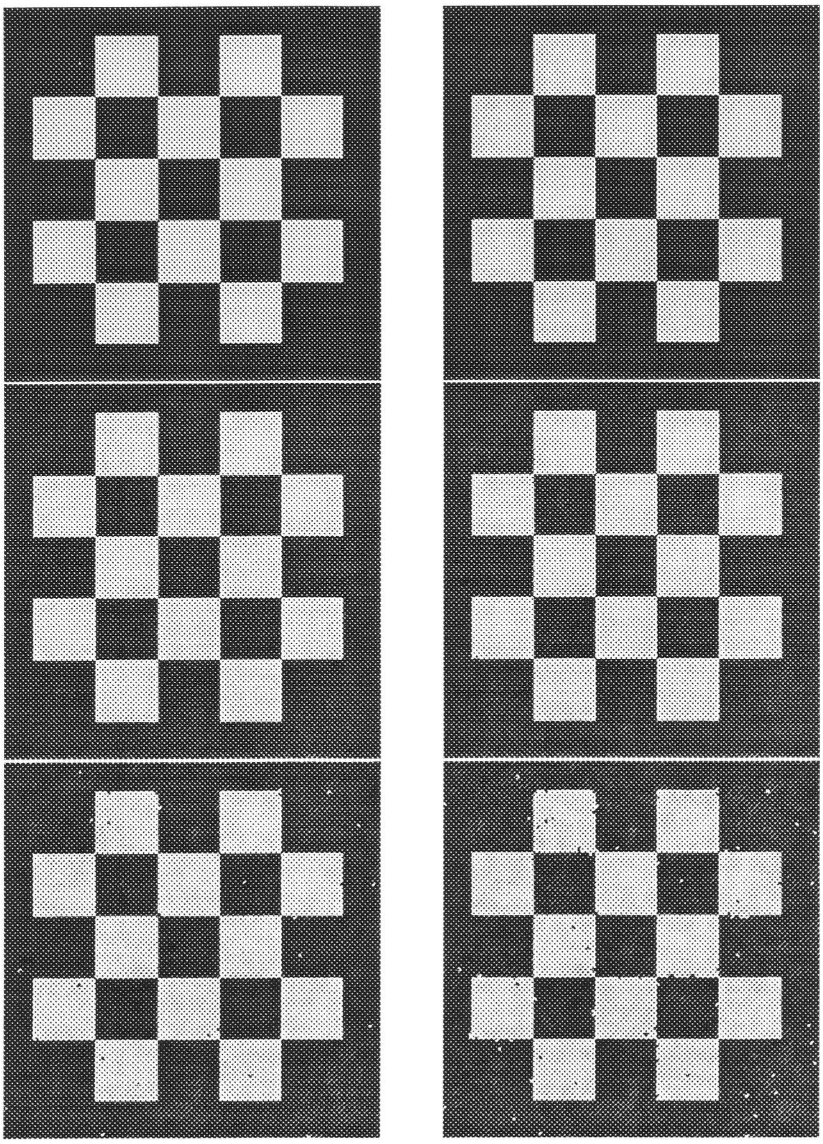


Figure 2: Reconstructed surfaces obtained by weak membrane modeling from noisy checkerboard images. SNR(dB) values from left to right: The first row: No noise, 10, 7. The second row: 5, 4, 3.

The Comparison

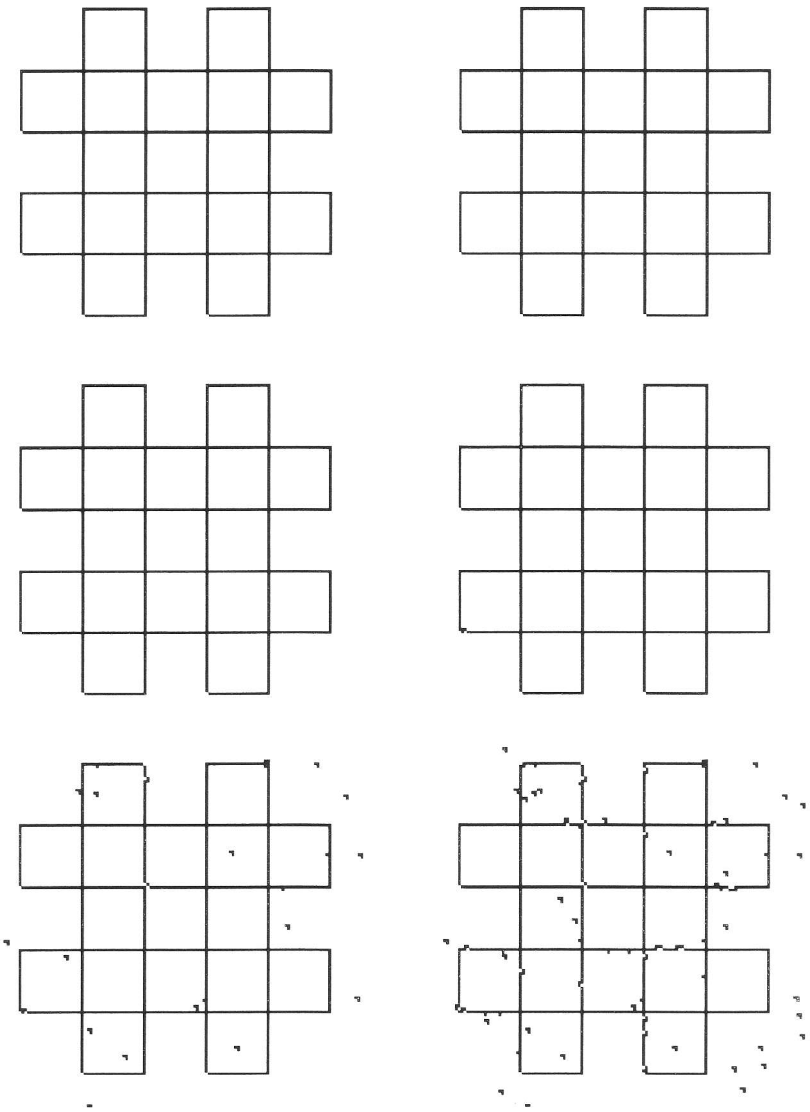


Figure 3: Edge maps obtained by weak membrane modeling from noisy checkerboard images. SNR(dB) values from left to right: The first row: No noise, 10, 7. The second row: 5, 4, 3.

The Comparison

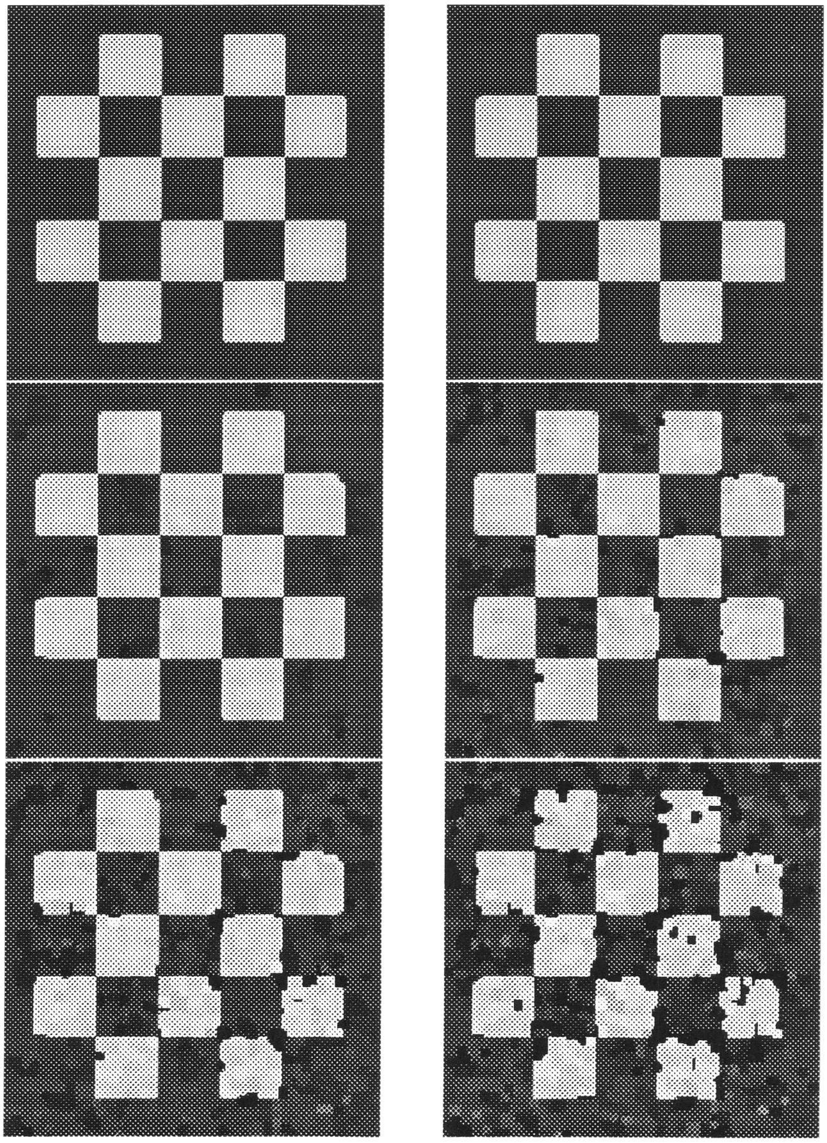


Figure 4: Reconstructed surfaces obtained by adaptive smoothing from noisy checkerboard images. SNR(dB) values from left to right: The first row: No noise, 17, 12. The second row: 10, 8, 7.

The Comparison

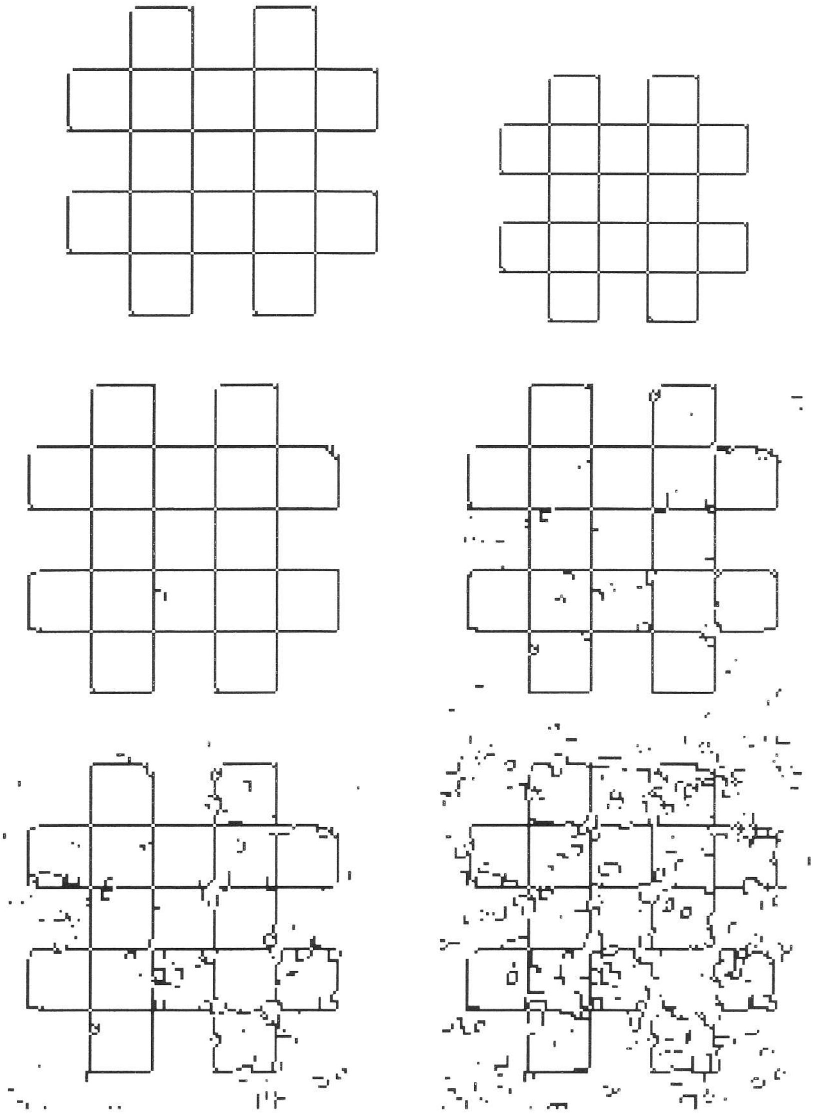


Figure 5: Edges obtained by adaptive smoothing from noisy checkerboard images. SNR(dB) values from left to right: The first row: no noise, 17, 12. The second row: 10, 8, 7.

The Comparison

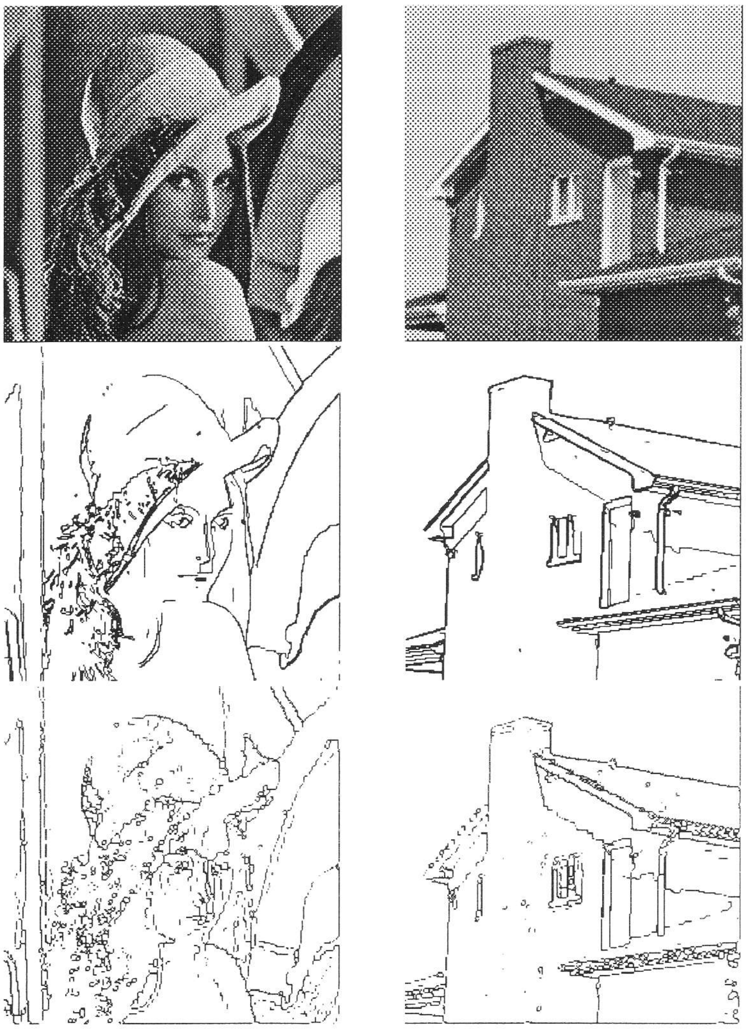


Figure 6: Edges obtained by weak membrane modeling (middle row) and adaptive smoothing (last row) from Lenna and House (first row) images where the same parameters are used for two of the images. $\lambda = 1.8$, $\alpha = 1000$ and $SOR - \omega = 1.2$ in weak membrane modeling. Scale parameter $k = 2$ and threshold $\tau = 24$ in adaptive smoothing.

Image Restoration

IMAGE RESTORATION USING REGULARIZATION

Goal: Restoration of blurred and noisy image and elimination of noise.

Reasons of degradation

- Blurring:
  - ◇ Motion of the object or the imaging system.
  - ◇ Misfocusing of the lens system.
  - ◇ Atmospheric reasons.
- Noise:
- Recording media used during imaging (such as film particles).
- Digitization of the image.
- Quantization errors.

Image Restoration

Blurred and Noisy Image Model:

$$g = D \cdot f + n \tag{16}$$

where,

g: Blurred and noisy (degraded) image,

D: Blurring function,

f: Original image,

n: The noise effect.

A restoration system should satisfy the following criteria:

- Deblurring,
- Suppression of noise,
- Preservation of discontinuities,
- Reducing the ringing effect.

Image Restoration

The proposed method

Modification of weak membrane model to include blurring which had been developed for surface reconstruction and edge detection.

Addition of blurring to the energy functional associated with the model is established by the following modified functional:

$$\begin{aligned} E(\mathbf{f},\mathbf{l},\mathbf{m}) = & \sum_i \sum_j [(f*b)_{i,j} - d_{i,j}]^2 + \\ & \lambda^2 \, [\sum_i \sum_j (f_{i,j} - f_{i-1,j})^2 (1 - l_{i,j}) + \\ & \quad \sum_i \sum_j (f_{i,j} - f_{i,j+1})^2 (1 - m_{i,j})] + \\ & \alpha \, \sum_i \sum_j (l_{i,j} + m_{i,j}) \end{aligned} \tag{17}$$

where, b is the point spread function which causes blurring (PSF).
In this model, the blurred version of the restored image, i.e. $f*b$, is desired to be close to the degraded input image d.

The Modified Functional

- The modified energy functional is minimized by using the Graduated Non-Convexity algorithm.

- The modified functional is further modified to use GNC in the minimization process. The modified functional is,

$$E^{(p)}(\mathbf{f}) = \sum_i \sum_j [(f * b)_{i,j} - d_{i,j}]^2 + \sum_i \sum_j g^{(p)}_{\alpha,\lambda}(f_{i,j} - f_{i-1,j}) + \sum_i \sum_j g^{(p)}_{\alpha,\lambda}(f_{i,j} - f_{i,j+1}) \quad (18)$$

where $g^{(p)}_{\alpha,\lambda}(.)$ is the modified local interaction function as defined before.

SOR Iterations

The SOR iteration is expressed as

$$f_{i,j}^{(n+1)} = f_{i,j}^{(n)} - \frac{\omega}{T} \cdot \frac{\partial E^{(p)}(\mathbf{f})}{\partial f_{i,j}}, \tag{19}$$

where,

w is the relaxation parameter and n denotes the iteration number.

The following iteration is obtained by taking the derivatives and taking

$$T = 2 \cdot (1 + 4\lambda^2) \tag{20}$$

$$\begin{aligned} f_{i,j}^{(n+1)} \quad = f_{i,j}^{(n)} - \omega[\; & \frac{\partial}{\partial f_{i,j}}((f^{(n)} * b)_{i,j} - d_{i,j})^2 + \\ & g'_{\alpha,\lambda}(f_{i,j}^{(n)} - f_{i-1,j}^{(n+1)}) + \\ & g'_{\alpha,\lambda}(f_{i,j}^{(n)} - f_{i,j-1}^{(n+1)}) + \\ & g'_{\alpha,\lambda}(f_{i,j}^{(n)} - f_{i+1,j}^{(n)}) + \\ & g'_{\alpha,\lambda}(f_{i,j}^{(n)} - f_{i,j+1}^{(n)}) \;] \; \frac{1}{2 + 8\lambda^2}. \end{aligned} \tag{21}$$

The Blurring Filter

Point spread function b is taken as the averaging filter which corresponds to misfocusing of the lens system.

The 3×3 averaging filter is,

$$b = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \qquad (22)$$

The derivative in Equation (19)

$$\frac{\partial}{\partial f_{i,j}}(f^{(n)} * b)_{i,j} - d_{i,j})^2 \qquad (23)$$

If the blurring function b is taken as in Equation (22), the SOR iterations becomes

The SOR Iterations

$$\frac{\partial}{\partial f_{i,j}}(f_{i-1,j-1}+ f_{i-1,j} + f_{i-1,j+1} + f_{i,j-1} + f_{i,j} + f_{i,j+1}+ f_{i+1,j-1} + f_{i+1,j} + f_{i+1,j+1} - d_{i,j})^2 = \qquad (24)$$

$$2\{ \frac{1}{9}[f_{i,j} + f_{i,j+1} + f_{i,j+2} + f_{i+1,j} + f_{i+1,j+1} + f_{i+1,j+2} + f_{i+2,j} + f_{i+2,j+1} + f_{i+2,j+2}] + \dots \dots \dots \dots \dots \dots \frac{1}{9}[f_{i-2,j-2} + f_{i-2,j-1} + f_{i-2,j} + f_{i-1,j-2} + f_{i-1,j-1} + f_{i-1,j} + f_{i,j-2} + f_{i,j-1} + f_{i,j}] - d_{i,j} \}. \qquad (25)$$

Experimental Results

Parameters

Typical values of λ and α values for restoration are found as $\lambda = 1.6$ and $\alpha = 1600$ in the energy functional.

Table 1: Normalized mean square error values for blurred and noise added Checkerboard, House and Clock images.

Image	-		SNR=30 dB	
	Gauss σ_n	NMSE	*Gauss* σ_n	NMSE
che1	-	133	7	6563
che2	-	22	7	6509
che3	-	113	6.9	6664
house1	-	798	5.5	8283
house2	-	544	5.4	8222
house3	-	570	5.3	8283
clock1	-	1314	5.6	6533
clock2	-	852	5.5	6501
clock3	-	902	5.5	6723

Experimental Results – The Images

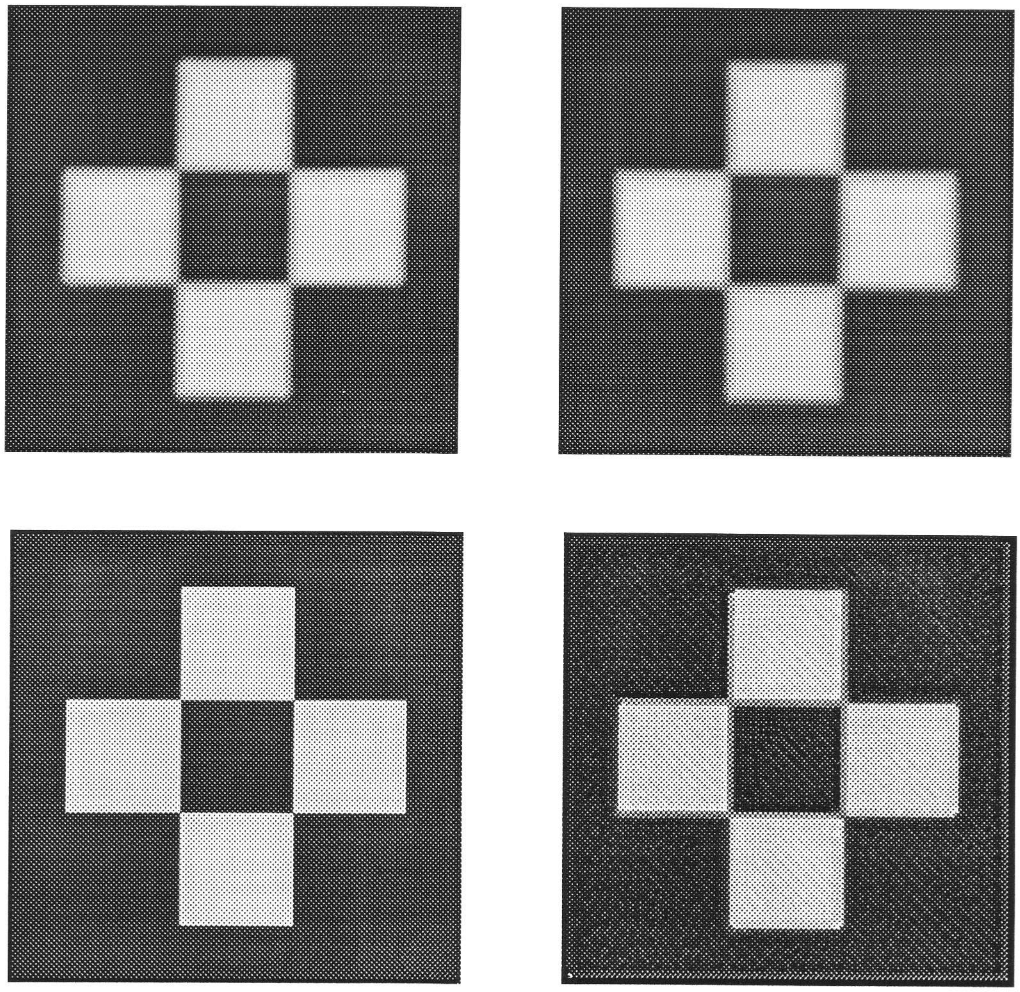


Figure 7: Top row: The blurred checkerboard images degraded by applying a 3x3 averaging filter two (left) and three (right) times. Bottom row: Restored versions of the images.

Experimental Results – The Images

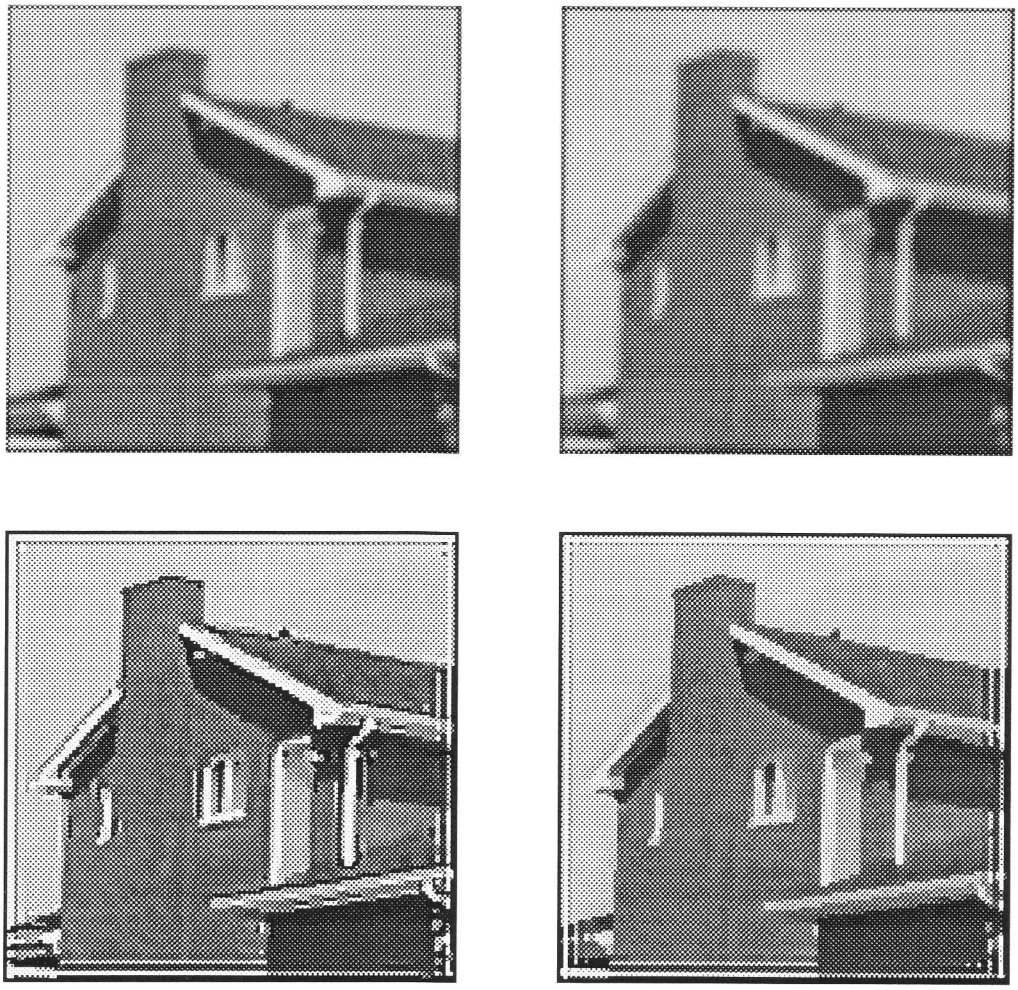


Figure 8: Top row: The blurred house images degraded by applying a 3x3 averaging filter two (left) and three (roght) times. Bottom row: Restored versions of the house images.

Experimental Results – The Images

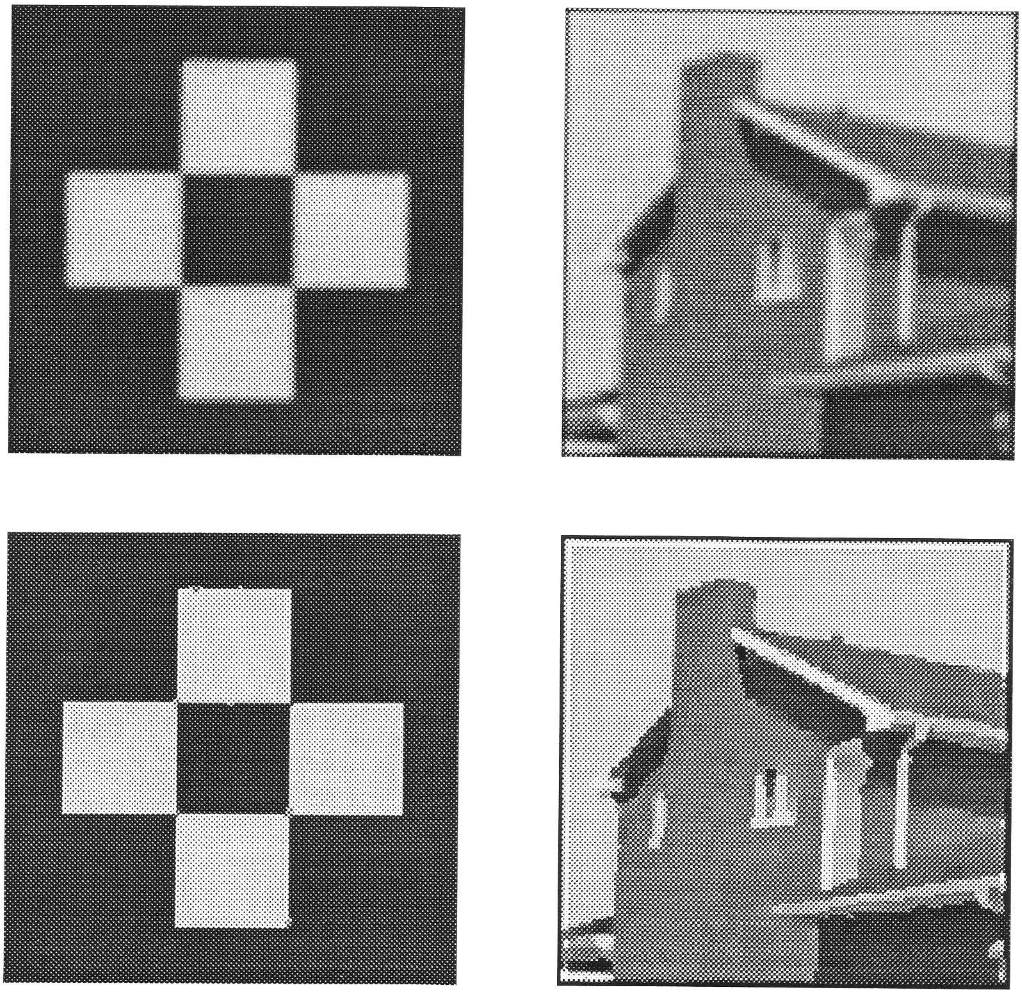


Figure 9: Blurred and noise added (SNR=25 dB) checkerboard and house images and their restored versions.

Image Compression

The Approach:

- An image can be reconstructed from sparse data without excessive degradation where the sparse data reside along discontinuities.
- In this approach, image is modeled as a collection of smooth regions separated by edge contours by using weak membrane model.
- This model allows us to determine edge contours, represented as line processes, by minimizing a non-convex functional associated with weak membrane, and to reconstruct the original image by using the same model.
- The line processes are coded by run length coding since they are boolean valued.
- The collection of gray values associated with line processes is coded by using the variable length coding via Huffman coding method.

Image Compression

- Transmitter
  - ◇ Detect edges and specify data along edge locations.
  - ◇ Code line processes by run length coding.
  - ◇ Code sparse data along discontinuities by variable length coding.
- Receiver
  - ◇ Unpack run length coded line processes and Huffman coded data.
  - ◇ Locate sparse data by using line processes.
  - ◇ Perform surface reconstruction from sparse data by using weak membrane model.

The Coding Part

The energy functional associated with weak membrane including sparse data is given by

$$E = D + S + P \tag{26}$$

$$D = \sum_i \sum_j \beta_{i,j} \cdot (d_{i,j} - u_{i,j})^2 \tag{27}$$

$$S = \lambda^2 \sum_i \sum_j (1 - \ell_{h_{i,j}})(u_{i,j} - u_{i-1,j})^2 + (1 - \ell_{v_{i,j}})(u_{i,j} - u_{i,j-1})^2 \tag{28}$$

$$P = \alpha \sum_i \sum_j (\ell_{v_{i,j}} + \ell_{h_{i,j}}) \tag{29}$$

where β is used to mark sparse data as

$$\beta_{i,j} = \begin{cases} 1, & \text{if data is available at } (i,j), \\ 0, & \text{otherwise.} \end{cases} \tag{30}$$

The line processes may be illustrated as

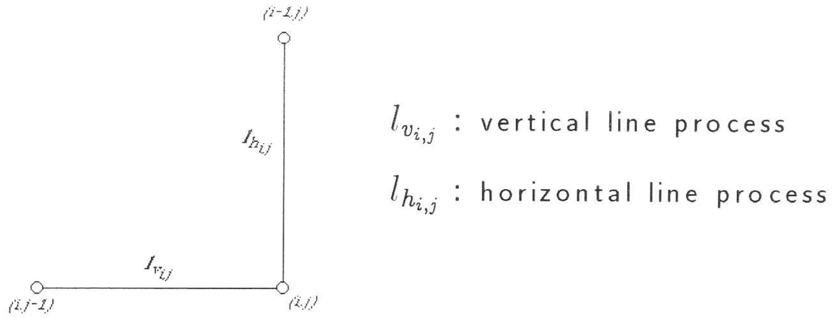


Figure 10: Line processes in two dimensions.

Experimental Results – Compression

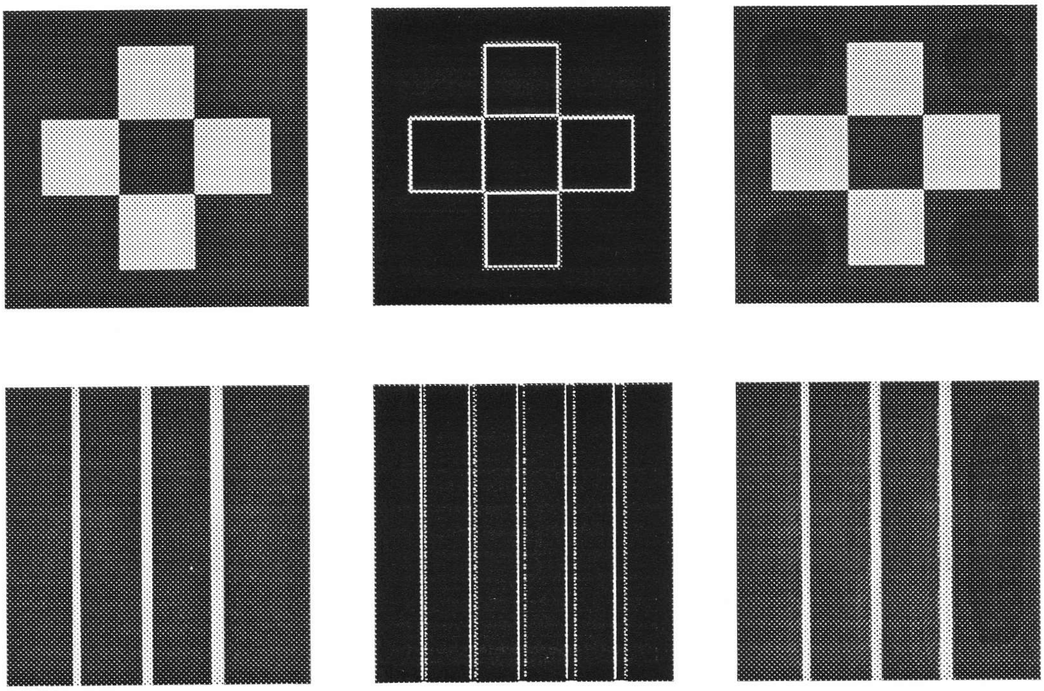


Figure 11: First column: Original 128x128 checkerboard and 180x180 bars images. Second column: Sparse data to be transmited. Third row: Reconstructed versions of checkerboard and bars images.

Experimental Results – Compression

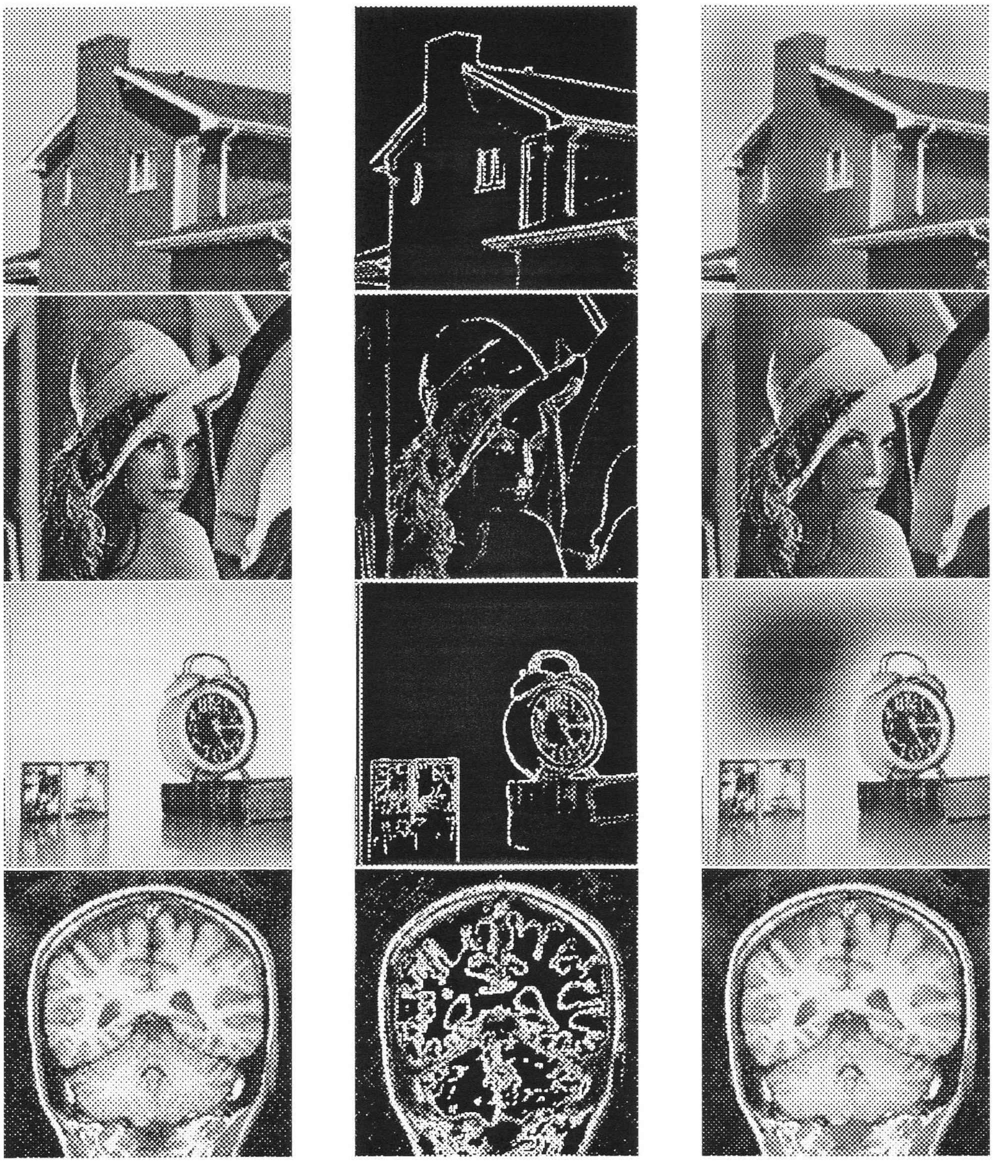


Figure 12: First column: Original 256x256 House, Lenna, Clock and 175x175 Brain images. Second column: Sparse data to be transmitted. Third column: Reconstructed versions of images.

The Degradation

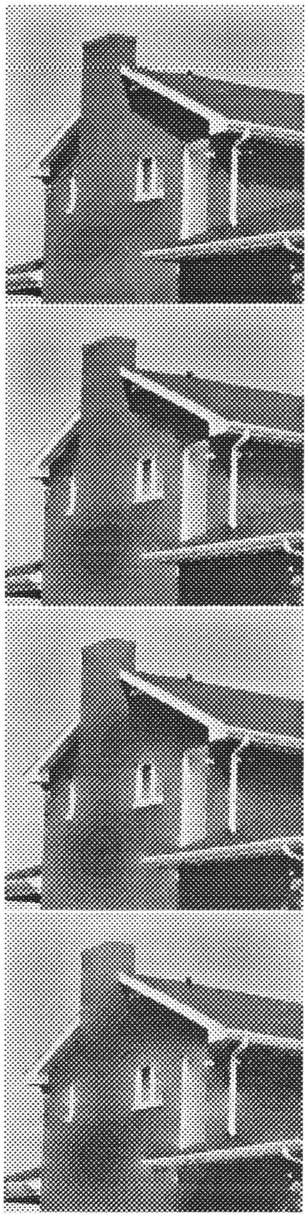


Figure 13: Change of reconstructed image as the compression ratio increases from left to right as 4.8:1, 5.3:1, 6:1, 8:1.

Experimental Results – Compression

Table 2: Quantitative evaluation of the coding method on various images.

Image	Size	Compressed Size (bytes) Run bits	Line proc.	Data entropy	Data size	Coded Data	Total	Compression Ratio	%	NMSE
Che.	16kb	6	527	0.876	1258	188	715	23:1	96	62
Bars	16kb	6	708	0.942	1636	234	942	17:1	94	142
House	64kb	6	3153	7.538	8613	9037	12190	5.3:1	81	313
Lenna	64kb	4	5203	7.572	14004	14663	19866	3.2:1	70	238
Clock	64kb	5	4121	7.607	10302	11028	15149	4.3:1	77	4998
Brain	30kb	3	3522	6.281	14693	11954	15476	2:1	50	89

DORS Representation

Each level of this representation, called the *Difference Of Regularized Solutions* (DORS) representation, is obtained by taking the difference between two regularized solutions with different regularization parameters.

The regularized solutions are obtained by

$$\begin{aligned} v(x,y,\lambda_i) \;&=\; \{v(x,y) : E_m(f,\lambda) = \\ &\inf_{f\in V} \int\!\!\int_\Omega (f-d)^2\, dxdy + \lambda_i \int\!\!\int_\Omega (f_x^2 + f_y^2)\, dxdy\} \end{aligned} \tag{31}$$

where λ_i is taken from the Λ set as

$$\Lambda \;=\; \{\lambda_1, \lambda_2, ..., \lambda_n\}, \qquad \forall_i,\ \lambda_i < \lambda_{i+1}. \tag{32}$$

DORS Representation

The $DORS(x, y; \lambda_i, \lambda_j)$ is defined as the difference of two regularized solutions with regularization parameters λ_i and λ_j, that is,

$$DORS(x, y; \lambda_i, \lambda_j) = v(x, y; \lambda_i) - v(x, y; \lambda_j), \qquad \lambda_i \neq \lambda_j. \tag{33}$$

The multiscale edge images are obtained by

$$w_{i,j} = zc\,[\,DORS(x, y; \lambda_i, \lambda_j)\,] \tag{34}$$

and

$$zc[DORS(x, y; \lambda_i, \lambda_j)] = \begin{cases} 1 & \text{if } DORS(x, y; \lambda_i, \lambda_j) \\ & \text{has a zero crossing at } (x, y) \\ 0 & \text{otherwise.} \end{cases} \tag{35}$$

Vietnamese Hat Operator

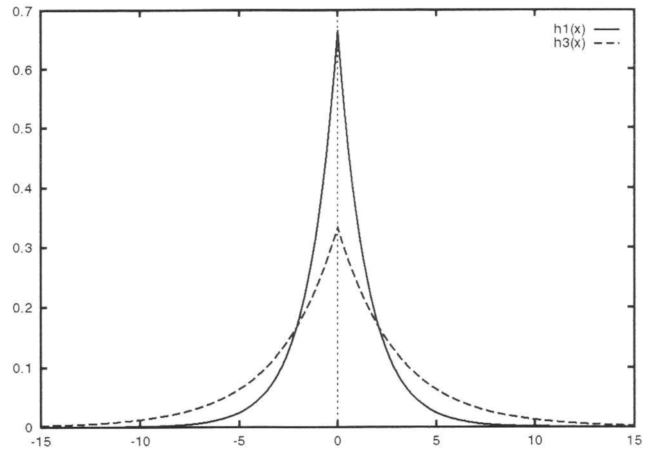


Figure 14: R-filters $h1(x) = h_r(x; 1.5)$ and $h2(x) = h_r(x; 3)$.

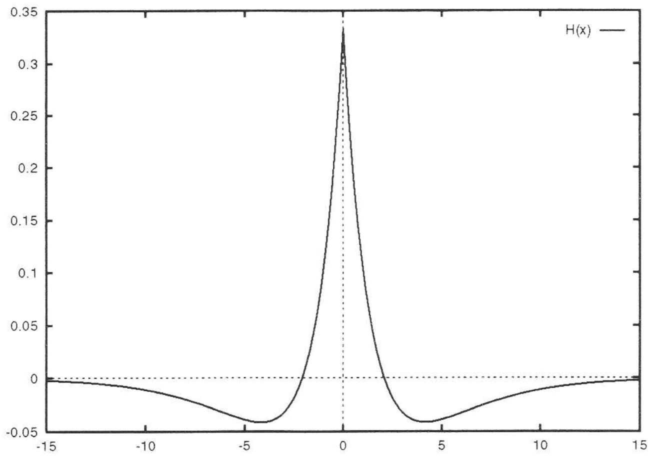


Figure 15: The Vietnamese hat operator $H(x; 1.5, 3)$.

DORS Representation

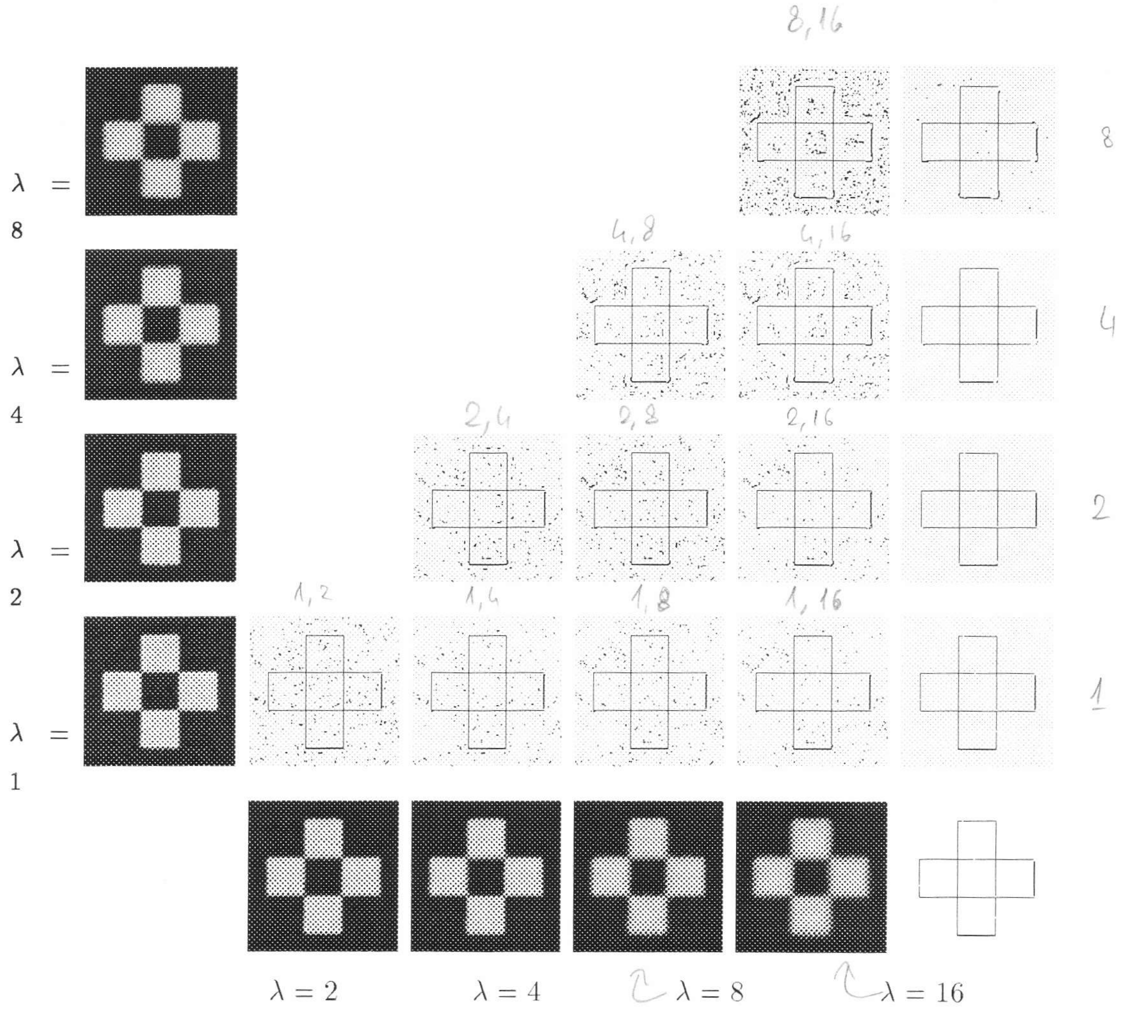


Figure 16: Checkerboard images (SNR=7dB) used in DORS and integration. The first column and the last row show regularized solutions with the indicated parameters. The noisy edge images obtained by using two images with DORS in the corresponding row and column are presented in the lower part of the diagonal. In the right-most column, results of intermediate integration for DORS and in the lower-right corne, the final integrated edge image are presented.

Multiscale Edge Integration

Goal: Integration of multiscale edge images to obtain a single edge image since edges have multisize features.

Approach: Using a multiscale edge integration scheme to overcome shifting, appearing and disappearing and branching of edges in scale space, which uses a weighted accumulation array.

Problems in multiscale edge integration

- *Shifting:* The location of edges in coarse scales is generally wrong.
- *Disappearing:* An edge point may not occur in all of the scales.
- *Appearing:* New edge points may disappear in different scales.
- *Branching:* There may be more than one edge point in the finer scale corresponding to an edge in the coarse scale.

Multiscale Integration

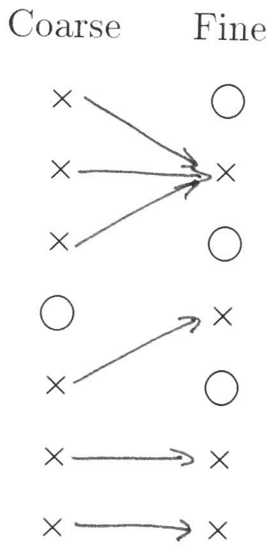
Edge Image Set Used in Integration

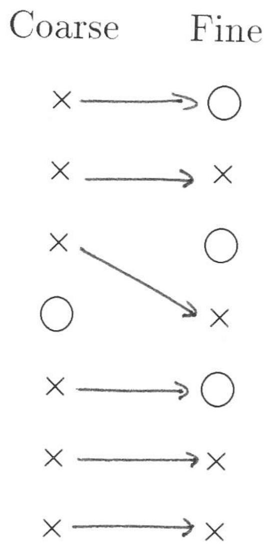
$$W = \{w_1, w_2, ..., w_n\} \tag{36}$$

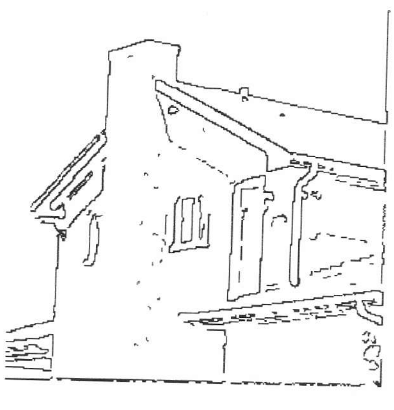
COARSE	w_n	The coarsest scale edge image
MID	w_{n-1}	Edge images ordered from the coarsest scale w_n to the finest scale w_1 $(w_n, ..., w_1)$.
⋮	⋮	
MID	w_2	
FINE	w_1	The finest scale edge image.

Edge Tracking

Edge tracking starts from the coarsest scale and proceeds down to the finest scale, processing the consequitive edge images in each step.





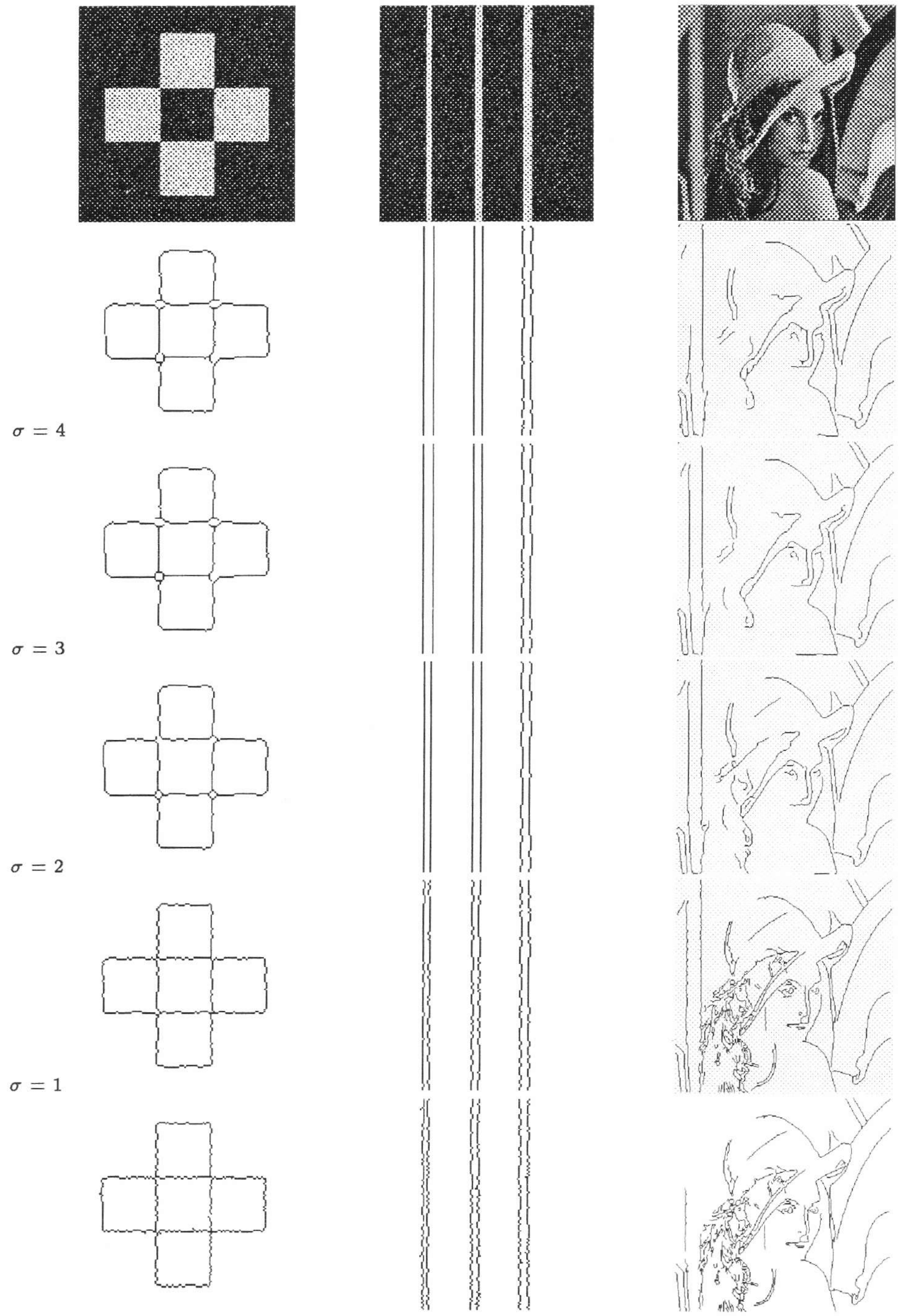


Edge Tracking

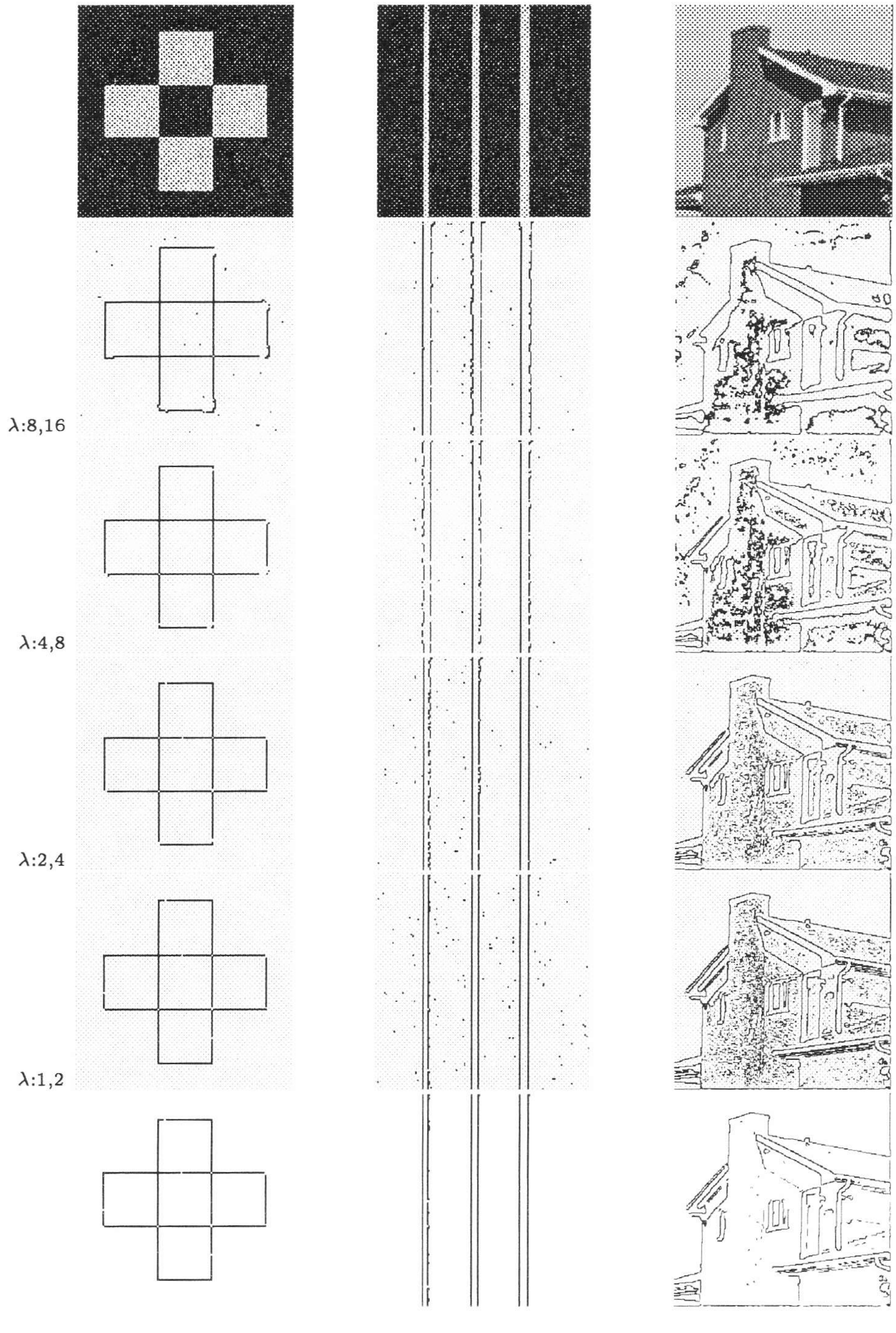
	COARSE	MID	FINE	
Appearance	YES	YES	NO	Edge exists in fine scale, but not in coarse scale.
Prevention disappearance	YES	YES	YES	Edge exists in coarse scale, but not in fine scale.
Branching	YES	YES	NO	There are more than one fine scale edge points corresponding to one edge point in coarse scale.

- A search for edge points in the fine scale is performed in a 3x3 sized window for an edge point in coarse scale.
- The coarse scale edge point is linked to fine scale edge point(s) starting from the nearest one (*branching*).
- The accumulation array is updated in each link, $acc(k,l) = \alpha_n(i,j) \cdot w_n(k,l) + acc(i,j)$.
- The initial values of the accumulation array are the edge locations in the coarsest scale, i.e. $acc(i,j) = w_n(i,j)$.

Integrated Edges (Canny)



Integrated Edges (DORS)



Summary of Contributions

- It is observed that, edge detection performance of weak membrane modeling is superior than that of the adaptive smoothing.
- A non-standard regularization based image restoration method is developed and deblurred images are obtained.
- A coding/decoding scheme is introduced where discontinuities and sparse information along discontinuities are used for compression.
- The DORS representation is used in two dimensions to obtain multiscale edge images from two regularized solutions.
- A multiscale edge integration algorithm has been extended for two dimensions which uses a weighted accumulation array.

Suggestions for Future Research

- The blurring filter used in image restoration is the 3x3 averaging filter. The introduced method may be used with different filters and the behaviour should be investigated.
- The compression algorithm can be improved to achieve higher compression ratio. The first stage that should be considered is to implement differential Huffman coding for sparse data along discontinuities.
- The integration algorithm can be developed by incorporating a more sophisticated multiscale edge tracking algorithm. The edge linking and search processes can be realized not only between adjacent scales, but also between all multiscale edge images.