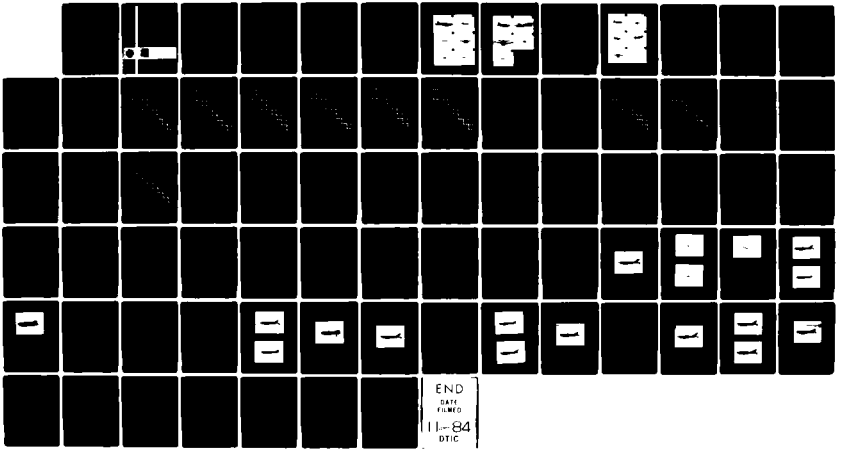
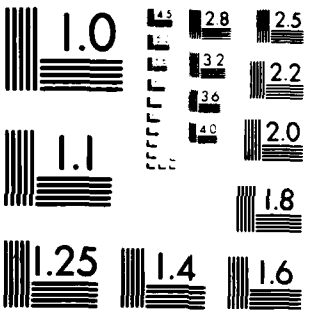
AD-A146 728 PATTERN RECOGNITION OF IMAGES IN ATMOSPHERIC TURBULENCE 1/1
(U) ARIZONA UNIV TUCSON DIGITAL IMAGE ANALYSIS LAB
B R HUNT ET AL. 1983 SIE/DIAL-83-006 N00014-82-K-2060
UNCLASSIFIED F/G 4/1 NL





12

DIAL-83-006

Pattern Recognition of Images in Atmospheric Turbulence

Final Report

Contract No. N00014-82-K-2060

B. R. Hunt

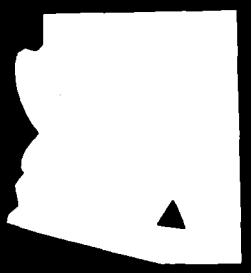
K. Morgan

K. West

Department of Electrical and Computer Engineering

University of Arizona

Tucson, Arizona 85721



ENGINEERING EXPERIMENT STATION
COLLEGE OF ENGINEERING
THE UNIVERSITY OF ARIZONA
TUCSON, ARIZONA

DIAL-83-006

Pattern Recognition of Images in Atmospheric Turbulence

Final Report

Contract No. N00014-82-K-2060

B. R. Hunt

K. Morgan

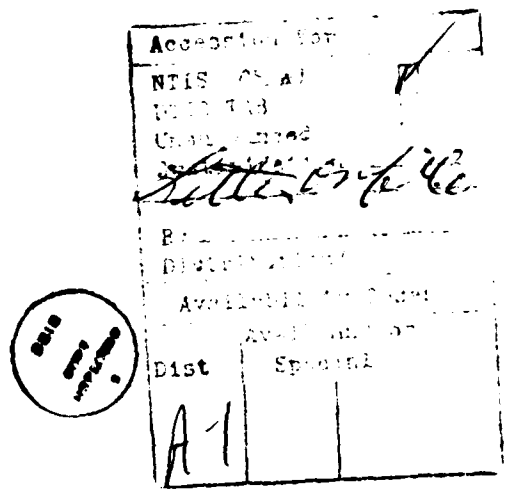
K. West

Department of Electrical and

Computer Engineering

University of Arizona

Tucson, Arizona 85721



Introduction

This report covers the two major activities investigated this year. (1) Preprocessing, feature extraction and classification algorithms have been applied to classify data from the NRL data-base. Invariant moments and normalized Fourier descriptors were the features used in these classification experiments, which show that the Fourier descriptors are better suited to classification of aircraft in turbulent atmosphere. Weak points of the algorithms are discussed and directions for further research are indicated. (2) Post-processing of image data to remove the effects of atmospheric turbulence was also studied. Atmospheric turbulence was simulated, by generation of random phase components, and applied to an idealized model of the drone in the NRL data-base. An algorithm for atmospheric deblurring, known as shift-and-add, was used. The shift-and-add algorithm produced images free of turbulence affects but degraded by blur. Image restoration was applied to the processed images and definite improvements in image quality were obtained.

(I) Data Base Analysis

NRL has supplied a data base of FLIR imagery of a stationary drone acquired at White Sands Missile Range (see Final Technical Report No. DIAL-82-010, October 30, 1982). Twenty images from each of the eleven distinct sequences in the base were selected for classification studies. The parameters for these image sequences are as given in the table of Figure 1. This imagery was processed as follows:

(1) Windowing to eliminate artificial background in the original data. This results in the smaller image sizes shown in the table.

(2) Optimum global binary thresholding, basically as described in the last Final Report. This scheme assumes a bimodal Gaussian distribution for the image, picking a threshold which maximizes the between-class variance. The technique was modified to disallow thresholds not contained in the "saddle" of the bimodal distribution. The threshold ranges obtained by this automatic routine are given in the table of Figure 1.

(3) Median filtering to reduce noise.

Figure 2 contains representative pictures of the eleven groups after preprocessing.

The images thus prepared were used in classification studies utilizing invariant moment (IM) and normalized Fourier descriptor (NFD) features. It became evident that the three groups at a distance of 15 km. (groups 3, 6, 11) were, for practical purposes, of no use

GROUP	FID	ORIGINAL NRL TAPE	FILE # ON ORGTAPE	SIZE	PITCH	YAW	RANGE (KM)	TURB.	THRESH.
1	1-20	2	1-20	155x45	0	30	3	HIGH	116-120
2	21-40	4	1-20	78x18	0	30	6.7	MOD	108-115
3	41-60	1	43-62	33x8	0	45	15.7	LOW	102-112
4	61-80	1	165x184	155x45	0	0	3	HIGH	101-105
5	81-100	3	83x102	78x18	0	0	6.7	MOD	97-113
6	101-120	1	83-102	33x8	0	0	15.7	LOW	81-90
7	121-140	3	42-61	155x45	0	15	3	HIGH	105-108
8	141-160	3	1-20	155x45	15	30	3	HIGH	111-115
9	161-180	1	124-144	155x45	-15	0	3	HIGH	103-105
10	181-200	3	124-144	78x18	15	0	6.7	MOD	109-119
11	200-220	1	1-20	33x8	0	-90	15.7	LOW	71-80

Figure 1: Parameters of Selected Database.

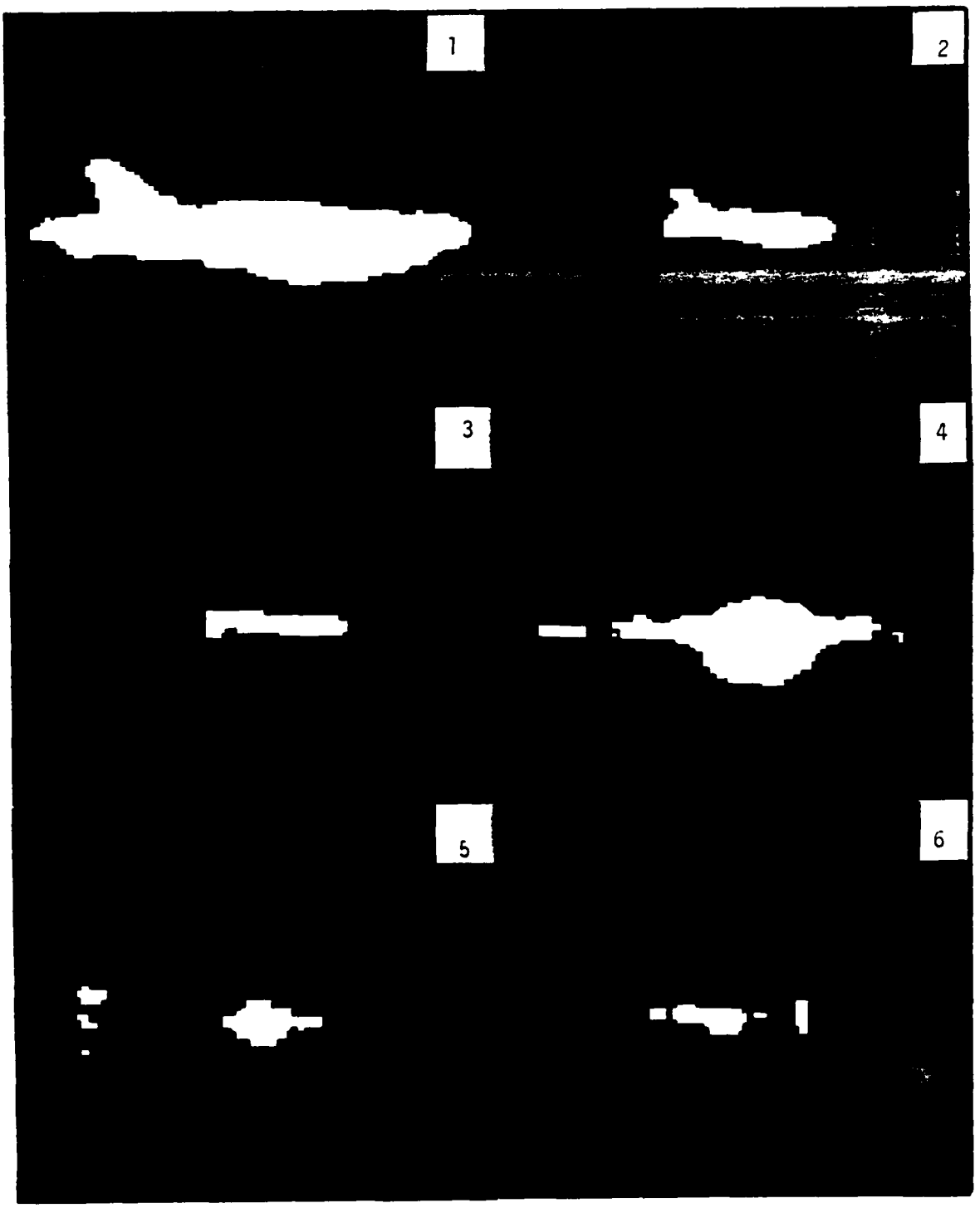


Figure 2: Representative Pictures of Data Base. Group Number Appears in Upper Right Corner.

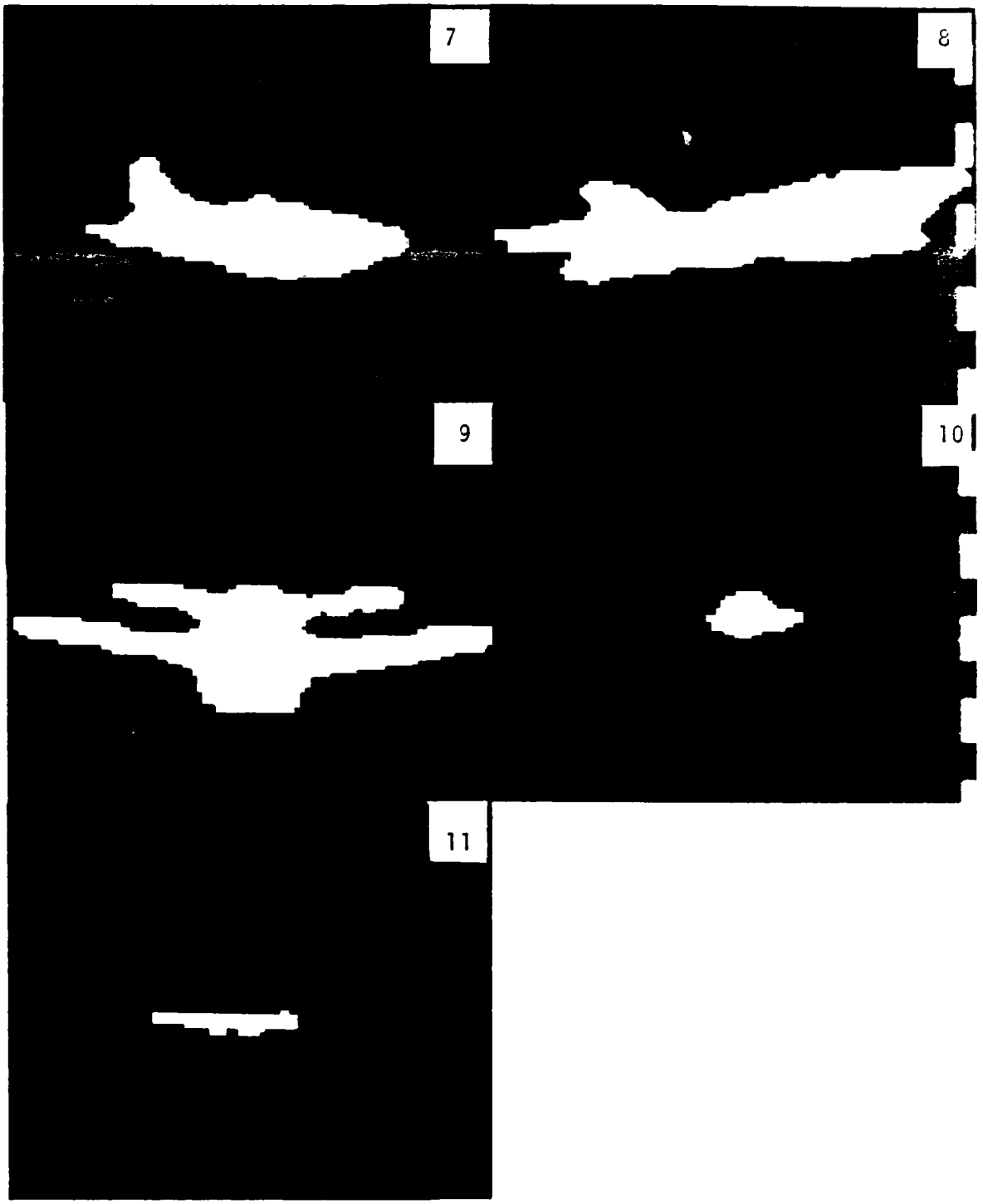


Figure 2: Continued

and were omitted from these studies. The eight remaining groups were represented by six pitch/yaw combinations for classification purposes, since groups one and four differ only in distance from groups two and five respectively, and since the classification is invariant to size change.

The groups depicting symmetric views (groups 4, 9, 10) are, apparantlv, slightly nonsymmetric. It may be that a bias exists in the ımaging system and should be compensated for in some way. Our experiments did not explicitly include such compensation.

Synthesis of Imagery

As explained above, eleven groups of twenty images were selected from the NRL data, representing all the distinct pitch/yaw/distance combinations. Neglecting distance, these eleven groups are characterized by six pitch/yaw combinations. For classification purposes, it was desired to obtain an "ideal" representation for each of the pitch/yaw groups. Accordingly, as an approximation to these images, one representative image was constructed for each orientation as an optical (and subjective) best fit to the data. Pictures of the synthesized imagery appears in Figure 3.

Unlike the real data, the synthesized images from groups 4, 9, and 10 are perfectly symmetric, since they are intended to depict pictures of symmetric objects. The features from these six images were used to characterize the six pitch/yaw combinations for classification experiments.

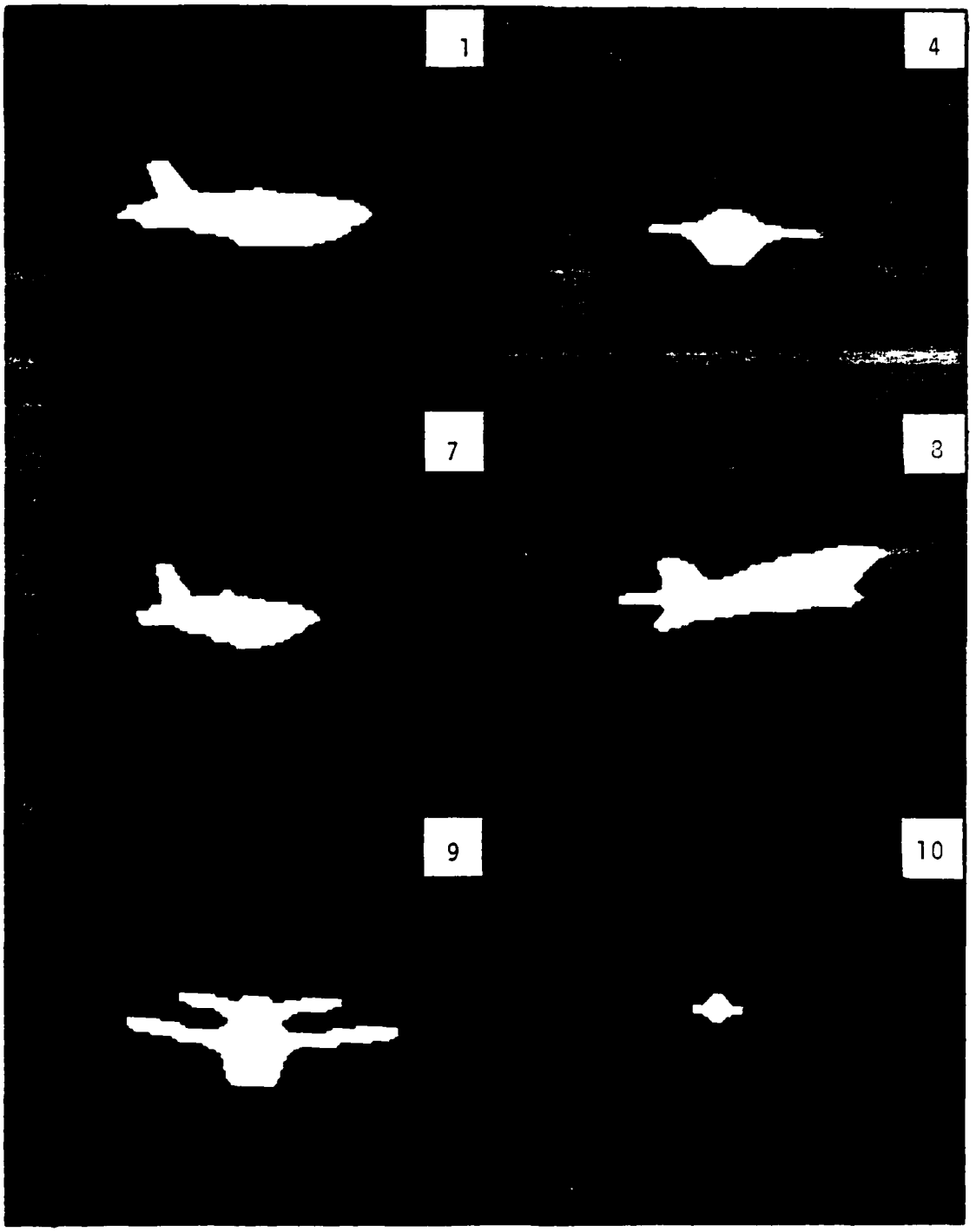


Figure 3: Symmetric drone images, group numbers in upper right corner.

Feature Extraction

Feature extraction reduces the total information contained in an image to a small number of parameters. These parameters are insensitive to change in size, rotation and translation, and are easily incorporated into a classification scheme. Normalized Fourier descriptors (NFD's) and invariant moments (IM's) were the features used to classify the NRL data. The algorithms are briefly described below.

Normalized Fourier Descriptors

The Fourier analysis, and the associated preprocessing, are as follows:

(1) Preprocessing consists of threshold and median filter.

(2) Following the median filter, a chain code representation of the object outline is constructed. The chain code representation constitutes an intricate piece of software, because it must be capable of handling a number of exceptions, e.g., isolated pixels, object extensions only one pixel wide, objects with embedded "holes", etc. The ideal would be the usage of preprocessing algorithms that would eliminate such anomalies. However, no combination of preprocessing steps has been found to entirely eliminate s ch anomalies under the variety of of combinations of noise and atmospheric turbulence. An alternative, which has been used for the purposes of studying the Fourier descriptor behavior, has been to use

the interactive features of our image display system to "prune" such anomalies by manual action. Approximately seven of the 160 processed NRL images required such action.

(3) Following the chain-coding, the chain-coded elements are interpolated to create an exact power of two number of code elements for Fourier analysis. The FFT is then used and normalized descriptors are calculated.

(4) Variation of the number of retained Fourier coefficients from N=16 to N=512 was carried out. From these variations it was established that a small number of coefficients, e.g. $16 \leq N \leq 64$, are sufficient to produce a good representation of the object outline.

The NFD's normally constitute an excellent characterization of the object. However, in the presence of noise, there is a possibility of an irremediable error in the normalization calculation. Noise in the original image causes perturbations in the corresponding Fourier coefficients. In particular, when the coefficients are of small magnitude, the noise can cause a sign change in the real part of those coefficients. Since the normalization criterion is to maximize:

$$\sum_k \mathrm{Re}\{F(k)\}|F(k)|$$

normalization errors can result. For group 9, this feature error was significant and caused significant classification error. The literature contains no mention of this problem.

Invariant Moments

The analysis by invariant moments, and the associated preprocessing are as follows:

(1) Preprocessing consists of threshold and median filter as in the NFD calculations.

(2) Following the median filter, the edges of the object are determined as in the boundary-tracing algorithm associated with the NFD's (see Final Report, 1982).

(3) The seven IM's of the object and the object boundaries are computed, giving fourteen numbers invariant under size change, rotation and translation. The inclusion of the seven moments of the edges is to insure good description of objects with prominent edges, but which occupy little area.

Classification Libraries

A classification library is a set of features computed from objects known a-priori. Given a piece of data, the solution of the classification problem is to minimize a criterion function with respect to the elements of the library. The nearest-neighbor criterion gives a simple and useful example of such a function.

The objects in the NRL data are the noisy two dimensional images of the drone at various orientations. Since no perfect images were available to construct a classification library, two alternatives were pursued. The first was to use the features

of the six synthesized drone images as a library. These features correspond to the noiseless imaging case and are referred to as the ideal library.

The second alternative was to construct an "average-data" library which likewise contained six elements but now each element is the group average of the features. It was reasoned that such a library should be more accurate in the presence of noise and would partially compensate for any bias in the imaging system. Both libraries were used in the classification experiments.

Classification

Classification based on a nearest neighbor criterion is the simplest and was considered first. Let N be the dimension of the features. For IM's, N=14, while for NFD's, N=512. Denote the feature vector of a piece of data by

$$D = \{d_i\}_{i=1}^{N}$$

and the feature of the Mth class of a library by

$$C^M = \{C_i^M\}_{i=1}^{N}$$

The classification of the data is accomplished by determining that value of j for which:

$||C^j - D||_p \leq ||C^k - D||_p$ for all k. The data is then "closest to" and is considered to be a member of class j. $||\cdot||_p$ is the ℓ_p norm given by

$$||X||_p = \sum_{i=1}^{N} |X|^p \quad p_{real}, \quad p \geq 1.$$

This norm with p = 1,2 was used to classify the NRL data. Classification by NFD's is described first.

Nearest Neighbor Classification by NFD's

Nearest Neighbor Classification by NFD's was carried out with 16, 32 and 64 retained coefficients. The results are presented in the "confusion matrices" of Figures 4 - 9. The matrices demonstrate that the classification is very insensitive to change in the number of retained coefficients for $16 \leq N \leq 64$. In fact, there is almost no advantage in choosing more than 16 coefficients. It is therefore clear that N=16 is the best choice for economy of representation.

For the NRL data, the use of the ℓ^2 norm resulted in fewer errors as can be seen from the following table derived from Figures 4 - 9:

Norm	Library	%Error	
ℓ^1	averaged data	32	
ℓ^1	ideal	29	
		31	average error
ℓ^2	averaged data	31	
ℓ^2	ideal	24	
		28	average error

Table 1: Average Error for NFD Classification.

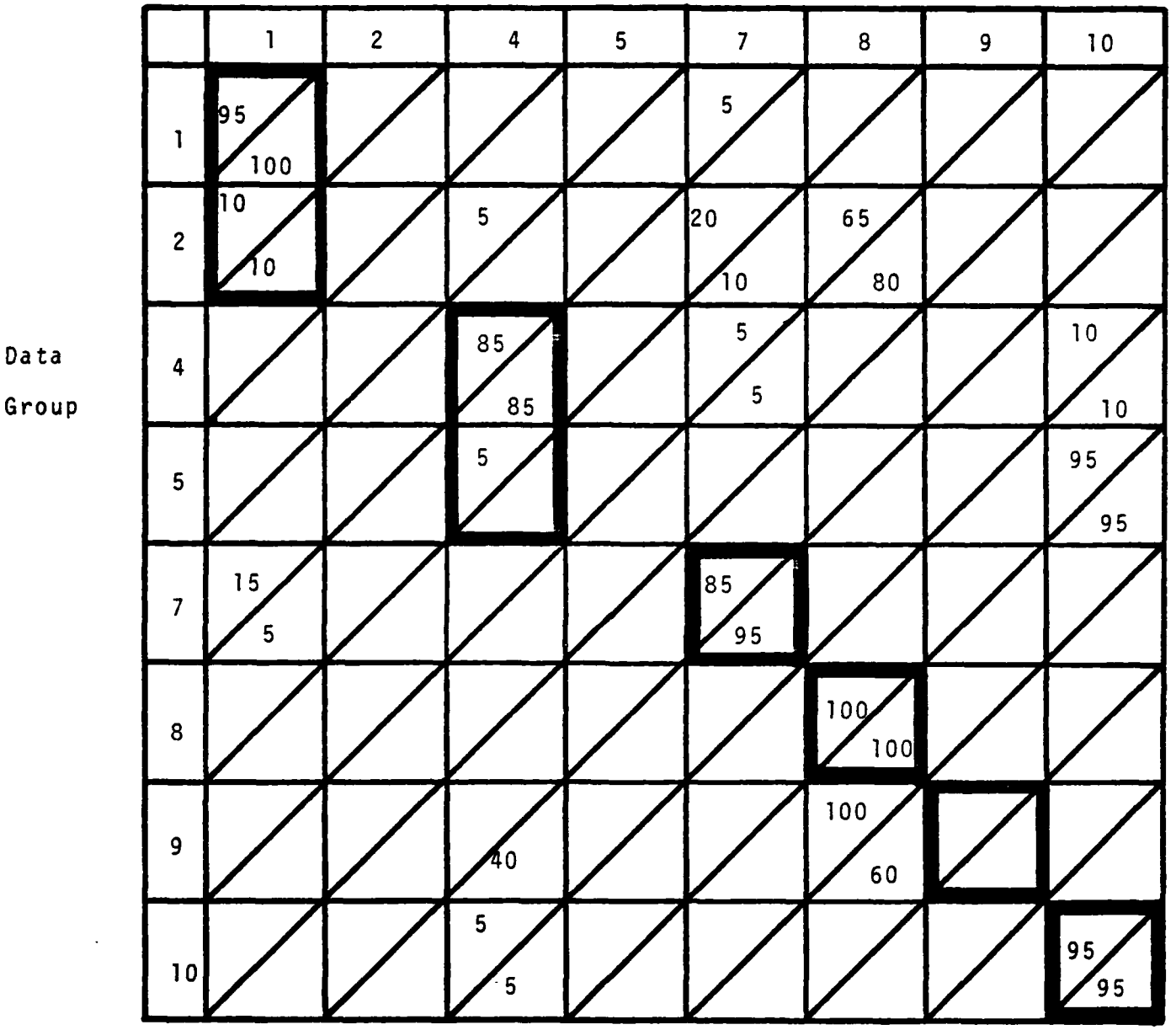


Figure 4: Nearest Neighbor NFD Classification with averaged-data library and 16 retained coefficients. Upper = ℓ^1 norm, lower = ℓ^2 norm. Heavy outline represents correct decision.

Classified Group (%)

Data Group

	1	2	4	5	7	8	9	10
1	95 / 100				5			
2	15 / 10		5		35 / 20	45 / 70		
4			85 / 85			5 / 5		
5			5	5				95 / 95
7	15 / 5				85 / 95			
8						100 / 100		
9			40				100 / 60	
10			5 / 5					95 / 95

Figure 5: Nearest Neighbor NFD Classification with averaged-data library and 32 retained coefficients. Upper = ℓ^1 norm, lower = ℓ^2 norm. Heavy outline indicates correct decision.

Classified Group (%)

Data Group	1	2	4	5	7	8	9	10
1	95 / 100				5			
2	15 / 10		5		35 / 20	45 / 70		
4			85 / 85			5 / 5		
5			5	5				95 / 95
7	15 / 5				85 / 95			
8						100 / 100		
9			40				100 / 60	
10			5 / 5					95 / 95

Figure 6: Nearest Neighbor NFD Classification with averaged-data library and 64 retained coefficients. Upper = ℓ^1 norm, lower = ℓ^2 norm. Heavy outline represents correct decision.

Classified Group (%)

Data Group	1	2	4	5	7	8	9	10
1	95 / 100				5			
2	50 / 75		10		35 / 15	5 / 10		
4			75 / 75			5 / 5		20 / 20
5					5 / 5			95 / 95
7	15 / 10				85 / 90			
8	5 / 5					95 / 95		
9	60 / 60						40 / 40	
10			5 / 5					95 / 95

Figure 7: Nearest Neighbor NFD Classification using ideal library and 16 retained coefficients. Upper = ℓ^1 norm, lower = ℓ^2 norm. Heavy outline represents correct decision.

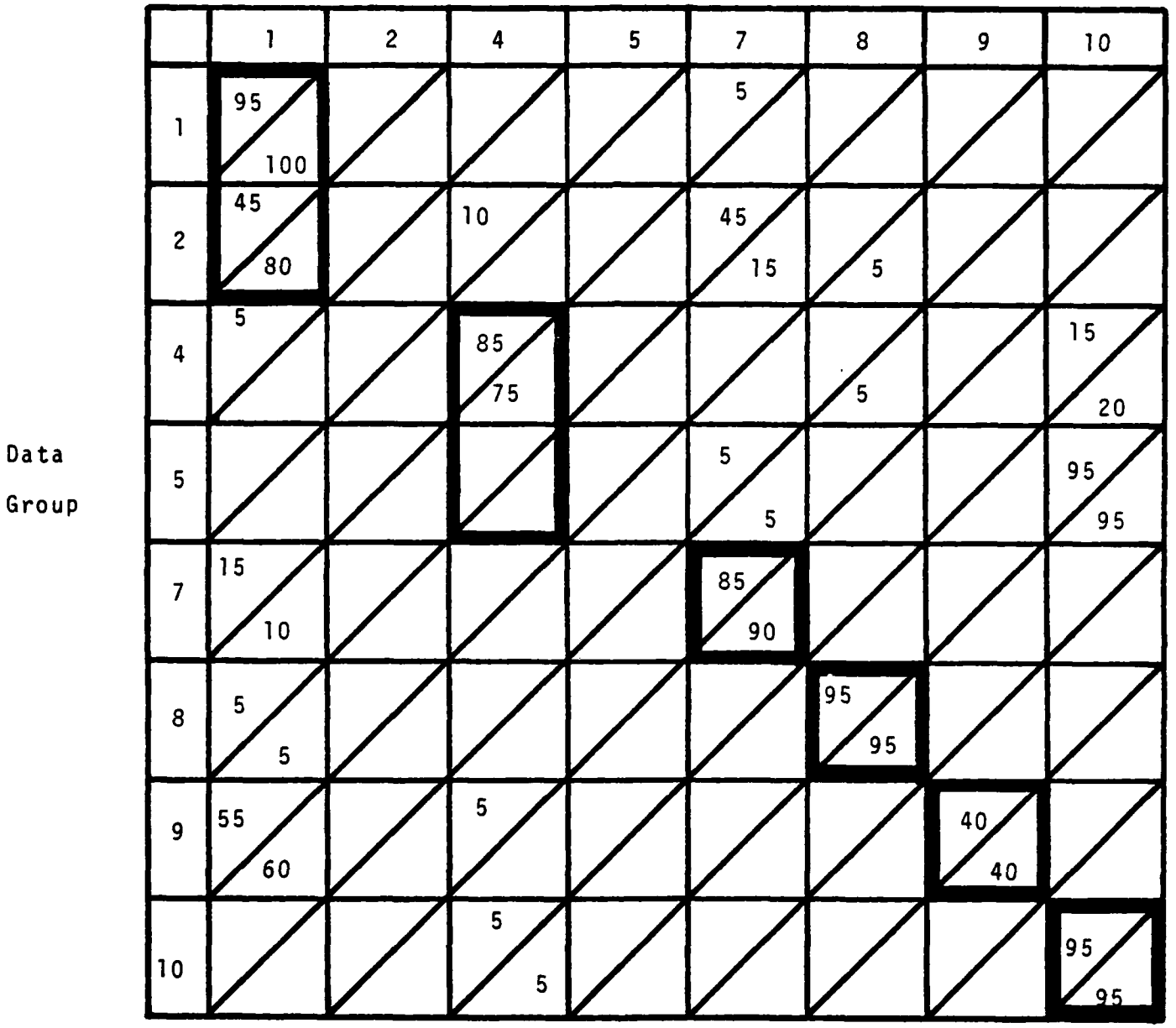


Figure 8: Nearest Neighbor NFD Classification with ideal library and 32 retained coefficients. Upper = ℓ^1 norm, lower = ℓ^2 norm. Heavy outline represents correct decision.

Classified Group (%)

Data Group

	1	2	4	5	7	8	8	10
1	95 / 100				5			
2	55 / 80		10		35 / 15	5		
4	5		80 / 75			5		15 / 20
5					5 / 5			95 / 95
7	15 / 10				85 / 90			
8	5 / 5					95 / 95		
9	55 / 60		5				40 / 40	
10			5 / 5					95 / 95

Figure 9: Nearest Neighbor NFD Classification with ideal library and 64 retained coefficients. Upper = ℓ^1 norm, lower = ℓ^2 norm. Heavy outline represents correct decision.

In compiling this table, group 9 was omitted to insure that normalization errors did not affect the choice of norm. Despite the 30% or so error listed in Table 1, the NFD Classification is an excellent one. Examination of the confusion matrices of Figures 4 - 9 and examination of the original images reveals that, while the classified group may not always be the correct one, the selected group always bears a strong resemblance to the correct one. That is, perturbations in the NFD coefficients correspond in a natural way to perturbations in shape.

It is interesting to note that use of the ideal library resulted in a lower percentage error than when the averaged-data library was used. Since a noise-free library may be the only one available in practice, this fact makes the NFD Classification attractive.

In summary, the confusion matrices of Figures 4 - 9 indicate the use of 16 retained coefficients with the ℓ^2 norm while Table 1 supports the use of an ideal noise-free library. The Nearest Neighbor NFD Classification gives excellent performance.

Nearest Neighbor Classification by IM's

The results of Nearest Neighbor Classification by IM are presented in the confusion matrices of Figures 10 - 11. For the NRL data, the use of the ℓ^1 norm resulted in slightly fewer errors, as examination of the following table reveals:

Norm	Library	% Error
ℓ^1	averaged data	41
ℓ^1	ideal	54
		48 average error
ℓ^2	averaged data	53
ℓ^2	ideal	59
		56 average

Table 2: Average Error for Nearest Neighbor IM Classification

However, use of the ℓ^1 norm improves the situation only slightly and the performance of the classification is disappointing overall. The average error is about twice as much as in the NFD classification with 16 retained coefficients. It is possible that the low performance of an IM classifier is due to large correlations between the moments, in which case the nearest-neighbor criterion is suboptimum. This question is taken up later.

Comparison of Nearest Neighbor Classification Schemes

The performance of the classification by NFD's was considerably higher than that by IM's, the NFD average classification error being about 1/2 that of the IM. Furthermore, the NFD classification was less affected by imaging system bias as can be seen from Tables 1 and 2. Table 1 indicates the NFD error is lower with the ideal (unbiased) library whereas Table 2

Classified Group (%)

Data Group	1	2	4	5	7	8	9	10
1	80 / 70				10 / 15	10 / 15		
2			10 / 25		5	20 / 10	65 / 65	
4	5		35		10	30	20	
5			5			35 / 30		65 / 65
7					100 / 95	5		
8	20 / 20		10 / 10		5 / 5	65 / 65		
9			5 / 5				95 / 95	
10					5		10	95 / 90

Figure 10: Nearest Neighbor IM Classification with average-data library. Upper = ℓ^1 norm, lower = ℓ^2 norm. Heavy outline represents correct decision.

Classified Group (%)

Data Group	1	2	4	5	7	8	9	10
1	30 / 25				10	70 / 65		
2			20 / 25			30 / 30	60 / 45	
4			40 / 25		5 / 5	35 / 35	20 / 35	
5			10 / 20			35 / 35	5 / 5	50 / 40
7	15 / 25				85 / 70	5		
8	5 / 15					95 / 85		
9			5 / 5			5 / 5	90 / 90	
10			30 / 10				50 / 75	20 / 15

Figure 11: Nearest Neighbor IM Classification with ideal library. Upper = ℓ^1 norm, lower = ℓ^2 norm. Heavy outline represents correct decision.

indicates IM error is lower with the averaged-data (biased) library. These facts make the NFD's the feature of choice over IM's for NN classification.

Maximum Likelihood Classification by IM's

In an attempt to improve the performance of the IM classification, a criterion was chosen to reduce the effect of moment correlations. Specifically, the approach may be stated as follows. The features of a piece of data are known to contain one of M library features C^i plus additive noise. The outcomes are assumed equally likely. That is, the ith hypothesis is:

$$H_i: \quad d = C^i + n \quad \text{where } d, C^i, n \text{ are}$$

N dimensional vectors, the C^i are equally likely and n, the noise, is zero mean, and Gaussian with covariance matrix R_n. Given a piece of data D, the optimal classification group C^i minimizes

$$(D - C^i)^T R_n^{-1} (D - C^i)$$

over all elements of the classification library [1]. If the inverse does not exist, a pseudoinverse must be used. Numerically, a pseudoinverse makes sense even if the matrix is only ill-conditioned.

The global covariance matrix of the IM data features was computed and is listed in Figure 12 (the matrix is symmetric). This matrix was computed as follows:

(1) The group mean was removed from each of the eight data groups to yield a zero mean vector sequence

$\{n^k\}_{k=1}^{160}$ (160 is the number of pieces of data).

(2) The ijth element R_{ij} of the covariance matrix was calculated using the formula:

$$R_{ij} = \frac{1}{159} \sum_{k=1}^{160} n_i^k \, n_j^k \quad .$$

It can be seen by inspection that R_n is nearly singular, hence ill-conditioned. Accordingly, the Moore-Penrose pseudoinverse was used for the required inversion. The pseudoinvers is listed in Figure 13 and the maximum likelihood classification based on it is summarized in Figure 14.

The performance is only trivially better than the Nearest Neighbor IM Classification. Table 3 presents the average error.

Library	% Error
average	36%
ideal	57%
	46% average error

Table 3: Average Error of Maximum Likelihood IM Classification.

This 46% is to be compared with the 48% from Table 2, and is still far above the 28% listed in Table 1. Thus no IM classification is nearly as good as the Nearest Neighbor NFD Classification with 16 retained coefficients.

The probable reason for the poor results of Figure 14 and Table 3 is that the noise is not stationary. Specifically, the

.018	.057	.100	.134	.255	.165	.255	.017	.057	.085	.107	.211	.149	.209
	.187	.325	.441	.835	.544	.833	.054	.191	.279	.351	.687	.456	.688
		.721	.922	1.786	1.117	1.761	.103	.348	.623	.717	1.426	.918	1.432
			1.366	2.581	1.647	2.533	.136	.475	.816	.995	1.952	1.281	1.948
				4.960	3.129	4.796	.259	.896	1.567	1.866	3.676	2.402	3.674
					2.057	3.048	.167	.584	.985	1.200	2.358	1.547	2.343
						4.827	.259	.895	1.560	1.878	3.688	2.406	3.691
							.018	.058	.093	.118	.233	.152	.230
								.206	.315	.400	.782	.516	.784
									.657	.766	1.531	.959	1.506
										1.050	2.044	1.310	1.995
											4.081	2.582	3.894
												1.697	2.490
													3.924

Figure 12: Covariance Matrix of the Data Features.

6043	-2217	-54.91	-129.0	49.46	21.57	-4.54	4774	1783	96.23	-59.36	18.94	13.91	-14.35
	959.9	6.01	44.48	-15.10	-8.20	0.34	1763	-803.4	-24.07	24.04	-10.13	-3.81	10.78
		31.35	16.50	-11.29	2.69	-4.99	-5.01	-5.58	-11.86	8.10	0.40	-0.53	-2.33
			101.1	-31.51	-10.93	-18.53	161.1	-55.78	2.99	-6.10	0.76	-3.63	1.51
				20.90	-5.47	1.55	-52.95	22.53	-1.15	3.78	0.29	-0.33	-0.94
					16.83	0.61	-18.12	-1.59	-0.57	-2.12	-0.76	-1.06	1.56
						10.07	-5.95	2.40	-1.26	-1.03	0.50	0.84	-0.24
							5007	-1704	-56.43	4.09	-35.48	1.99	33.37
								789.7	16.50	-9.31	16.48	-7.02	-18.00
									28.49	3.45	-8.76	5.94	-3.89
										82.53	-18.18	-12.18	-18.19
											16.56	-9.15	0.72
												23.09	0.44
													10.79

Figure 13: Pseudoinverse of the Covariance Matrix.

Classified Group (%)

Data Group	1	2	4	5	7	8	9	10
1	85 / 40				15 / 15	15 / 45		
2			50 / 35			10 / 10	50 / 55	
4	10 / 10		50 / 50		15 / 10	15 / 15	20 / 25	5 / 5
5			35 / 35		5 / 5	25 / 25	10 / 35	50 / 5
7	5 / 30				90 / 30	5 / 40		
8	25 / 15		10 / 10			65 / 75		
9			5 / 5				95 / 95	
10			5 / 10				60	95 / 20

Figure 14: Maximum likelihood IM Classification. Upper = average-data library, lower = ideal library. Heavy outline represents a correct decision.

features of some groups exhibited much more variation than others. The covariance matrix is a global measure, however, and cannot account for this nonstationarity.

Conclusions

The preprocessing, feature extraction and classification algorithms have been applied to classify eight groups of twenty images taken from the NRL data. The features used in the classification experiments were the normalized Fourier descriptors (NFD's) and the invariant moments (IM's). Use was made of two classification libraries: the ideal library comprised of features derived from synthesized imagery and the averaged-data library consisting of the group-averaged features. The Nearest Neighbor-NFD classification with 16 retained coefficients was the best overall performer in terms of accuracy and economy of representation. The best IM classification was only 1/2 as accurate as the sixteen coefficient NFD, in an average sense.

NFD's are normally an excellent characterization of an object. However, they are subject to two problems: noise in the original image can cause normalization errors, and the algorithm requires a sophisticated boundary-tracing subalgorithm. Both of these topics merit more research. These facts notwithstanding, the NFD's are well-suited to classification of aircraft in turbulent atmosphere.

References

[1] A. D. Whalen, Detection of Signals in Noise, Academic Press, New York, 1971.

Appendix - Software Listing

(1) Contr: Boundary-tracing routine.

(2) Interp: Interpolates a sequence to a power of two.

(3) FD: Normalizes the Four descriptors.

(4) MOMNT: Calculates the seven invariant moments of an image.

```
      PROGRAM CONTUR
C
      REAL A(512),B(512),C(512)
      INTEGER WF,IA(4),IADU(5)
      INTEGER IX1(512),IX2(512),IL(512)
      DATA IA/'80/
      INTEGER FCB(6)
C
      TYPE 100
 100  FORMAT(11X,'ENTER THE INPUT WORKFILE NUMBER .. '$)
      ACCEPT *,WF
C
C   SET UP THE THRESHOLDS
C
      THRESH = 111
      THR = 111
C
C   SET UP SADIE
C
      CALL SADINI
      CALL GETAPT
C
C  OPEN FILES
C
      CALL PRE(WF,1,NMSAMP,NMLINE)
C
C   SET UP STUFF FOR THE GRADIENT WINDOW
C
      VAL1=0
      VAL2=0
      VALUE=0
      KOUNT=1
      LINE = 2
      FLAG = 0
      CALL GETLIN(WF,A,IEOF)
      CALL GETLIN(WF,B,IEOF)
 01   CALL GETLIN(WF,C,IEOF)
C
C   EXTRACT EDGES AS LINE SEGMENTS DESCRIBED
C   BY TWO ENDPOINTS AND THE LINE NUMBER: IX1, IX2, IL
C
      DO 201 I=2,NMSAMP-1
      DV=B(I)-B(I+1)
      DV1=B(I)-B(I-1)
      DH=B(I)-C(I)
      DH1=B(I)-A(I)
      VAL1=SQRT(DV**2+DH**2)
      VAL2=SQRT(DV1**2+DH1**2)
      VALUE=(VAL1+VAL2)/2.
      IF(VALUE.LT.THRESH.OR.B(I).LT.THR) GO TO 210
      IF(FLAG.NE.0) GO TO 201
      IXI = I
      IXF = I
      IY = LINE
      FLAG = 1
      GO TO 201
210   IF(FLAG.NE.1) GO TO 201
      IXF = I-1
      FLAG = 0
      IX1(KOUNT) = IXI
      IX2(KOUNT) = IXF
      IL(KOUNT) = IY
```

```
201     CONTINUE
        IF(LINE.GE.NMLINE)GO TO 203
        DO 202 J=1,NMSAMP
        C(J)=B(J)
        B(J)=C(J)
202     CONTINUE
        LINE = LINE + 1
        GO TO 101
203     CONTINUE
C
C   CLOSE THE WORKFILE
C
        CALL POST(WF)
        KOUNT = KOUNT - 1
C
C   OPEN THE LINE ADJACIENCY SCRATCHFILE
C
        OPEN(UNIT=11,NAME='LAG',TYPE='SCRATCH',ACCESS='DIRECT',RECORDSIZE=2)
        DO 399 K = 1,KOUNT
        WRITE(11'K)(IA(J),J=1,4)
        C(K) = 0
399     CONTINUE
C
C   DETERMINE WHICH LINE SEGMENTS ARE ADJACIENT, STORING THE
C   RESPECTIVE KTH LINE ADJACIENCIES IN THE KTH RECORD
C
        DO 400 N = 2,KOUNT
        DO 455 J = 1,4
        IADJ(J) = 0
455     CONTINUE
        ISTART = MAX0(1,(N-12))
        NUM = 0
        DO 401 K = ISTART,N-1
        IF(ABS(IL(K)-IL(N)).GT.1.OR.IL(K).EQ.IL(N)) GO TO 401
        IF(IX2(K).LT.(IX1(N)-1).OR.IX1(K).GT.(IX2(N)+1)) GO TO 401
        READ(11'K) (IA(J),J=1,4)
        NUM = NUM + 1
        DO 403 J = 1,4
        IF(IA(J).NE.0) GO TO 403
        IA(J) = N
        IADJ(NUM) = K
        GO TO 405
403     CONTINUE
405     CONTINUE
        WRITE(11'K) (IA(J),J=1,4)
        WRITE(11'N) (IADJ(J),J=1,4)
401     CONTINUE
400     CONTINUE
C
C   START THE CONTOUR TRACING ALGORITHM
C
        NUM = 1
C
C   DETERMINE WHICH SEGMENTS ARE ADJACIENT TO THE FIRST ONE
C
        READ(11'1)(IA(J),J=1,4)
C
C   DETERMINE THE DIRECTION OF THE FIRST LINE SEGMENT
C
        II = IA(1)
        ID1 MIN0(IX1(1)-IX1(II),IX1(1)-IX2(II))
        ID2=MIN0(IX2(1)-IX1(II),IX2(1)-IX2(II))
        IXX1 = IX1(1)
        IXX2 = IX2(1)
        ILL = IL(1)
```

```
      IIN = I
      IF(ID1.GE.ID2) GO TO 601
      IXT = IXX1
      IXX1 = IXX2
      IXX2 = IXT
      INC = -1
601   CONTINUE
C
C   LIST THE CURRENT-LINE POINTS IN THE (X,Y) ARRAYS CONR, CONC
C
      DO 602 K = IXX1,IXX2,INC
      A(NUM) = FLOAT(K)
      B(NUM) = FLOAT(ILL)
      NUM1 = NUM
      NUM = NUM + 1
602   CONTINUE
C
C   SEE IF THE ENTIRE CONTOUR HAS BEEN TRACED
C
      AT = ABS(A(1)-A(NUM1))
      BT = ABS(B(1)-B(NUM1))
C
C   IF NOT, LOOK FOR SOMETHING ELSE TO DO
C
      IF(AT.LE.1.AND.BT.LE.1.AND.NUM.GT.4) GO TO 698
      READ(11'IIN)(IA(J),J=1,4)
C
C   SET THE IIN'TH OLD FLAG
C
      C(IIN) = 1
      ID = 1000
C
C   LOOK FOR THE CLOSEST NEW LINE SEGMENT WITH RESPECTIVE DIRECTION
C
      DO 603 K = 1,4
      IINT = IA(K)
      IF(IINT.EQ.0) GO TO 603
      IF(C(IINT).NE.0) GO TO 603
      ID1=ABS(INT(A(NUM1))-IX1(IINT))
      ID2=ABS(INT(A(NUM1))-IX2(IINT))
      IF(ID1.GE.ID.AND.ID2.GE.ID) GO TO 603
      ILL = IL(IINT)
      IIN = IINT
      IF(ID1.GT.ID2) GO TO 605
      IXX1 = IX1(IINT)
      IXX2 = IX2(IINT)
      ID = ID1
      INC = 1
      GO TO 603
605   CONTINUE
      IXX1 = IX2(IINT)
      IXX2 = IX1(IINT)
      ID = ID2
      INC = -1
603   CONTINUE
C
C   IF NO NEW SEGMENTS TO BE FOUND, BRANCH TO TRACEBACK SECTION
C
      IF(ID.EQ.1000) GO TO 799
      NUML = NUM1 - 1
C
C GO TO TRACEBACK SECTION IF THE SEGMENTS ARE PARTIALLY NONOVERLAPPING
C
      IF(ABS(INT(A(NUM1))-IXX1).GT.1) GO TO 681
```

```
C
      GO TO 601
C
C   SEARCH FOR A NEW SEGMENT
C
 799  CONTINUE
      NEW = 0
      DO 679 K = 1,KOUNT
      IF(C(K).NE.0) GO TO 679
      NEW = 1
      GO TO 680
 679  CONTINUE
 680  CONTINUE
C
C   IF NO NEW SEGMENTS, GO TO  END SECTION
C
      IF(NEW.NE.1) GO TO 911
      NUML = NUM1 - 1
C
C    TRACEBACK SECTION
C
 681  CONTINUE
      A(NUM) = A(NUML)
      B(NUM) = B(NUML)
      NUM1 = NUM
      NUM = NUM + 1
      NUML = NUML - 1
      ID = 1000
C
C SEE IF THERE'S NOW AN ADJACIENT SEGMENT TO THE LAST LISTED POINT
C
      DO 690 K = 1,KOUNT
      IF(C(K).NE.0) GO TO 690
      ID1 = ABS(INT(A(NUM1))-IX1(K))
      ID2 = ABS(INT(A(NUM1))-IX2(K))
      ID3 = ABS(INT(B(NUM1))-IL(K))
      IF((ID1.GT.1.AND.ID2.GT.1).OR.ID3.GT.1) GO TO 690
      IF(ID1.GE.ID.AND.ID2.GE.ID) GO TO 690
C
C   DETERMINE CLOSEST NEW ADJACIENT SEGMENT WITH DIRECTION
C
      IIN = K
      ILL = IL(K)
      IF(ID1.GT.ID2) GO TO 691
      IXX1 = IX1(K)
      IXX2 = IX2(K)
      INC = 1
      ID = ID1
      GO TO 690
 691  CONTINUE
      IXX1 = IX2(K)
      IXX2 = IX1(K)
      INC = -1
      ID = ID2
 690  CONTINUE
C
C   NEED TO TRACEBACK MORE ?
C
      IF(ID.EQ.1000) GO TO 681
      GO TO 601
 911  CONTINUE
C
C   END SECTION
C   DETERMINES IF A TRACEBACK IS NEEDED TO CLOSE THE CONTOUR
C
      DO 405 K = 1,NUM1
```

```
      IF(ABS(A(NUM1)-A(K)).GT.1.OR.ABS(B(NUM1)-B(K)).GT.1) GO TO 695
      M = K
      GO TO 696
695   CONTINUE

C     TRACEBACK TO BEGINNING

696   CONTINUE
      IF(M.EQ.1) GO TO 697
      DO 697 K = M,2,-1
      A(NUM) = A(K)
      B(NUM) = B(K)
      NUM1 = NUM
      NUM = NUM + 1
697   CONTINUE
698   CONTINUE

C     DELETE SCRATCHFILE

      CLOSE(UNIT=11)
      OPEN(UNIT=11,NAME='CONT.OUT')
      WRITE(11,511) NUM1
      DO 307 K = 1,NUM1
      WRITE(11,512) A(K),B(K)
307   CONTINUE
511   FORMAT(2X,I3)
512   FORMAT(2X,2F6.0)
      CLOSE(UNIT=11)

C     SET UP THINGS FOR THE IIS

      TYPE 510
      ACCEPT 977,ANS
      IF(ANS.NE.'Y') GO TO 999
      TYPE 500
      ACCEPT 977,ANS
977   FORMAT(A2)
      IF(ANS.EQ.'R') GO TO 913
      CALL GRINT(IX1)
      DO 121 K = 1,256
      IX1(K) = -1
121   CONTINUE
      NCHN = -32768
      TYPE 501
      ACCEPT *, NMASK
      GO TO 914
913   NMASK = 255
      TYPE 501
      ACCEPT *,NCHN
914   CONTINUE

C     SCALE THE COORDINATE FOR THE IIS

      DO 933 K = 1,NUM1
      A(K) = A(K) - NMSAMP/2 + 255
      B(K) = B(K) - NMSAMP/2 + 255
933   CONTINUE

C     WRITE THE CONTOUR TO IIS

      DO 939 K = 1,(NUM1-1)
      IX1 = INT(A(K))
      IY1 = INT(B(K))
      IX2 = INT(A(K+1))
      IY2 = INT(B(K+1))
```

```
 938     FORMAT(A2)
 939     CONTINUE
 941     CONTINUE
C
C    CLOSE THE CONTOUR
C
         IXX1 = INT(A(NUM1))
         IY1 = INT(B(NUM1))
         IXX2 = INT(A(1))
         IY2 = INT(B(1))
         CALL DVECT(FC8,IXX1,IY1,IXX2,IY2,NCHN,NMASK,IX1)
 999     CONTINUE
C
C    ALL DONE
C
         STOP
 300     FORMAT(11X,'DO YOU WANT GRAPHICS OR REFRESH DISPLAY ? (G OR R) ',$)
 301     FORMAT(11X,'WHAT COLOR ? (1=RED, 2=GREEN, 4=BLUE) ',$)
 305     FORMAT(11X,'GO ? ',$)
 310     FORMAT(11X,'DO YOU WANT DISPLAY ? ',$)
         END
```

```
      SUBROUTINE INTERP(CHNR,CHNC)
      DIMENSION CHNR(1024),CHNC(1024),RTEMP(1024),CTEMP(1024)
      DIMENSION NAME(15)
      DATA NAME/'CO','NT','UR','.O','UT','$',9*0/
      SQ2 = SQRT(2.)
C
C ***********************************************************************
C
C ***** OPEN OLD FILE AND READ THE CHAIN CODE VALUES.
C *****       NTOT= TOTAL # OF POINTS
C *****       CHNR(I)= REAL PORTION OF ITH CHAIN CODE
C *****       CHNC(I)= IMAGINARY PORTION OF ITH CHAIN CODE
C
      TYPE 11
 11   FORMAT(11X,'ENTERING INTERP')
      TYPE 12
 12   FORMAT(11X,'ENTER THE INPUT VERSION NUMBER .. ',$)
      ACCEPT 13,NAME(7)
 13   FORMAT(A2)
      OPEN(UNIT=11,NAME=NAME,TYPE='OLD')
      READ(11,1),NTOT
      DO 10 I=1,NTOT
      READ(11,2)RTEMP(I),CTEMP(I)
 10   CONTINUE
 1    FORMAT(2X,I3)
 2    FORMAT(2X,2F6.0)
      CLOSE(UNIT=11)
      K=1
C
C ***** DETERMINE CONTOUR LENGTH AND DISTANCE BETWEEN POINTS
C
      CONLEN=0
      DO 25 I=2,NTOT
      RINC = 1
      R1=RTEMP(I)-RTEMP(I-1)
      C1=CTEMP(I)-CTEMP(I-1)
      IF(R1.NE.0.AND.C1.NE.0) RINC = SQ2
      CONLEN=CONLEN+RINC
 25   CONTINUE
      RINC = 1
      R1=RTEMP(1)-RTEMP(NTOT)
      C1=CTEMP(1)-CTEMP(NTOT)
      IF(R1.NE.0.AND.C1.NE.0) RINC = SQ2
      CONLEN=CONLEN+RINC
      DIV=CONLEN/512.
C
C ***** STARTING PROCEDURE
C
      PLEN = 1
      RL=RTEMP(1)-RTEMP(NTOT)
      CL=CTEMP(1)-CTEMP(NTOT)
      IF(RL.NE.0.AND.CL.NE.0) PLEN = SQ2
 30   RMULT=K*DIV
      IF(RMULT.GT.PLEN)GO TO 40
      FACT=RMULT/PLEN
      CHNR(K)=RTEMP(NTOT)+FACT*RL
      CHNC(K)=CTEMP(NTOT)+FACT*CL
      K=K+1
      GO TO 30
 40   CONTINUE
C
```

```
      ISAVE=1
      R1=RTEMP(1)-RTEMP(NTOT)
      C1=CTEMP(1)-CTEMP(NTOT)
      PPLEN = 1
      IF(R1.NE.0.AND.C1.NE.0) PPLEN = SQ2
 60   CONTINUE
      RL = RTEMP(ISAVE+1)-RTEMP(ISAVE)
      CL=CTEMP(ISAVE+1)-CTEMP(ISAVE)
      PLEN = 1
      IF(RL.NE.0.AND.CL.NE.0) PLEN = SQ2
 50   CONTINUE
      RMULT = K*DIV
      IF(K.GT.512) GO TO 90
      IF(RMULT.GT.(PPLEN+PLEN)) GO TO 85
      FACT = (RMULT - PPLEN)/PLEN
      CHNR(K) = RTEMP(ISAVE) + FACT*RL
      CHNC(K) = CTEMP(ISAVE) + FACT*CL
      K=K+1
      GO TO 50
 85   ISAVE=ISAVE+1
      PPLEN = PPLEN + PLEN
      GO TO 60
 90   CONTINUE
      TYPE 14
 14   FORMAT(11X,'LEAVING INTERP')
      RETURN
      END
```

```
      SUBROUTINE MOMNTS(IWF,Q)

C  MOMNTS COMPUTES THE INVARIANT MOMENTS OF A SQUARE
C  WORKFILE IWF.  THE MOMENTS ARE RETURNED IN THE
C  SEVEN DIMENSIONAL ARRAY Q.

      DIMENSION A(512)
      DOUBLE PRECISION XM,YM,MU,S,D,F,G
      DOUBLE PRECISION M(0:3,0:3),E(0:3,0:3),U(0:3,0:3),Q(7)
      DO 23 I=0,3
      DO 23 J=0,3
      M(I,J)=0.
3     CONTINUE
      TYPE 5555
5555  FORMAT(11X,'ENTERING MOMNTS ')

C  PREPARE TO READ WORKFILE IWF

      CALL PRE(IWF,1,N,L)

C  COMPUTE MOMENTS M UP THROUGH ORDER THREE

      DO 4 Y=1,L
      CALL GETLIN(IWF,A,IEOF)
      DO 3 X=1,N
      IF(A(X).EQ.0)GO TO 3
      DO 2 I=0,3
      DO 1 J=0,3
      IF(I+J.GT.3)GO TO 1
      M(I,J)=M(I,J)+(DBLE(X))**I*(DBLE(Y))**J*DBLE(A(X))
1     CONTINUE
2     CONTINUE
3     CONTINUE
4     CONTINUE

      CLOSE WORKFILE

      CALL POST(IWF)

C     COMPUTE MEANS XM AND YM

      XM=M(1,0)/M(0,0)
      YM=M(0,1)/M(0,0)

C     COMPUTE CENTRAL MOMENTS U UP THROUGH ORDER THREE

      U(0,0)=M(0,0)
      U(0,1)=0
      U(1,0)=0
      U(2,0)=M(2,0)-XM*M(1,0)
      U(0,2)=M(0,2)-YM*M(0,1)
      U(1,1)=M(1,1)-XM*M(0,1)
      U(3,0)=M(3,0)-3*M(2,0)*XM+2*M(1,0)*XM**2
      U(0,3)=M(0,3)-3*M(0,2)*YM+2*M(0,1)*YM**2
      U(2,1)=M(2,1)-2*XM*M(1,1)-YM*M(2,0)+2*XM**2*M(0,1)
      U(1,2)=M(1,2)-2*YM*M(1,1)-XM*M(0,2)+2*YM**2*M(1,0)

C  COMPUTE THE NORMALIZED MOMENTS E(J,K) WHERE J+K = 2,3

      DO 6 J=0,3
      DO 5 K=0,3
      S=J+K
```

```
      R=(R/2.+1.)
      E(J,K)=U(J,K)/U(0,0)**R
5     CONTINUE
      CONTINUE

C     COMPUTE THE INVARIANT MOMENTS Q
C
      S=E(3,0)-3*E(1,2)
      D=3*E(2,1)-E(0,3)
      F=E(3,0)+E(1,2)
      G=E(2,1)+E(0,3)
      Q(1)=E(2,0)+E(0,2)
      Q(2)=(E(2,0)-E(0,2))**2+4*E(1,1)**2
      Q(3)=S**2+D**2
      Q(4)=F**2+G**2
      Q(5)=S*F*(F**2-3*G**2)+D*G*(3*F**2-G**2)
      Q(6)=(E(2,0)-E(0,2))*(F**2-G**2)+4*E(1,1)*F*G
      Q(7)=-S*F*(F**2-3*G**2)+D*G*(3*F**2-G**2)
C
C   TAKE LOG TO COMPRESS DYNAMIC RANGE
C
      DO 15 I=1,7
      IF(Q(I).EQ.0.)GO TO 15
      Q(I)=DLOG10(DABS(Q(I)))
15    CONTINUE
      TYPE *,' '
      TYPE 6666
6666  FORMAT(11X,'LEAVING MOMNTS ')
      RETURN
      END
```

```
      SUBROUTINE FD(CHNR,CHNC,NS)

C
C   CALCULATES THE FOURIER DESCRIPTOR OF A CONTOUR

      REAL CHNR(512),CHNC(512)
      REAL MXR,MIR,MXC,MIC
      REAL MR,MC
      REAL CHNTR(512),CHNTC(512),MAG1,MAG2,MAG
      DOUBLE PRECISION ARC,ARCT,SR,PI,PI2,U,V
      INTEGER FCB(6),BUFF(256)
      DATA PI/3.1415926535897932D0/
      PI2 = 2*PI
      DATA BUFF/256*-1/
      DATA N/512/
      TYPE *,'           ENTERING FD'

C
C   TAKE FFT

      CALL FFT2(CHNR,CHNC,N,-1)

C
C   SEE IF THE CONTOUR IS TRACED IN THE CLOCKWISE DIRECTION

      MAG1 = SQRT(CHNR(2)**2+CHNC(2)**2)
      MAG2 = SQRT(CHNR(N)**2+CHNC(N)**2)

C
C   IF NOT, REVERSE THE SEQUENCE
C
      IF(MAG1.GE.MAG2) GO TO 321
      DO 311 K = 2,N
      CHNTR(K) = CHNR(K)
      CHNTC(K) = CHNC(K)
311   CONTINUE
      DO 312 K = 2,N
      CHNR(K) = CHNTR(N+2-K)
      CHNC(K) = CHNTC(N+2-K)
312   CONTINUE
      MAG1 = MAG2
321   CONTINUE

C
C   MAG1 IS THE MAGNITUDE OF F(1), U IS THE PHASE OF F(1)
C
      U = ATAN(CHNC(2)/CHNR(2))
      IF(CHNR(2).LT.0) U = U - PI

C
C   NORMALIZATION

C   NORMALIZE POSITION

      CHNR(1) = 0
      CHNC(1) = 0


      MAG2 = 0
      DO 200 K = 2,N

C   DIVIDE BY THE MAGNITUDE OF THE LARGEST COEFFICIENT

      CHNR(K) = CHNR(K)/MAG1
      CHNC(K) = CHNC(K)/MAG1
      MAG = SQRT(CHNR(K)**2 + CHNC(K)**2)
```

```
C   DETERMINE THE COEFFICIENT WITH THE SECOND LARGEST MAGNITUDE
C
      IF(K.EQ.2.OR.MAG.LE.MAG2) GO TO 200
      MAG2 = MAG
      K2 = K
  200 CONTINUE
C
C   F(K2) IS THE COEFFICIENT OF SECOND LARGEST MAGNITUDE
C
C   U IS THE PHASE OF F(K2)
C
      U = ATAN(CHNC(K2)/CHNR(K2))
      IF(CHNR(K2).LT.0) U = U - PI
C
C   SHIFT K2 TO BE IN (-N/2+1,N/2)
C
      K2 = K2 - 1
      IF(K2.GT.N/2) K2 = K2 - N
C
C   CALCULATE THE NORMALIZATION MULTIPLICITY OF F(K2)
C
      MK2 = ABS(K2-1)
C
C   PERFORM THE FIRST OF THE MK2 NORMALIZATIONS
C
      DO 375 K = 2,N
      K1 = K - 1
      IF(K1.GT.N/2) K1 = K1 - N
      SR = ((K1-K2)*U + (1-K1)*U)/FLOAT(K2-1)
      CHNTR(1) = CHNR(K)*DCOS(SR) - CHNC(K)*DSIN(SR)
      CHNC(K) = CHNR(K)*DSIN(SR) + CHNC(K)*DCOS(SR)
      CHNR(K) = CHNTR(1)
      CHNTR(K) = CHNR(K)
      CHNTC(K) = CHNC(K)
  375 CONTINUE
C
C   CALULATE THE CORRESPONDING ABIGUITY RESOLVING CRITERION
C
      ARC = 0
      DO 583 K = 2,N
      ARC = ARC + CHNR(K)*ABS(CHNR(K))
  583 CONTINUE
C
C   PERFORM THE REST OF THE NORMALIZATIONS WITH THEIR RESPECTIVE ARCS
C
      IF(MK2.EQ.1) GO TO 901
      DO 586 NUM = 2,MK2
      ARCT = 0
      DO 584 K = 2,N
      K1 = K - 1
      IF(K1.GT.N/2) K1 = K1 - N
      SR = (K1-1)*PI2/FLOAT(MK2)
      SR= - SR
      CHNTT = CHNR(K)*DCOS(SR) - CHNC(K)*DSIN(SR)
      CHNC(K) = CHNR(K)*DSIN(SR) + CHNC(K)*DCOS(SR)
      CHNR(K) = CHNTT
      ARCT = ARCT + CHNR(K)*ABS(CHNR(K))
C
C   PICK THE NORMALIZATION WITH THE LARGEST ARC
C
  584 CONTINUE
      IF(ARCT.LE.ARC) GO TO 586
      DO 585 K = 2,N
      CHNTR(K) = CHNR(K)
      CHNTC(K) = CHNC(K)
  585 CONTINUE
```

```
      ...  ...
  586     CONTINUE
          DO 902 K = 2,N
          CHNR(K) = CHNTR(K)
          CHNC(K) = CHNTC(K)
  902     CONTINUE
C
C   FILTERING - IDEAL LOPASS
C
  901     IF((NS+1).GT.N) GO TO 555
          NC = NS/2 + 1
          ND = N - NS/2
          DO 1 K = NC,ND
          CHNR(K) = 0
          CHNC(K) = 0
    1     CONTINUE
C
C   TAKE INVERSE FFT OF THE FOURIER DESCRIPTOR
C
  555     CONTINUE
          OPEN(UNIT=11,NAME='FD.OUT',FORM='UNFORMATTED')
          DO 891 L = 1,512
          WRITE(11) CHNR(L),CHNC(L)
  891     CONTINUE
          CLOSE(UNIT=11)
          CALL FFT2(CHNR,CHNC,N,+1)
          DO 2 K = 1,N
          CHNR(K) = CHNR(K)/FLOAT(N)
          CHNC(K) = CHNC(K)/FLOAT(N)
    2     CONTINUE
C
          TYPE 502
          ACCEPT 503,ANS
          IF(ANS.NE.'Y') GO TO 333
C
C   SET UP THINGS FOR THE IIS
          N = 512
C
          TYPE 500
          ACCEPT 100,ANS
  100     FORMAT(A2)
          IF(ANS.EQ.'R') GO TO 201
          CALL GRINT(BUFF)
          DO 121 K = 1,256
          BUFF(K) = -1
  121     CONTINUE
          NCHN = -32768
          TYPE 501
          ACCEPT *, NMASK
          GO TO 202
  201     NMASK = 255
          TYPE 501
          ACCEPT *,NCHN
  202     CONTINUE
C   SCALE COORDINATES FOR IIS DISPLAY
C
          MXR = -1000
          MIR = 1000
          MXC = -1000
          MIC = 1000
          DO 601 K = 1,N
          MXR = MAX(MXR,CHNR(K))
          MIR = MIN(MIR,CHNR(K))
          MXC = MAX(MXC,CHNC(K))
          MIC = MIN(MIC,CHNC(K))
  601     CONTINUE
```

```
        MC = MXC - MIC
        MIR = MIN(MIR,MIC)
        SC = 412/MAX(MR,MC)
        DO 602 K = 1,N
        CHNR(K) = SC*(CHNR(K) - MIR) + 50
        CHNC(K) = SC*(CHNC(K) - MIR) + 50
 602    CONTINUE
C
C   PERFORM NEAREST NEIGHBOR INTERPOLATION
C
        DO 41 K = 1,N
        ADDIR = 0
        ADDIC = 0
        IF(INT(CHNR(K)).LT.INT(CHNR(K)+0.5)) ADDIR = 1
        IF(INT(CHNC(K)).LT.INT(CHNC(K)+0.5)) ADDIC = 1
        CHNR(K) = INT(CHNR(K)) + ADDIR
        CHNC(K) = INT(CHNC(K)) + ADDIC
 41     CONTINUE
C
C   WRITE THE CONTOUR TO IIS
C
        DO 939 K = 1,(N-1)
        IX1 = INT(CHNR(K))
        IY1 = INT(CHNC(K))
        IX2 = INT(CHNR(K+1))
        IY2 = INT(CHNC(K+1))
        CALL DVECT(FCB,IX1,IY1,IX2,IY2,NCHN,NMASK,BUFF)
 939    CONTINUE
        IX1 = INT(CHNR(N))
        IY1 = INT(CHNC(N))
        IX2 = INT(CHNR(1))
        IY2 = INT(CHNC(1))
        CALL DVECT(FCB,IX1,IY1,IX2,IY2,NCHN,NMASK,BUFF)
 333    TYPE *,'          LEAVING FD'
        RETURN
 500    FORMAT(11X,'DO YOU WANT GRAPHICS OR REFRESH DISPLAY ? (G OR R) ',$)
 501    FORMAT(11X,'WHAT COLOR ? (1=RED, 2=GREEN, 4=BLUE) ',$)
 502    FORMAT(11X,'DO YOU WANT DISPLAY ? ',$)
 503    FORMAT(A2)
        END
```

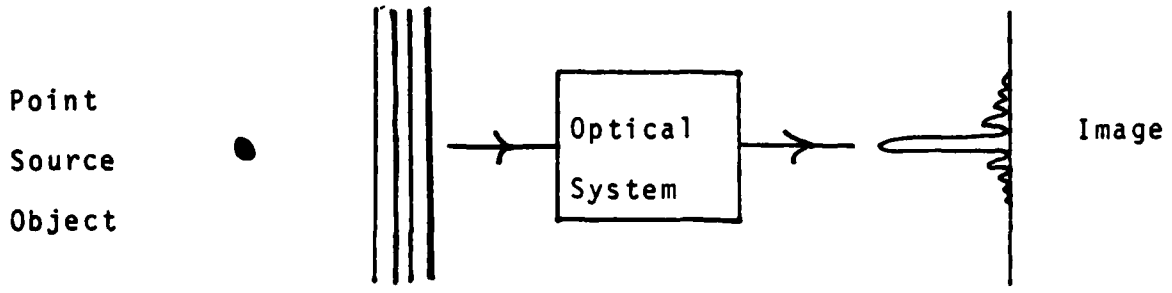
(. I) Atmospheric Turbulence

In this section, we will examine our recent work in the area of images degraded by the turbulent atmosphere. Our efforts have been concentrated in the areas of simulation of an atmospheric turbulence point spread function, restoration of turbulence degrad d images by the shift-and-add method, deblurring of shift-and-add images, and the effects of atmospheric turbulence on the invariant moments. It will be seen that the combination of shift-and-add processing deblurring substantially improves turbulence images.

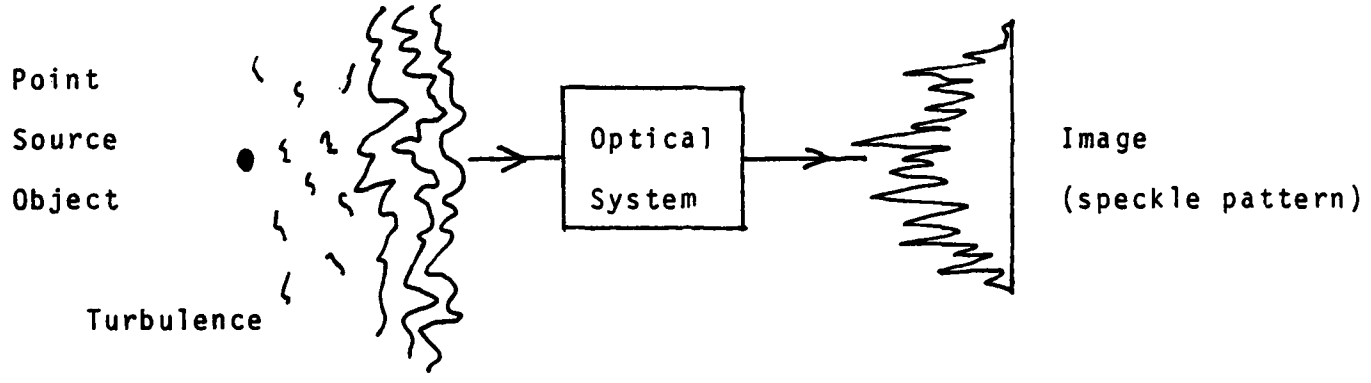
Background

The turbulent atmosphere degrades images of objects which are observed through it. These degradations are due to the passage of the optical wavefront through regions of the atmosphere having variable indices of refraction caused by inhomogeneities in temperature, density, and velocity. The variable indices of refraction introduce phase and amplitude shifts in the wavefront, although for relatively mild turbulence, the phase shifts are the dominent factor in image degradation. Phase shifts have the effect of distributing a point source such as a star into a speckle image as in Figure II.1.

We now consider an object f(x,y), which may or may not be a point source, and assume a mathematical model for turbulence degradations



(a) Imaging in the absence of turbulence degradations.



(b) Imaging tnrough the turbulent atmosphere.

Figure 1

$$S_m(x,y) = h_m(x,y)**f(x,y) + C_m(x,y) \qquad (1)$$

where $S_m(x,y)$ is the mth turbulence image, $h_m(x,y)$ is the mth turbulence point spread function (speckle pattern), and $C_m(x,y)$ is the mth contamination term encompassing all degradations such as system nonlinearities and noise. This model implicitly assumes an exposure sufficiently short that the atmosphere may be considered to be "frozen". It also assumes that the entire object is imaged through a single isoplanatic patch (region where the point spread function may be considered to be shift invariant). The next section presents our simulation of this model.

Simulation

In order to carry out the restoration and deblurring of atmospheric turbulence images, we first simulated the degraded images. In particular, we simulated the point spread functions h_m of equation (1) which were then convolved with our original undegraded images. At this point, no noise was added. Our simulation of the psf's h_m followed an algorithm developed by B. L. McLamery [1]. This particular algorithm assumes that the phase variations produced by the atmosphere are the dominant components of image degradation, and ignores any effects due to amplitude variations. We summarize this method as follows:

(1) Generate a complex array of Gaussian random numbers.

(2) Multiply this array by $f^{-11/6}$, the square root of the

Kolmogorov spatial power spectrum (which is typically used for atmospheric turbulence).

(3) Fourier transform the array

(4) Separate the complex result of this transform into its real and imaginary parts, each which is considered an instaneous phase map denoted $\phi(u,v)$.

(5) Choose a phase map $\phi(u,v)$ (from step 4) and compute the path length difference map

$$\ell(u,v) = \frac{\phi(u,v)\lambda}{2\pi}$$

where λ denotes wavelength.

(6) Compute the coherent point spread function

$$s(x,y) = F\left\{p(u,v)e^{\frac{i2\pi\ell(u,v)}{\lambda}}\right\}$$

where we have at this point set $p(u,v)\equiv 1$, and $F\{\cdot\}$ denotes Fourier transform.

(7) Finally compute the intensity point spread function $S(x,y)$ as

$$S(x,y) = |s(x,y)|^2.$$

This $S(x,y)$ is our hm of equation (1).

Using this method, we started at step (1) with arrays of Gaussian random numbers having standard deviations of 0.5, 1.5, 2.5, and 3.5 and generated a series of 20 psf's for each standard deviation. Each series of psf's was then convolved with an undegraded image to produce a series of atmospheric turbulence images on which the shift-and-add process could be tested. For

our original undegraded images, we chose a drone against a black (gray level 0) background. We chose three different gray levels for the drone : 64, 128 and 192. Also, because the shift-and-add processing requires a "brightest point" in the original object, a "glint" or point source at a gray level of 255 was inserted on the drone body.

Results of this simulation are depicted in Figures 2-4. Figure 2 shows the original drone at a gray level of 128. Figure 3 shows examples from the series of 20 atmospheric turbulence psf's. Finally, Figure 4 depicts the corresponding examples from the series of turbulence degraded images.

Restoration by Shift-and-Add Processing

Having simulated atmospheric turbulence images, we attempted restoration by the shift-and-add method developed by Bates and Cady [2,3]. We continue to assume the degradation model of equation (1)

$$S_m(x,y) = h_m(x,y)**f(x,y) + C_m(x,y)$$

where, as before, $S_m(x,y)$ denotes the mth turbulence image. We also assume that we have M of these images. The shift-and-add process is then simple to describe. Let (ξ_m,μ_m) denote the spatial coordinates at which the degraded image $S_m(x,y)$ takes on its maximum value. The shift-and-add image is the result of translating each $S_m(x,y)$ so that (ξ_m,μ_m) is at the origin of coordinates and summing the M translated images:

$$s(x,y) = \frac{1}{M} \sum_{m=1}^{M} S_m(x + \xi_m, y + \mu_m). \qquad (2)$$

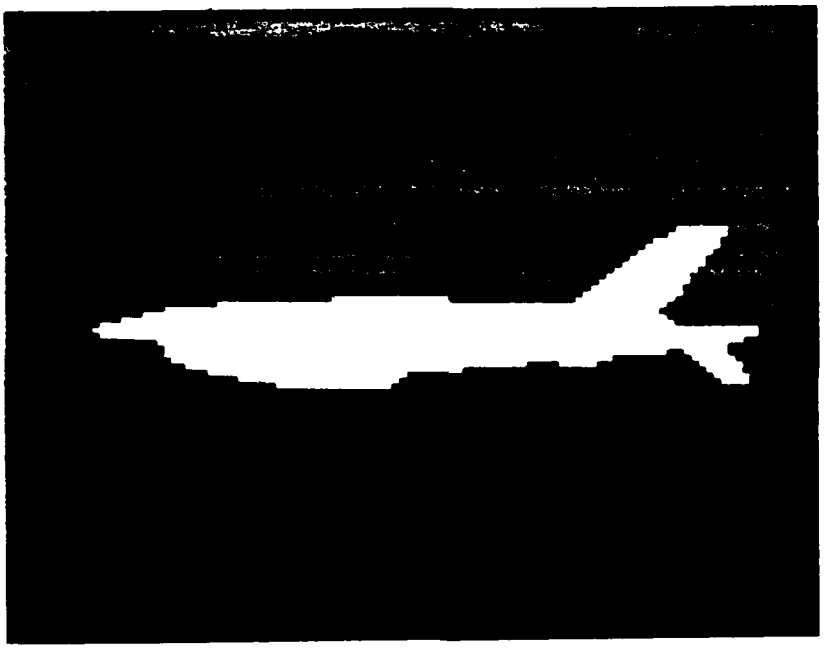
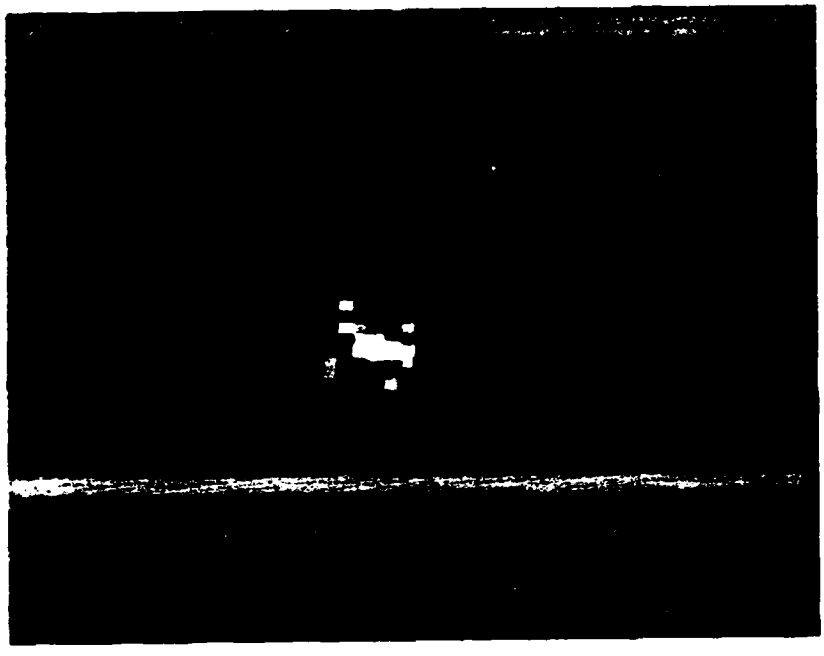
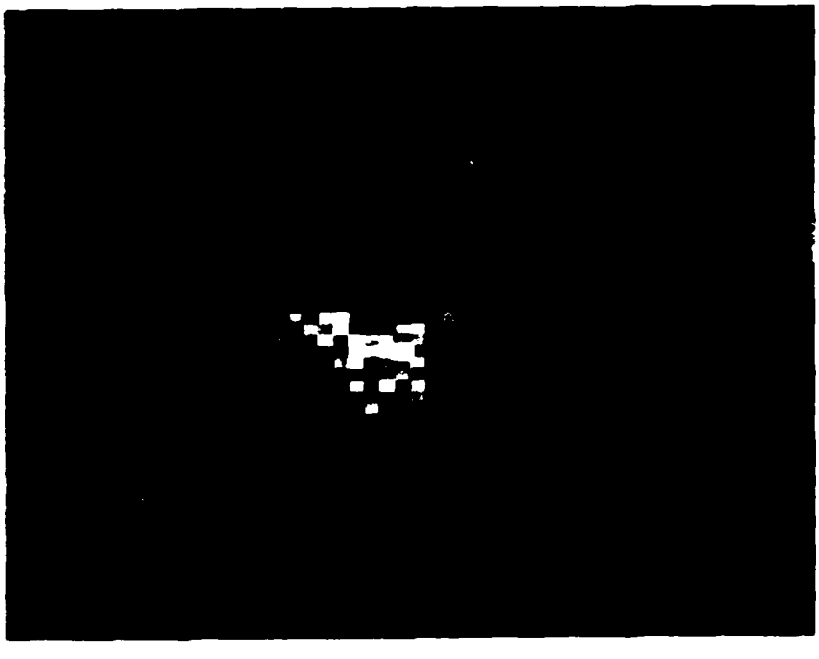


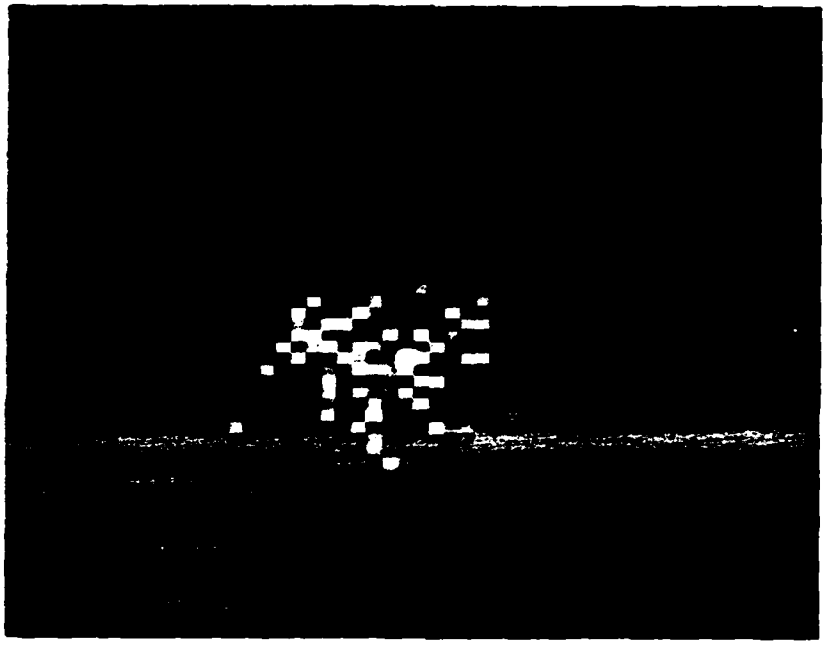
Figure 2: Original drone at gray level 128 (magnified 4x).



3(a) Original s.d. of Gaussian random numbers was 1.5.

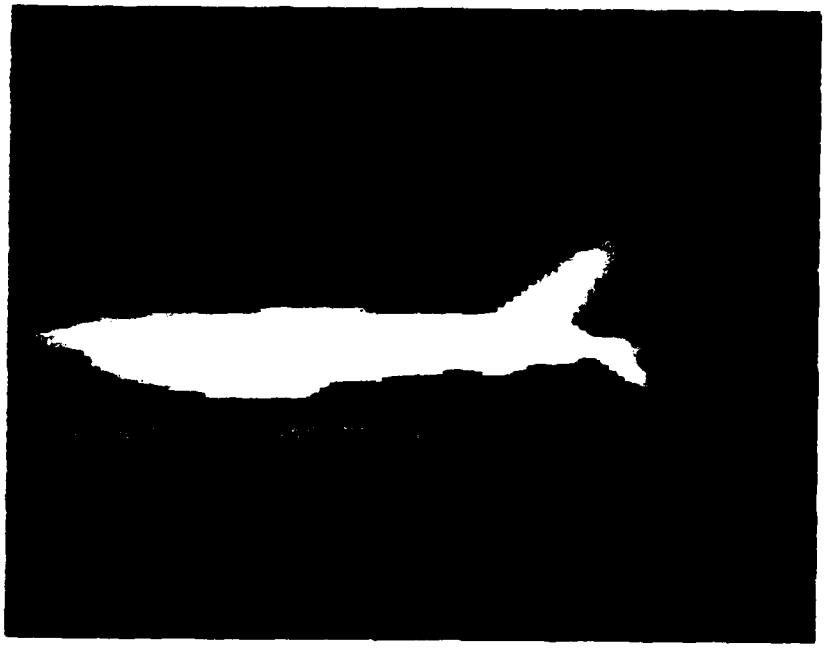


3(b) Original s.d. of Gaussian random numbers was 2.5.

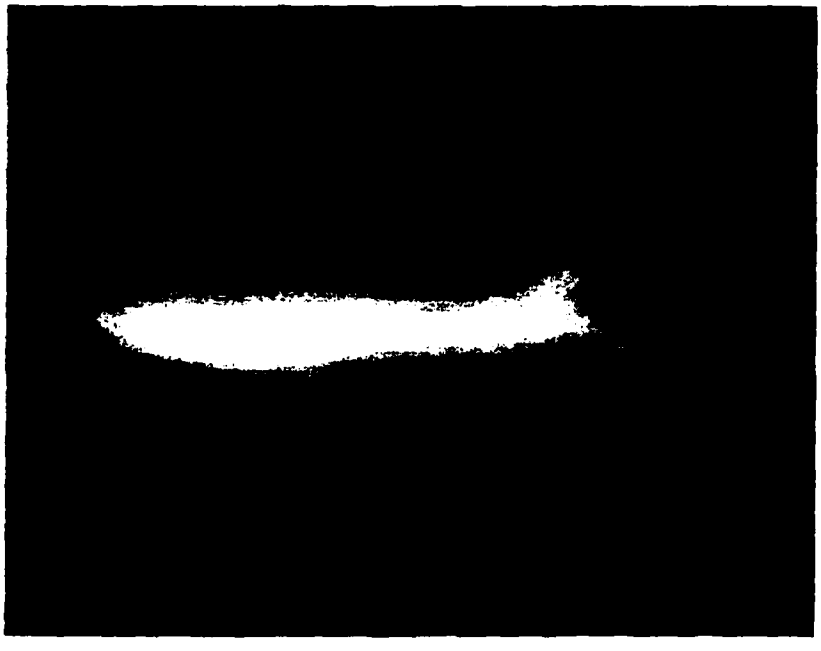


3(c) Original s.d. of Gaussian random numbers was 3.5.

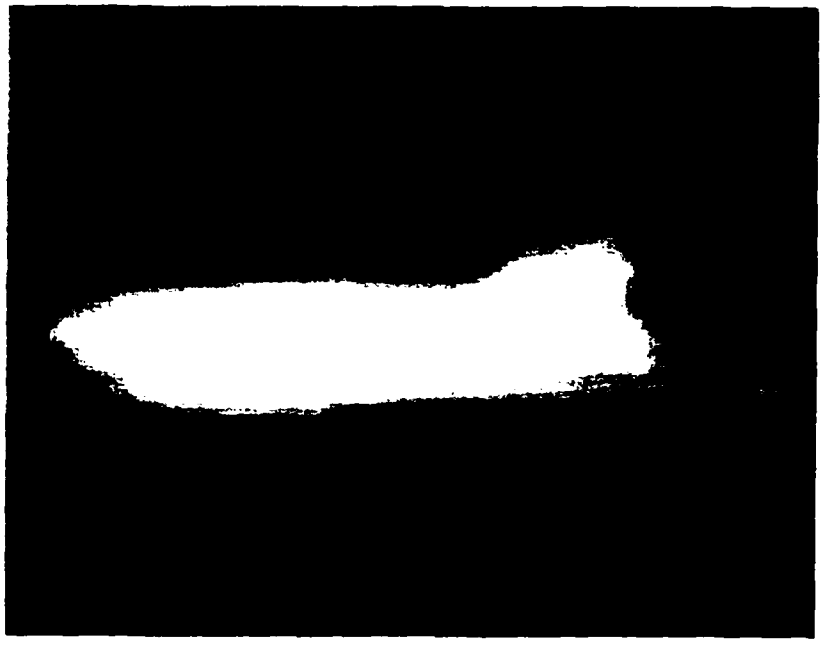
Figure 3: Simulated turbulence point spread functions (magnified 8x).



4(a) Original s.d. of Gaussian random numbers was 1.5.



4(b) Original s.d. of Gaussian random numbers was 2.5.



4(c) Original s.d. of Gaussian random numbers was 3.5.

Figure 4: Simulated turbulence images (magnified 4x).

Because all operations involved are linear, we may also write the shift-and-add image s(x,y) as

$$s(x,y) = h(x,y)**f(x,y) + c(x,y) \qquad (3)$$

where h(x,y) is the sum of the shifted hm(x,y) and c(x,y) is the sum of the shifted c(x,y).

A mathematical analysis of this method has been performed by Hunt, Fright, and Bates [4]. This analysis indicates that for a large number M of speckle images, a spike corresponding to the "brightest point" in the original object will be created near the origin of coordinates. The proper alignment of the other portions of the image is due to the linear nature of convolution. This process may break down when a "brightest point" is unavailable in the original object or when several points or areas of approximately equal brightness are present. In this case, the degraded images may be incorrectly aligned causing "ghosts" to appear in the result.

We performed shift-and-add processing on each series of 20 degraded images which had been simulated by the method of section II.2, and the results are summarized in Table 1. We see from this table that both the contrast between the point source (or "glint") and the aircraft body and also the original standard deviation of the Gaussian random numbers used in computing the turbulence psf affect the performance of shift-and-add processing. It does appear however, that even in the cases where shift-and-add processing produced a blurred resultant image, the shift-and-add image should be a better candidate for deblurring than are

original s.d. of Gaussian iterates	Aircraft Gray Level 64	128	192
0.5	good image reconstruction	good image reconstruction	good image reconstruction
1.5	good image reconstruction	good image reconstruction -very faint "ghost" in background	"ghost" image appears in background of reconstructed image
2.5	"ghost" image appears in background of reconstruction	blurred image reconstruction	blurred image reconstruction
3.5	highly blurred image reconstruction	highly blurred image reconstruction	highly blurred image reconstruction

Table 1

the individual turbulence images. Shift-and-add images for the case of an original drone gray level of 128 are depicted in Figure 5.

We also ran one test to check the effect of increasing the number M of processed images M from 20 to 50. The case chosen was that of the aircraft gray level at 128 and the turbulence psf generated from Gaussian random numbers with a standard deviation of 3.5. It is interesting to note that for this case, the shift-and-add images for M = 20 and M = 50 are visually indistinguishable. It would therefore appear likely that beyond some point, for this case M = 20 or perhaps less, no improvement is gained by processing more images.

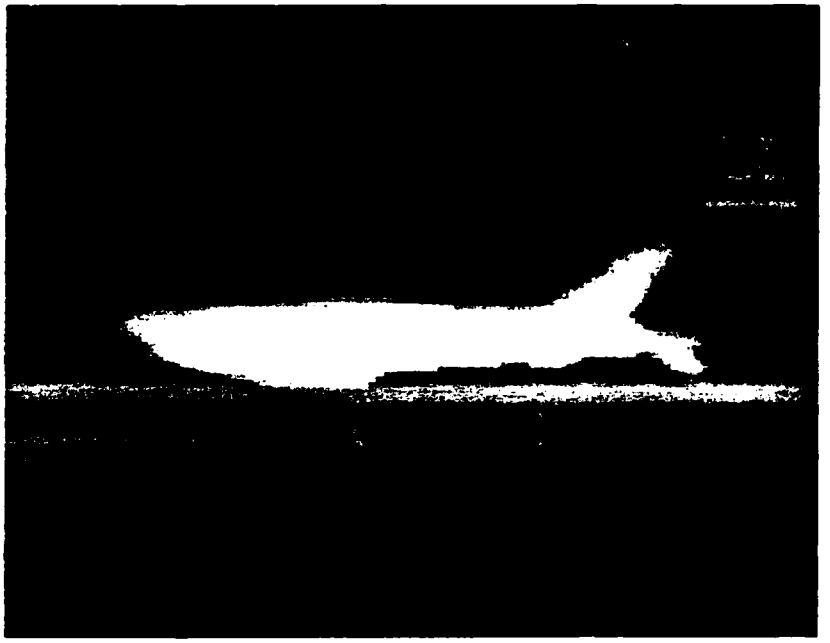
Deblurring

As noted in the previous section, the results of shift-and-add processing tended to be blurred, especially for higher levels of turbulence. Assuming the model of equation (3) for the shift-and-add image

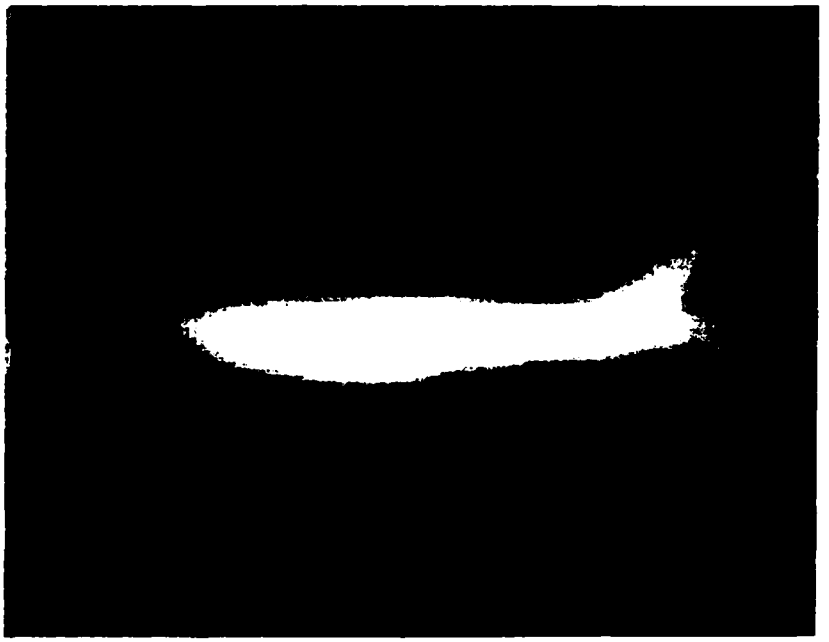
$$s(x,y) = h(x,y)**f(x,y) + c(x,y),$$

we wish to recover f(x,y). Our task is somewhat simplified by the fact that our simulated images are free of degradations other than atmospheric turbulence, i.e., we may essentially ignore the term c(x,y). Thus the removal of blur involves estimation of the cumulative point spread function h(x,y) and filtering.

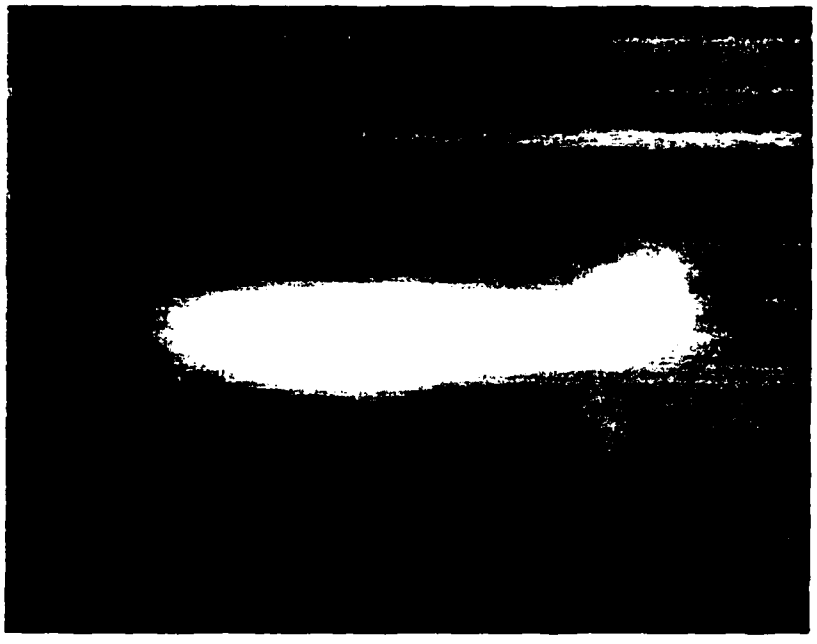
To estimate the cumulative psf, we inserted a point source near but outside the drone body (Figure 6). This point source was at a lower gray level than the one of the drone body so that



5(a) Original s.d. of Gaussian random numbers was 1.5.



5(b) Original s.d. of Gaussian random numbers was 2.5.



5(c) Original s.d. of Gaussian random numbers was 3.5.

Figure 5: Shift-and-add processed images (magnified 4x).

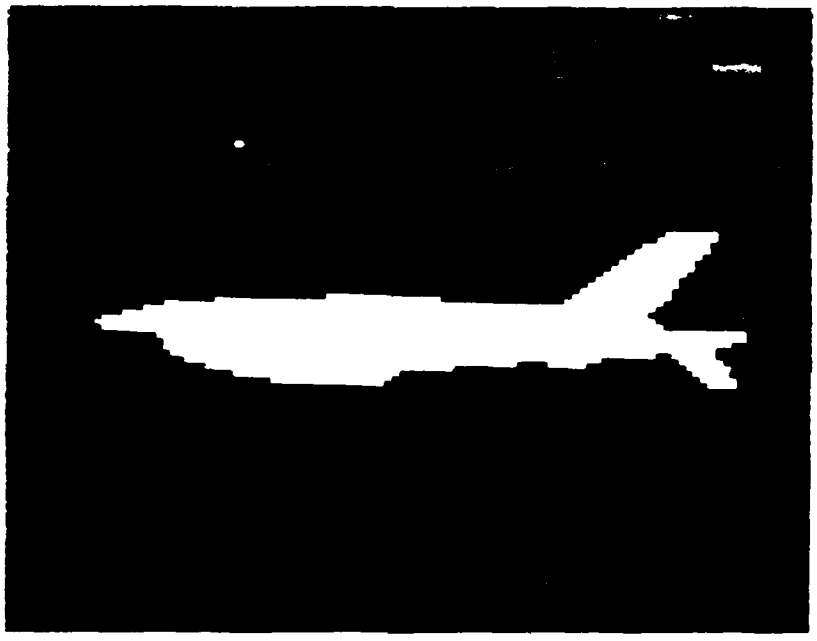


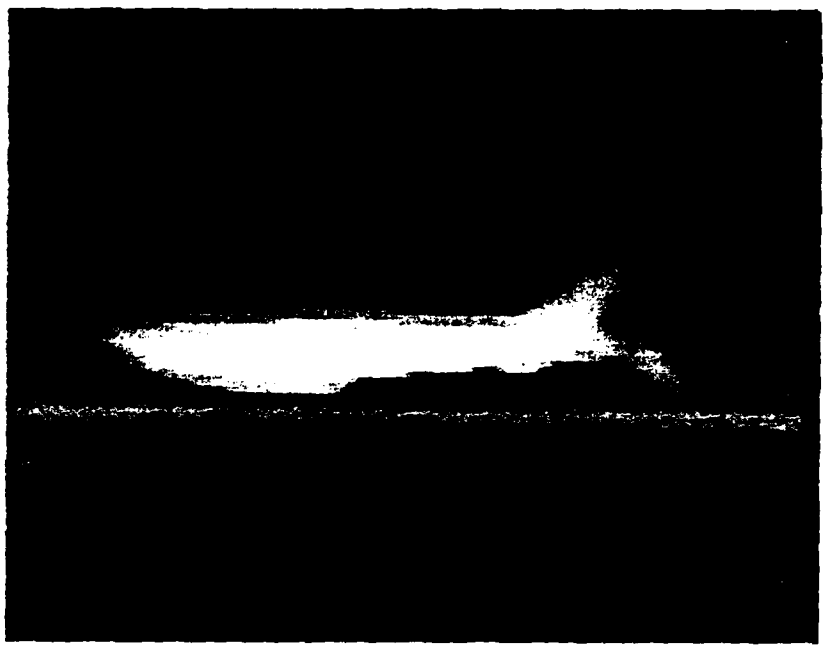
Figure 6. Addition of a second point source for use in cumulative psf estimation (magnified 4x).

it would not interfere with the shift-and-add processing. Series of 20 turbulence images were then created by convolution with the simulated point spread functions $h_m(x,y)$, and shift-and-add images were formed from each of these series. By checking the results of this processing on the second point source, we were able to obtain a point spread function $h(x,y)$ for the entire process. Using a least squares fit, this point spread function was approximated by a Gaussian to be used in filtering.

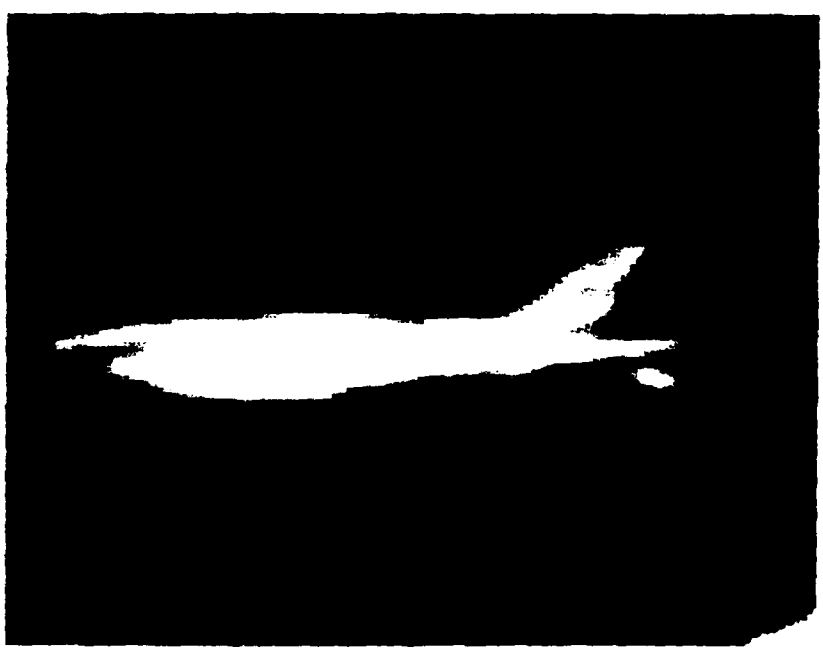
In order to recover the original object, we used the Wiener and Cannon filters. The form of the Wiener filter used was

$$W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{\sigma_n^2}{\Phi_f(u,v)}}$$

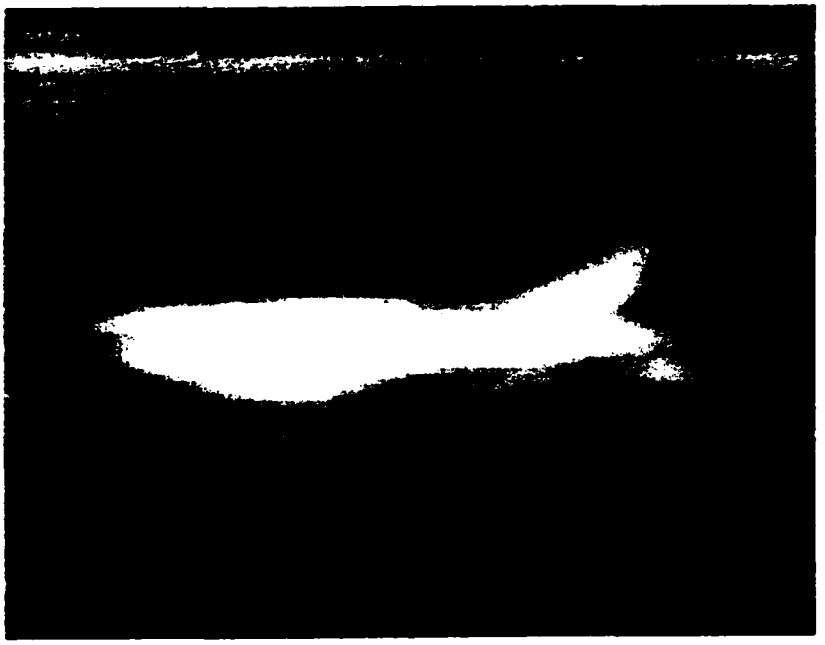
where $H(u,v)$ is the Fourier transform of the Gaussian approximation to $h(x,y)$, * denotes complex conjugation, $|\cdot|$ denotes the modulus of the transform, σ_n^2 is noise variance, and $\Phi_f(u,v)$ is the ideal image power spectrum. The results of this filtering are shown in Figure 7, and we see a great improvement in sharpness of the drone outline, especially for the cases of 2.5 and 3.5 standard deviation in the original Gaussian iterates. We also note fairly severe "ringing" in these filtered images - more than is actually visible in Figure 7. Because we are particualrly interested in obtaining the silhouette or outline of the object for use in calculating moments or Fourier descriptors, it is advantageous to accept some ringing in order to retain sharpness of the outline. We thresholded the restored images in order to



7(a) Original s.d. of Gaussian random numbers is 1.5.



7(b) Original s.d. of Gaussian random numbers is 2.5.



7(c) Original s.d. of Gaussian random numbers is 3.5.

Figure 7: Wiener filter deblurring of shift-and-add images (magnified 4x).

compare silhouettes (or outlines) with that of the thresholded original. These images are shown in Figures 8 and 9.

We also tried deblurring with the Cannon filter which sometimes increases object sharpness. We used the form

$$C(u,v) = \sqrt{\frac{1}{|H(u,v)|^2 + \sigma_n^2/\Phi_f(u,v)}} \quad .$$

In our case, the results from this filter were poorer than those from the Wiener filter.

From Figures 4 and 9, we can see that the combination of shift-and-add processing and Wiener filter deblurring produces marked improvements in images degraded by atmospheric turbulence.

Moments

We have previously run tests to determine the effect of rotation, scale change, and noise on invariant moments. These results indicated that any classification system based on moments should weight the third and seventh moments less heavily and should use the largest weights on the first and second moments.

We have now considered the effect of atmospheric turbulence on the invariant moments. Our first test involved the effect of atmospheric turbulence and of shift-and-add processing alone. For each standard deviation, moments were calculated for the original undegraded image (aircraft gray level 128), for two of the turbulence images from the series of 20, and for the resulting shift-and-add image. The percent error was measured

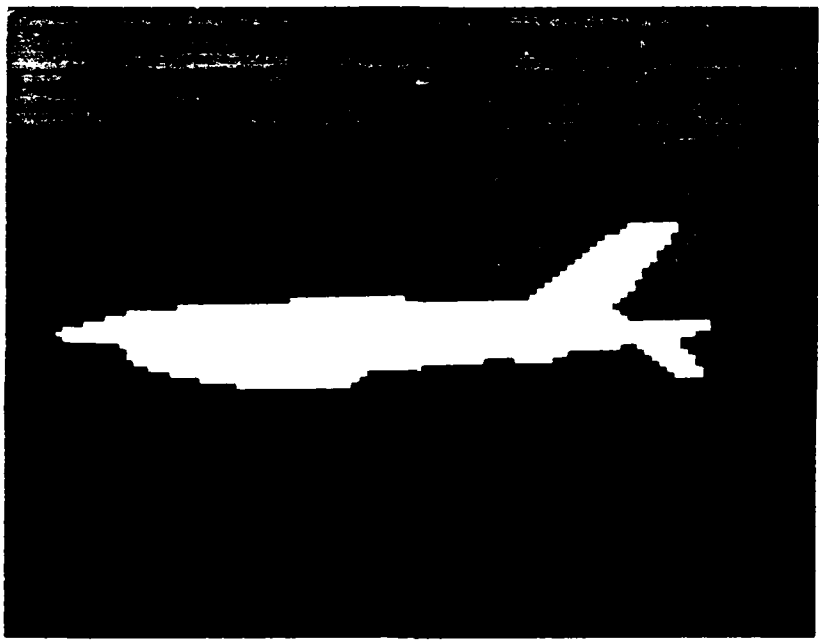
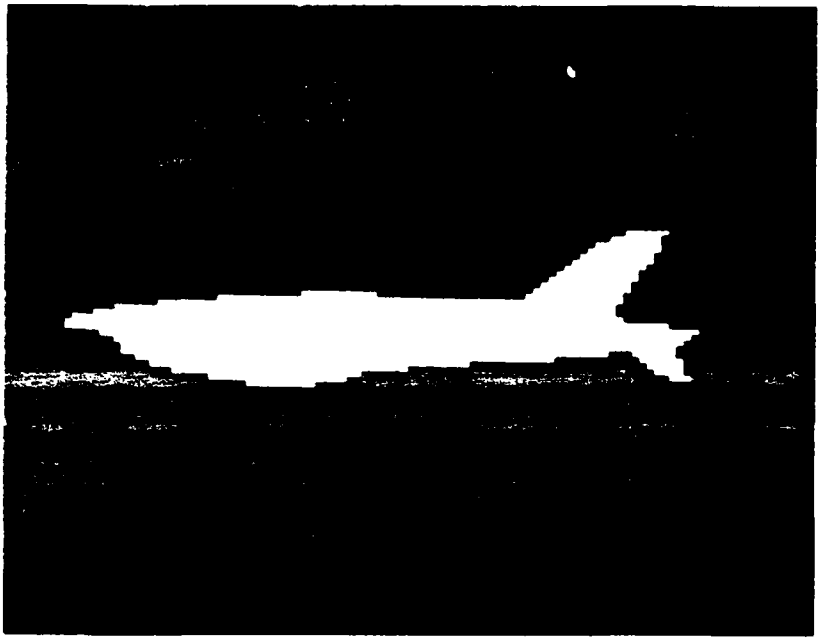
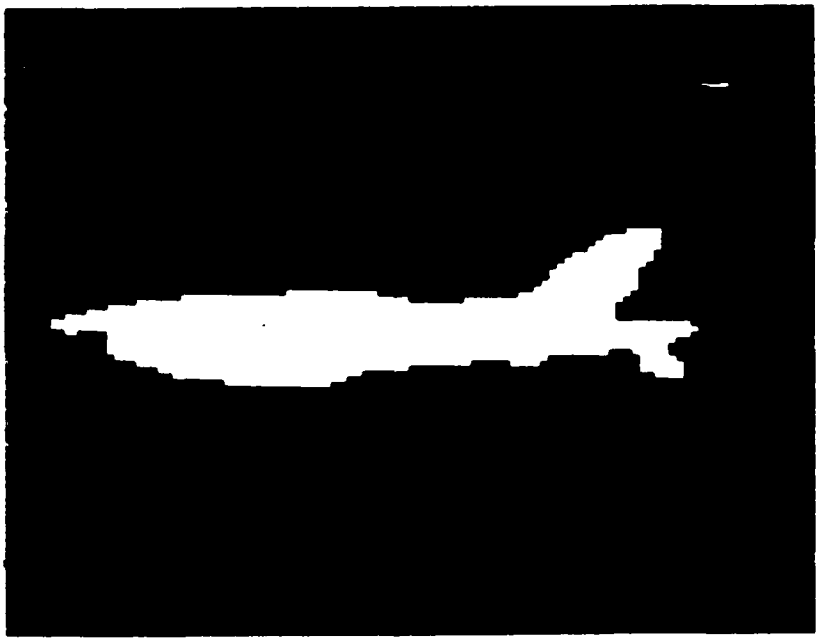


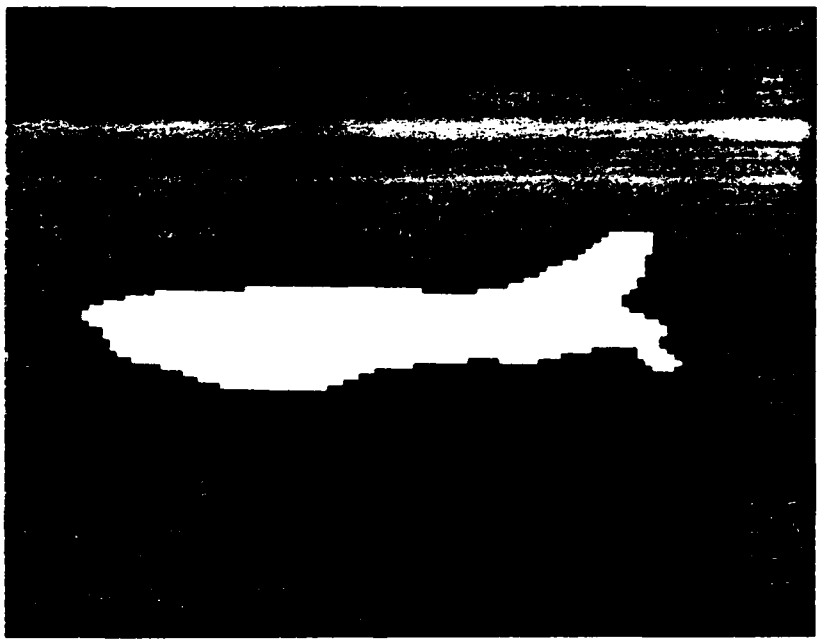
Figure 8: Thresholded original drone (magnified 4x).



9(a) Original s.d. of Gaussian random numbers was 1.5.



9(b) Original s.d. of Gaussian random numbers was 2.5.



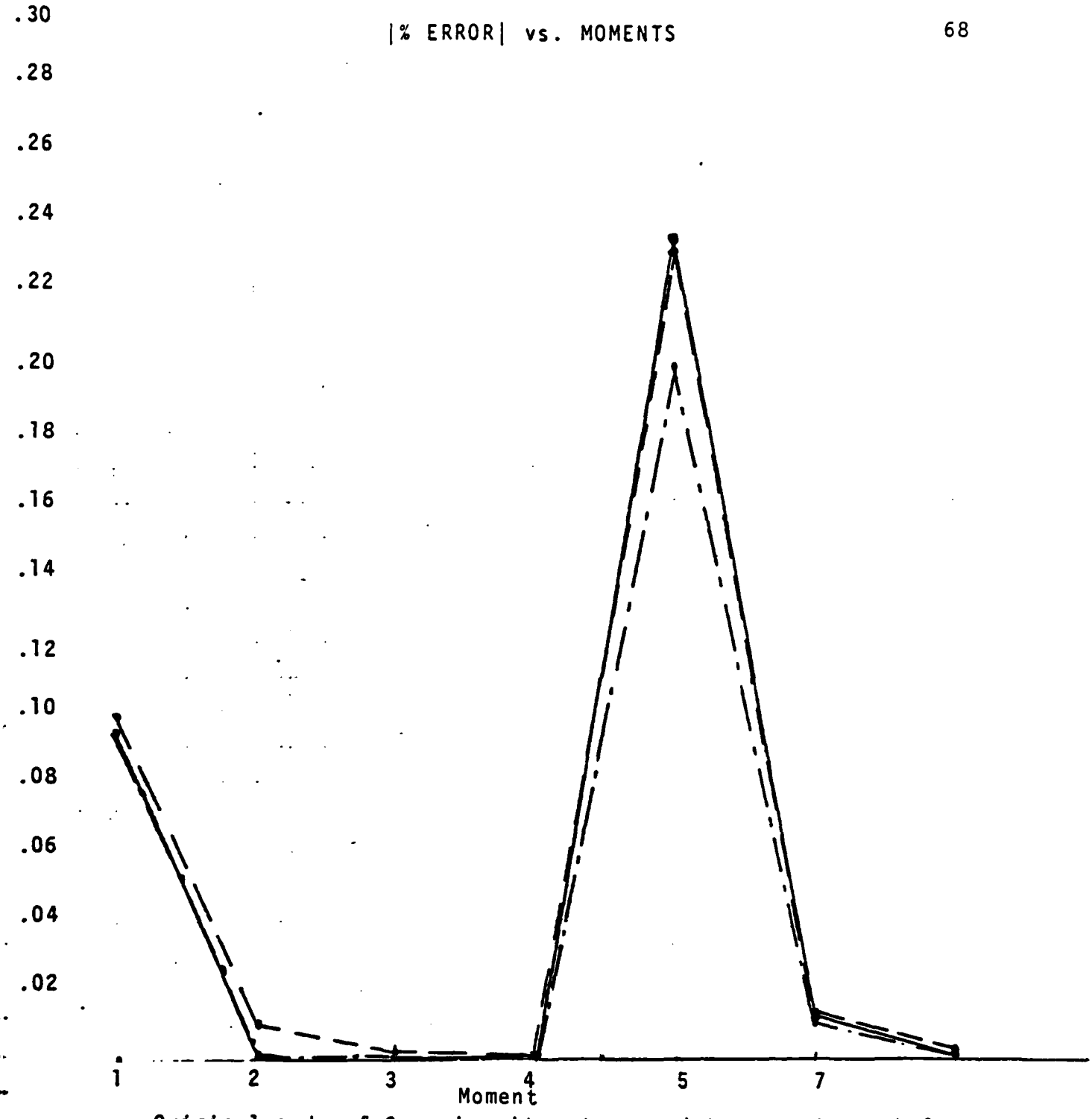
9(c) Original s.d. of Gaussian random numbers was 3.5.

Figure 9: Thresholded restored images (magnified 4x).

relative to the moments of the original image and is plotted in Figures 10-13. From these figures, it appears that the first and especially the fifth moment are most sensitive to turbulence degradations and possibly should be weighted less heavily in any classification schemes. These figures also show that contrary to expectation, an original standard deviation of 1.5 produces the most error in the moments, more than the more severe degradations produced by standard deviations of 2.5 and 3.5. The moments of the shift-and-add image for this case are also higher in error than are those of the other shift-and-add images. We may explain this by referring to Table 1 where we note the presence of a faint "ghost" aircraft in the background. We do note that in all cases, except for 1.5 standard deviation in the original Gaussian iterates, shift-and-add processing did reduce the error in the moments.

Our second test involved calculating the moments of the binary thresholded original image (Figure 8) and of the shift-and-add images after Wiener filtering and thresholding (Figure 9). This added processing actually brought about little reduction in the error of the moments as may be seen in Figure 14. However, the accuracy of the first moment was improved, which is fortunate, since our previous tests involving rotation, size change, and noise indicated that the first moment should be weighted heavily in classification algorithms. As in the case for turbulence or shift-and-add images, the fifth moment appears to be most subject to error and should thus be weighted less heavily in any classification schemes based on moments.

|% ERROR| vs. MOMENTS



Original s.d. of Gaussian iterates used to compute turbulence psf's is 0.5

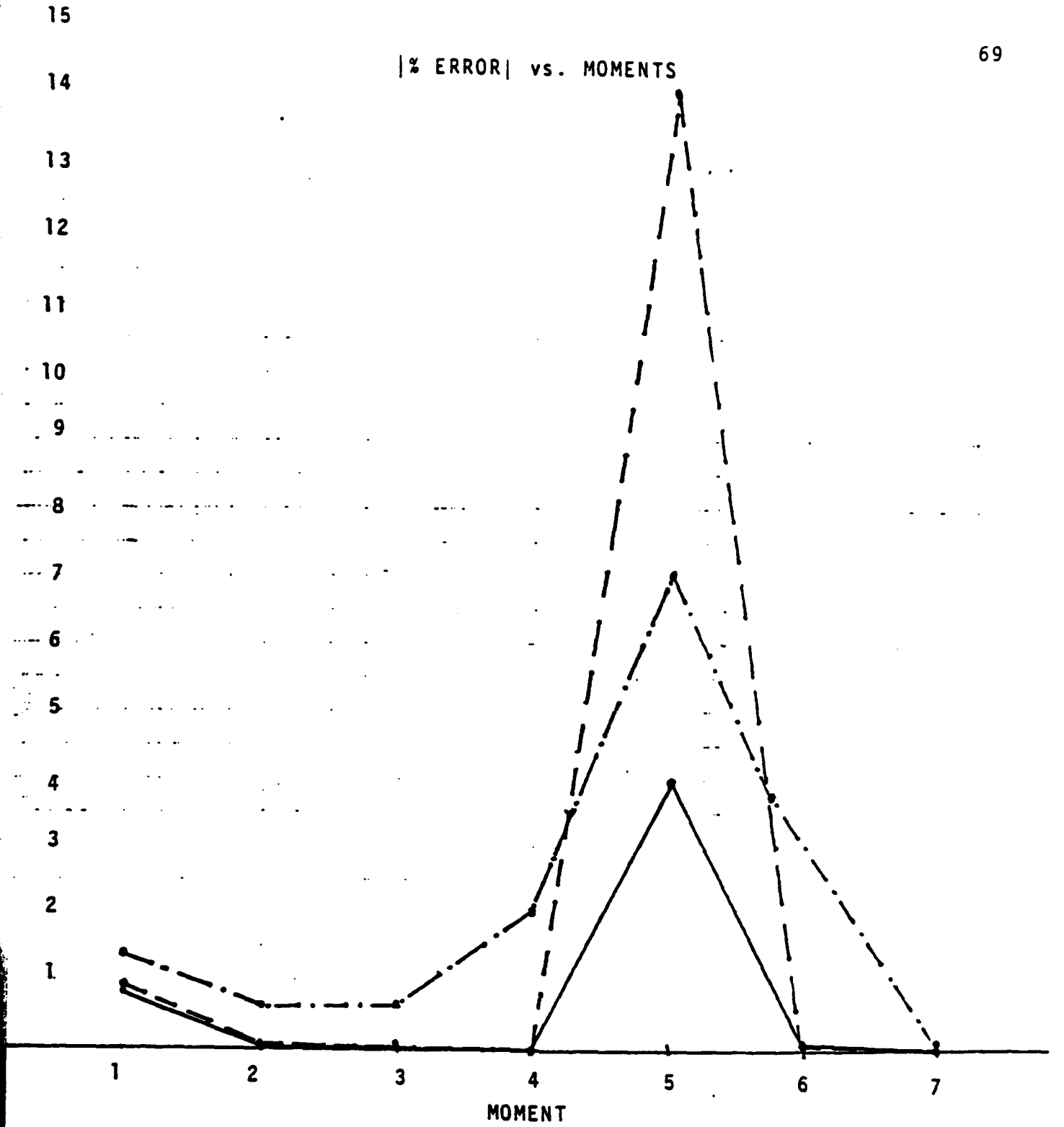
- turbulence corrupted image #5

- turbulence corrupted image #15

-.- shift-and-add image (20 files)

-%error measured relative to the moments of the uncorrupted image

Figure 10



Original s.d. of Gausian iterates used to compute turbulence psf is 1.5

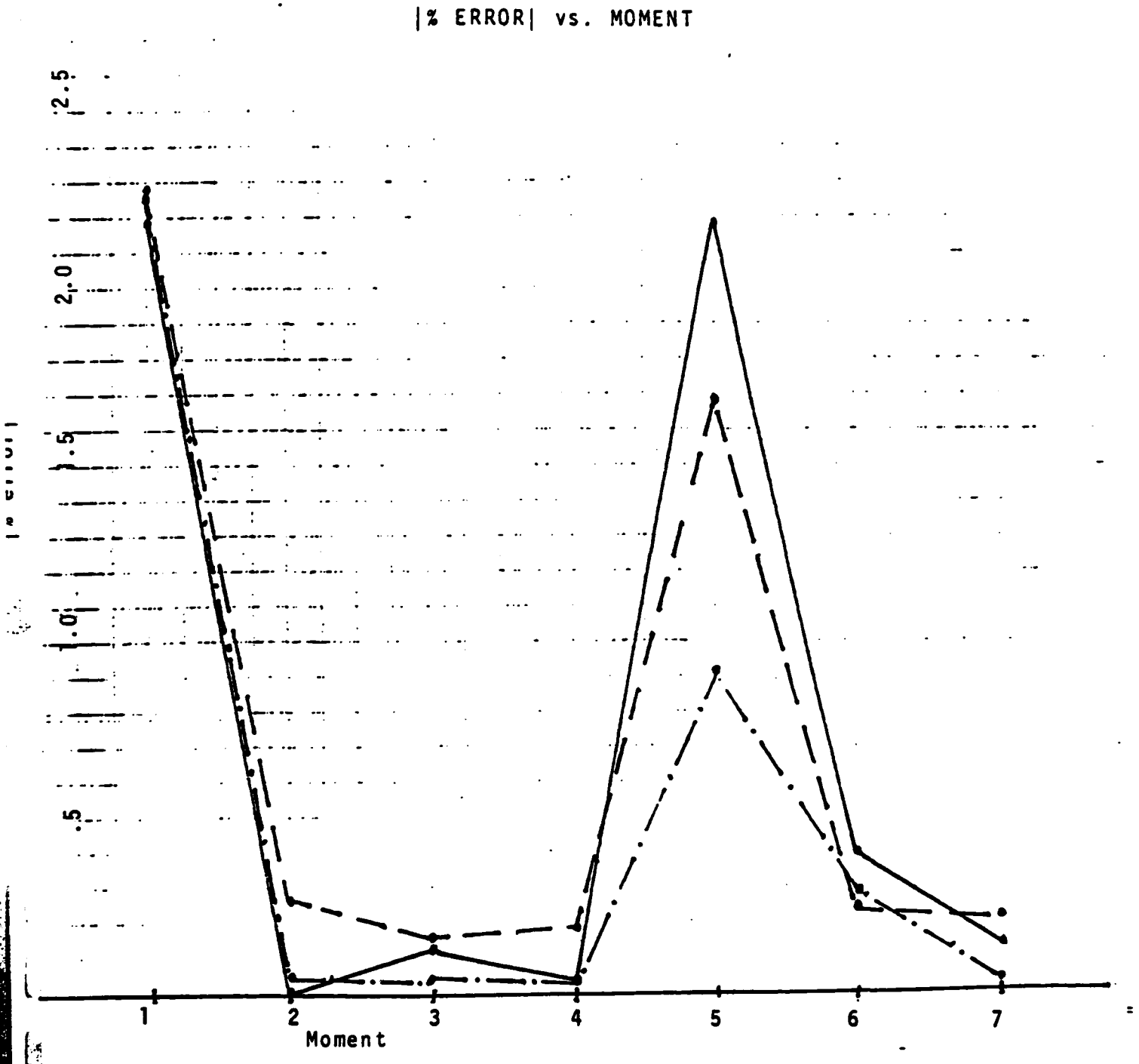
- turbulence corrupted image #5

- turbulence corrupted image #15

-.- shift-and-add image (20 files)

-% error measured relative to the moments of the uncorrupted image

Figure 11



original s.d. of Gaussian iterates used to compute the turbulence psf's is 2.5

-turbulence corrupted image #5

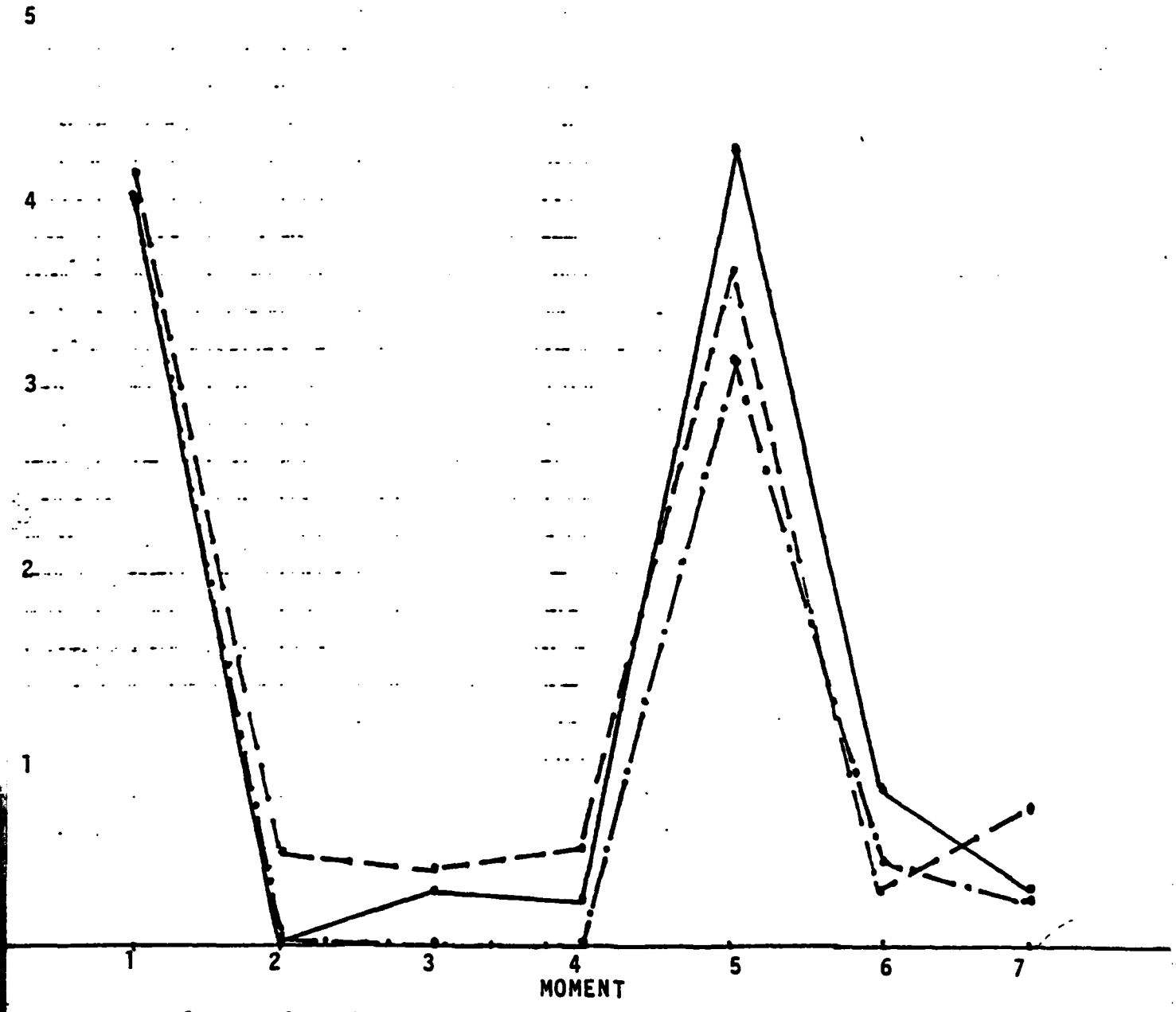
-turbulence corrupted image #15

-.-shift-and-add image (20 files)

-%error measured to the moments of the uncorrupted image

Figure 12

|% ERROR| vs. MOMENT



Original s.d. of the Gaussian iterates used in computing the turbulence psf's is 3.5

- turbulence corrupted image #5

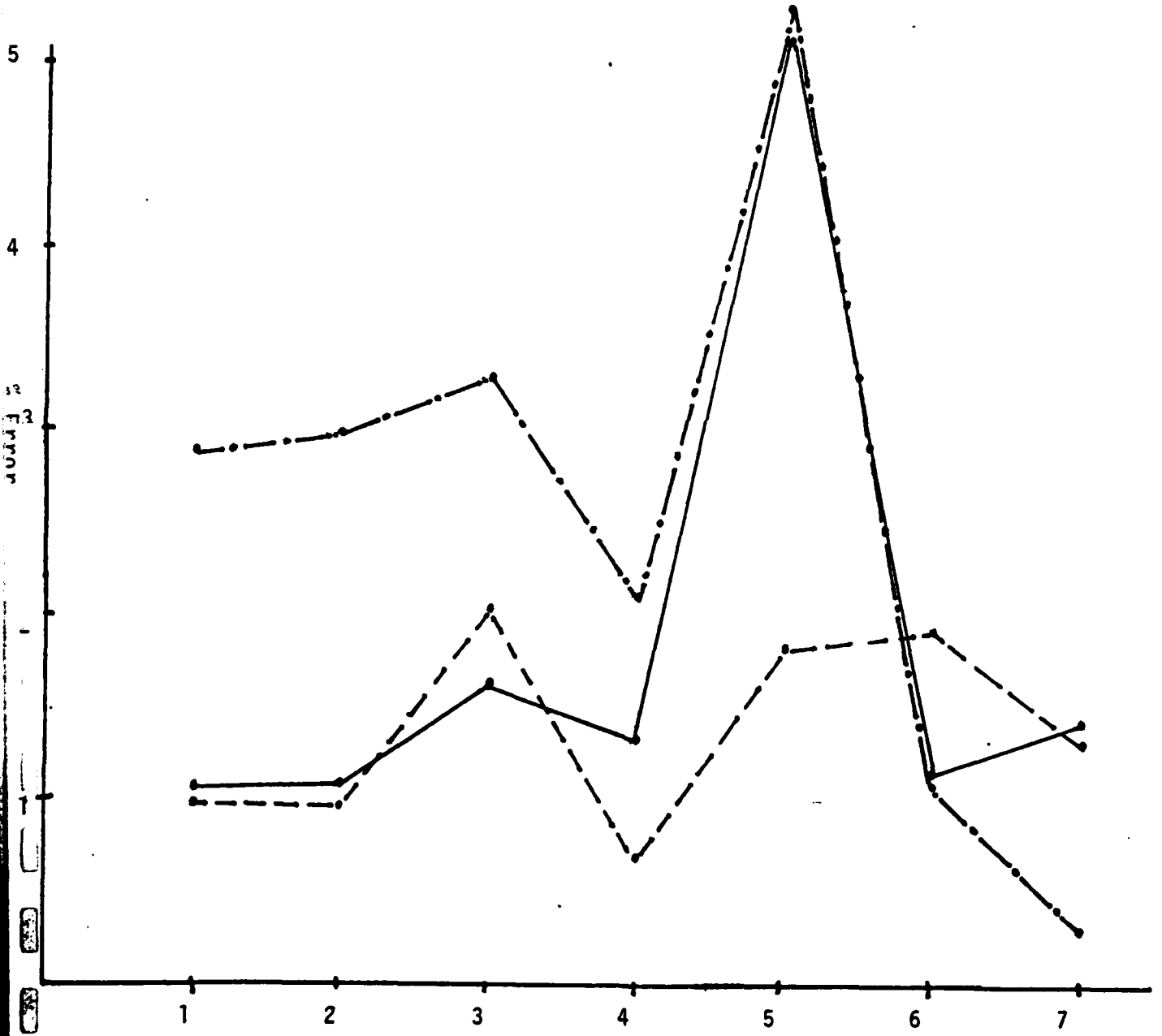
- turbulence corrupted image #15

---. shift-and-add image (20 files)

% error is measured relative to the moments of the uncorrupted image

1% ERROR 1 vs. INVARIANT MOMENT

- after shift-and-add, Wiener filter restoration, and thresholding



——————original s.d. of Gaussian iterates is 1.5

– – – – – – " " " " " " "

– . – . – . " " " " " " "

% error is measured relative to the moments of the thresholded original image

Figure 14

References

[1] Benjamin L. McGlamery, "Computer Simulation Studies of Compensation of Turbulence Degraded Images," Proceedings SPIE, vol. 74, pp. 225-233, 1976.

[2] R. H. T. Bates and F. M. Cady, "Towards True Imaging by Wideband Speckle Interferometry," Optics Communications, vol. 32, pp. 365-369, 1980.

[3] F. M. Cady and R. H. T. Bates, "Speckle Processing Gives Diffraction-Limited True Images from Severely Aberrated Instruments," Optics Letters, vol. 5, pp. 438-440, 1980.

[4] B. R. Hunt, W. R. Fright, and R. H. T. Bates, "Analysis of the Shift-and-Add Method for Imaging Through Turbulent Media," JOSA, vol. 73, no. 4, pp. 456-465, 1983.