STANFORD ARTIFICIAL INTELLIGENCE PROJECT
MEMO AIM-166

STAN-CS-72-281

COMPUTER INTERACTIVE PICTURE PROCESSING

BY

LYNN H. QUAM SIDNEY LIEBES, JR.
ROBERT B. TUCKER MARSHA JO HANNAH
BOTOND G. EROSS

SUPPORTED BY
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AND
ADVANCED RESEARCH PROJECTS AGENCY
ARPA ORDER NO. 457

APRIL 1972

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

\NFORD ARTIFICIAL INTELLIGENCE PROJECT APRIL 1972
MO AIM-166

JMPUTER SCIENCE DEPARTMENT
EPORT CS-281

COMPUTER INTERACTIVE PICTURE PROCESSING

by

Lynn H. Quam, Sidney Liebes, Jr,
Robert B. Tucker, Marsha Jo Hannah
Botond G. Eross

ABSTRACT:

This report describes work done in image processing using an interactive computer system, Techniques for Image differencing are described and examples using images returned from Mars by the Mariner Nine spacecraft are shown, Also described are techniques for stereo image processing, Stereo processing for both conventional camera systems and the Viking 1975 Lander camera system is reviewed.

This research was supported by the National Aeronautics and Space Administration and the Advanced Research Projects Agency,

Reproduced in the USA, Available from the National Technical Information Service, Springfield, Virginia 22151, pricer full size copy $3,00; microfiche copy $0,95.

TABLE OF CONTENTS

A, INTRODUCTION

This report reviews the work of the Stanford Artificial Intelligence Laboratory done under NASA Grant NCR-05-020-508, For the purpose of this report the work la divided into three areas: image information management, automated image differencing, and stereoimage processing, Section B discusses some of the problems Involved with handling a large volume of Image data and someof the solutions, Section C reviews the Image differencing work together with various input processing steps used in Preparing the data for differencing, Section D describes work done in the area of near-field stereo image analysis oriented towards the Viking 1975 lander camera system, Appendicies A and B are two term papers related to the questior of stereo image processing which were Supported, In part, by this Want, The data base used in this work has come from the Mariner Mars probes of 1969 and 1971, from prototype Viking 1975 camera equipment, and from locally produced images of earth scenes,

In addition to the above mentioned grant, this work recieves support from JPL Contract 9132489, Langley Conttaot NAS 1-9682, and ARPA Contract SD-183,

8, IMAGE INFORMATION MANAGEMENT

An Information retrieval capability has been Implemented at the Stanford A/I Project which enables us to quickly review the planet coverage of the MM-71 TV Mission, It is primarily oriented toward revealing the extent of repeated TV coverage of any area specified by latitude and longitude, It enables the User to quickly determine if an area has been photographed, and if so, how many times, on which orbits, by which camera, and by which pictures within an orbit, On a display screen is shown the d'sk of the planet, the footprints of the Images, and vectors indicating view and sun angles, The correspondence between DAS shutter time and the orbit-camera-picture within orbit (Experimenter) identifier is also shown, The user is also able to alter the scale of the display to improve clarity, Most of the input data for this system comes form the MM'71 LIBSET system Operated by the Science Data Team (SDT) at JPL.

As additional TQL (preliminary navigation parameters) Picture catalog data is received it merged with the data previously received and stored on the disk system, This operation provides us with the navigation data necessary to perform the geometric projections described below and also provides the library information used to locate Images on the image data (PTV) tapes and the JPL film products, The exact manner In which the above has been don8 has varied dur Ins the mission since the reliability and timeliness with which we have received the TQL and Picture Catalog data tapes has varied, we began receiving the Picture Cataloa data on tape near the end of the nominal mission primarily In an attempt to make up for the absence of the complete SEDR, SDT also includes the IPL Enhancement Log on the tape which Allows us to automatically review the RDR (final decalibration processing) status,

As additional data on the Images is more readily available, It will become a part of the system, In the short time that the system has been operational it has proven to be a valuable asset, Since more than six thousand images exist, the need for such a system is obvious, Its importance will grow as the number of images, and the information about these Images, increases, This capability could prove quite usefu I in picture targeting and landing site selection for the Viking mission,

It Is important to note that this is an interactive system oriented towards the needs of the scientists, Its success depends on its ability to present data in a manner consistant in format and organization with the way the experimenters view the object under investigation,

The above mentioned capability actually represents the initial phase of the process for the projection and differencing operation, With the identifiers and footprints of all the images before the user the list can be pruned Until just the footprints of interest are present, The user can then proceed directly to the projection and differencing steps,

One area which deserves additional attention is the cataloging of output products, When simple operations are performed on a single image it is generally quite easy to catalog the output products, The problem becomes somewhat complex when several images are combined as in the case of Image differencing, color reconstruction, and polarization studies, The approach we are using involves the use of a header of arbitrary size which is stored with every Processed image, Actually, two headers are used; one is a "short form" which simply indicates the images and processes used to create the image in question, the other is a "long form" which includes the actual parameters used In each of the processes, The benifits of such a system come only with the ability to quickly retrieve images utilizing the information in these headers and informatlon about the original Images (location on the planet, camera, filter, shutter time and date, orbit, and etc),

The above capacity, when combined with a disc based storage system (see Input processing below), gives the scientist a significant degree of flexibility to review the image data and the Processing carried out on it,

C. AUTOMATED IMAGE DIFFERENCING

Input Processing

The processing of a series of images is generally initiated by an investigator supplying us with a list of picture identifiers (either DAS time or orbit and picture within orbit) of the image8 in which he is interested.

The identifiers are entered at a terminal and a search of library information is made to determine: if navigation data is available for the images, if the PTV Or RDR tapes for the images are available in our library, if the images have already been loaded into our disk system (see below) as the result of previous processing, and other information related to the image (filter, exposure time, and etc).

If the image has not been previously processed but is available In our tape library it 13 mounted on a tape unit and read in, Several operation are applied to the data at this time. A "first order" photometric correction is made using a two dimensional interpolation Of a matrix of vidicon response parameters, Reseau marks are also located and square areas (which approximate their actual shape in an image) are set to a zero DN value, thus identifing those points as invalid data for later operations, Next, a "custering" operation is performed to identify bit droppages in the image data, This is done by comparing • ach point to the mean of a three by three area around it, A difference of more than three standard deviations from the mean identifies it as and error. This procedure also has the effect of identifying a8 errors the various "fringes" left as the result of modeling the images of the reseaus a3 squares, Finally, these pixels identified as errors are replaced by averaging the neighboring points, The result is an imagee with a reasonable degree of photometrto integrity (significant error8 still exists along the edges) with reseaus shown as square arrays of invalid (zero) data,

As these processes are being applied, the image is being transfered from the tape to the general system disk area, One additional operation take8 Place, The successive differences between each pixel and it3 neighbor on the left is calculated and a histogram of all these values is developed, This histogram is used to develop a modified Huffman coding scheme for the differences, This is a variable length coding system with which we • re able to compress the images by a faator of between two and three (much better on satellite Images) without any information loss, As this compression is being done the resulting data Is stored under our User Disk Pack (UDP) arrangement on the IBM 3330 disk system, This facility allows a user to have an exchangeable disk pack for his own use, We expect to be able to store about 200 complete images on a single disk pack using the above compression scheme.

The user is thus left with the compressed version on the UDP for future reference (after decompression) and the normally coded version on the general file system. The normal one is used as described below to satisfy the current processing request and afterwards is deleted.

Image Differencing (1)

Two images are differenced by transforming them to the same projection, aligning them, and then subtracting the aligned images.

The user is able to select any portion of the intersection of two images for differencing. Actually, what is selectd is a rectangular window in an orthographic projection for which there is data in both images. Thus, two new images are created. However, they are not accurately aligned due to errors in the navigation (TQL) data that is used in developing the projections. these errors in alignment are determined by successively aligning subareas of the window using a cross correlation technique. For each sub-area an error vector is thus determined. One of the original projections is then repeated incorporating this error information and the result is two images that are generally in alignment to within a pixel.

These two aligned images, call them A and B, are then subtractd. The difference images A-B and B-A are created and displayed on a standard TV monitor together with the images A and B.

Output Products

The experimenter is then able to make 4" by 5" Polaroid prints of the images displayed. Also produced is a computer print-out describing the processes by which these images were generated and the parameters used in each process. This log is maintained automatically by the individual processing steps and resides on the disk system.

The capacity also exists to put these resulting images and notations describing them, onto magnetic tape in a form that can be read by the JPL-IPL Video Film Converter for production of hard copy. These products are then entered into the Science Data Team's data library.

----

(1) See "Computer Comparison of Pictures", Lynn H. Quam, Stanford Artificial Intelligence Project Memo AIM-1448 May 1971.

Examples of Image Differencing

Discussed below are some examples of images which have been projected, aligned, and differenced using the techniques disscussed.

FIGURE 1 shows, at the top, Portions of two high resolution Mariner 9 images taken Of the Thyles Mons region of Mars (75 deg. S., 165 deg. W.). These were taken under almost identical illumination and viewing angles eighteen days apart (upper left on Orbit 113, upper right on Orbit 150). The upper frames show the the images after they have been transformed to a common projection and aligned as previously described. Even with this processing done, it is difficult to accurately determine the changes that have taken place. The bottom frames show their difference pairs, left minus right and right minus left. The subtle changes which have occurred in the "riffles" are clearly brought out in these lower frames.

Figure 2 is an example of a remarkable change in a dark tail strikingly brought Out by the picture differencing techniques. These images of the northern part of Thaumasia were taken nineteen days apart under similiar lighting oondltlons and viewed near verticle but from oposite directions. Note the small black tall In the upper right of the area. Since It has not changed It Is cleanly removed by the picture differencing process.

Two views of a portion of Pyrrhae Regio appear in Figure 3 and show some rather obvlous changes. Dark material has appeared along scarps, Crater walls, and other topographical boundaries. Although the major changes in the top frames are easily detectable by the eye, the minor ones emerge clearly only in the difference images.

In Figure 4 is another portion of Pyrrhae Regio. Again, dark material has appeared along a crater wall. These examples are from the same original images as those In Figure 3 and, like Figure 3, have simillar Illumination oondltlons but viewed in opposite directions from near the verticle. The toppographic features are completely removed in the picture difference, leaving only the true albedo changes.

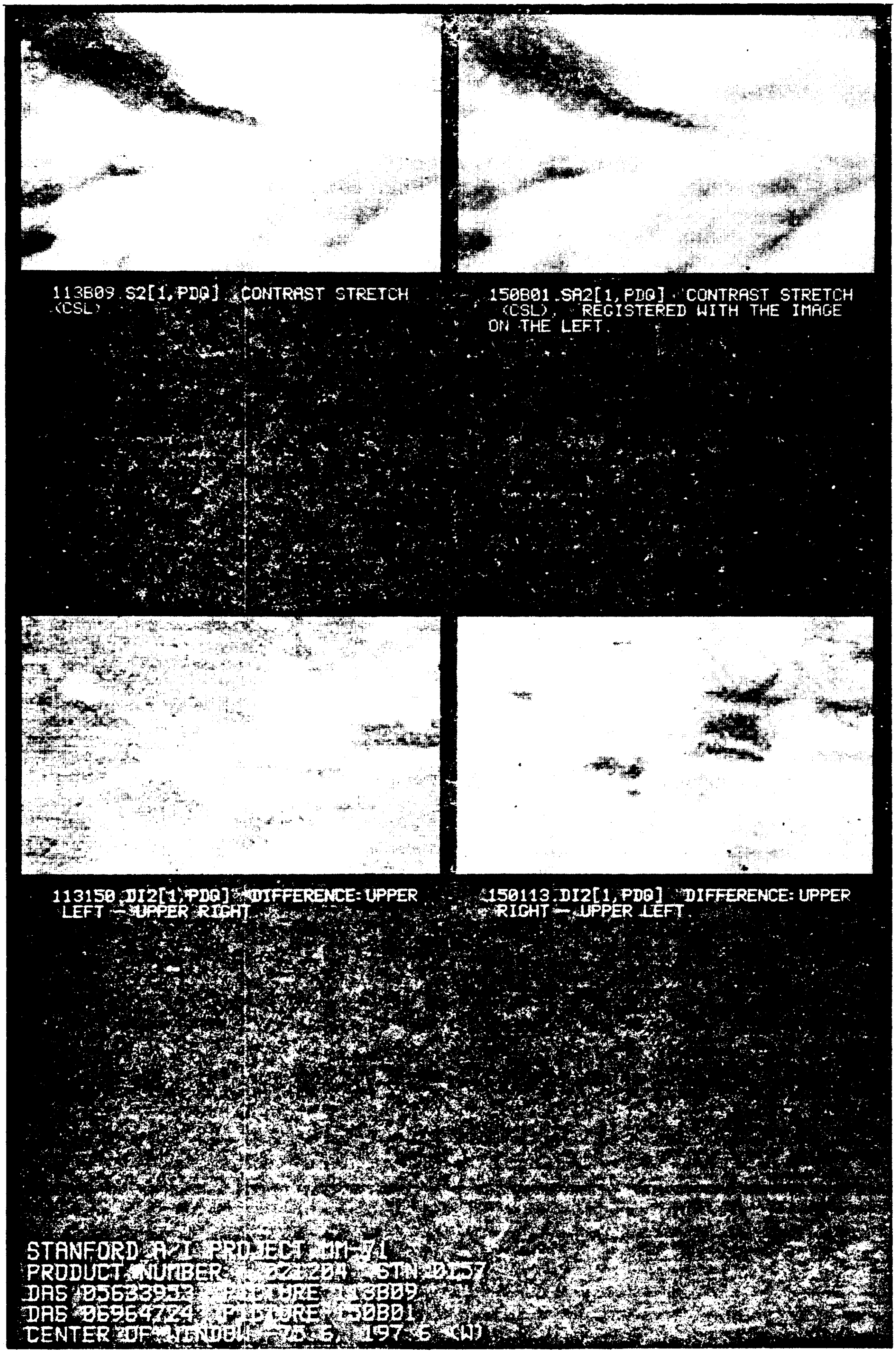
113B09.S2[1,PDQ] CONTRAST STRETCH (CSL)
150B01.SA2[1,PDQ] CONTRAST STRETCH (CSL). REGISTERED WITH THE IMAGE ON THE LEFT.
113150.DI2[1,PDQ] DIFFERENCE: UPPER LEFT - UPPER RIGHT
150113.DI2[1,PDQ] DIFFERENCE: UPPER RIGHT - UPPER LEFT.
STANFORD A.I. PROJECT MM-71
PRODUCT NUMBER
DAS 05633953 PICTURE 113B09
DAS 06964724 PICTURE 150B01
CENTER OF WINDOW

Figure 1

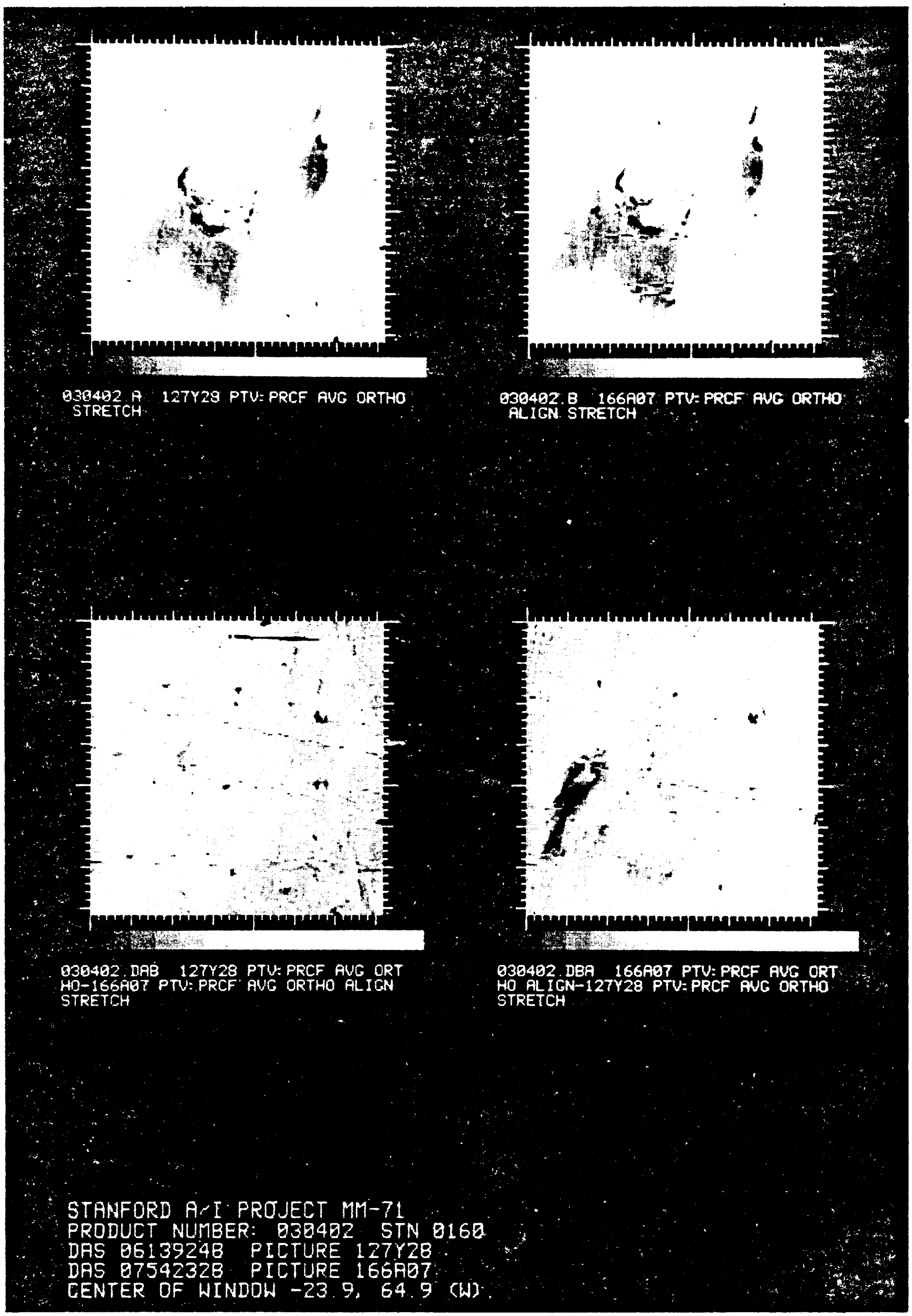
030402.A 127Y28 PTV:PRCF AVG ORTHO STRETCH
030402.B 166A07 PTV:PRCF AVG ORTHO ALIGN STRETCH
030402.DAB 127Y28 PTV:PRCF AVG ORTHO-166A07 PTV:PRCF AVG ORTHO ALIGN STRETCH
030402.DBA 166A07 PTV:PRCF AVG ORTHO ALIGN-127Y28 PTV:PRCF AVG ORTHO STRETCH
STANFORD A/I PROJECT MM-71
PRODUCT NUMBER: 030402 STN 0160
DAS 06139248 PICTURE 127Y28
DAS 07542328 PICTURE 166A07
CENTER OF WINDOW -23.9, 64.9 (W)

Figure 2

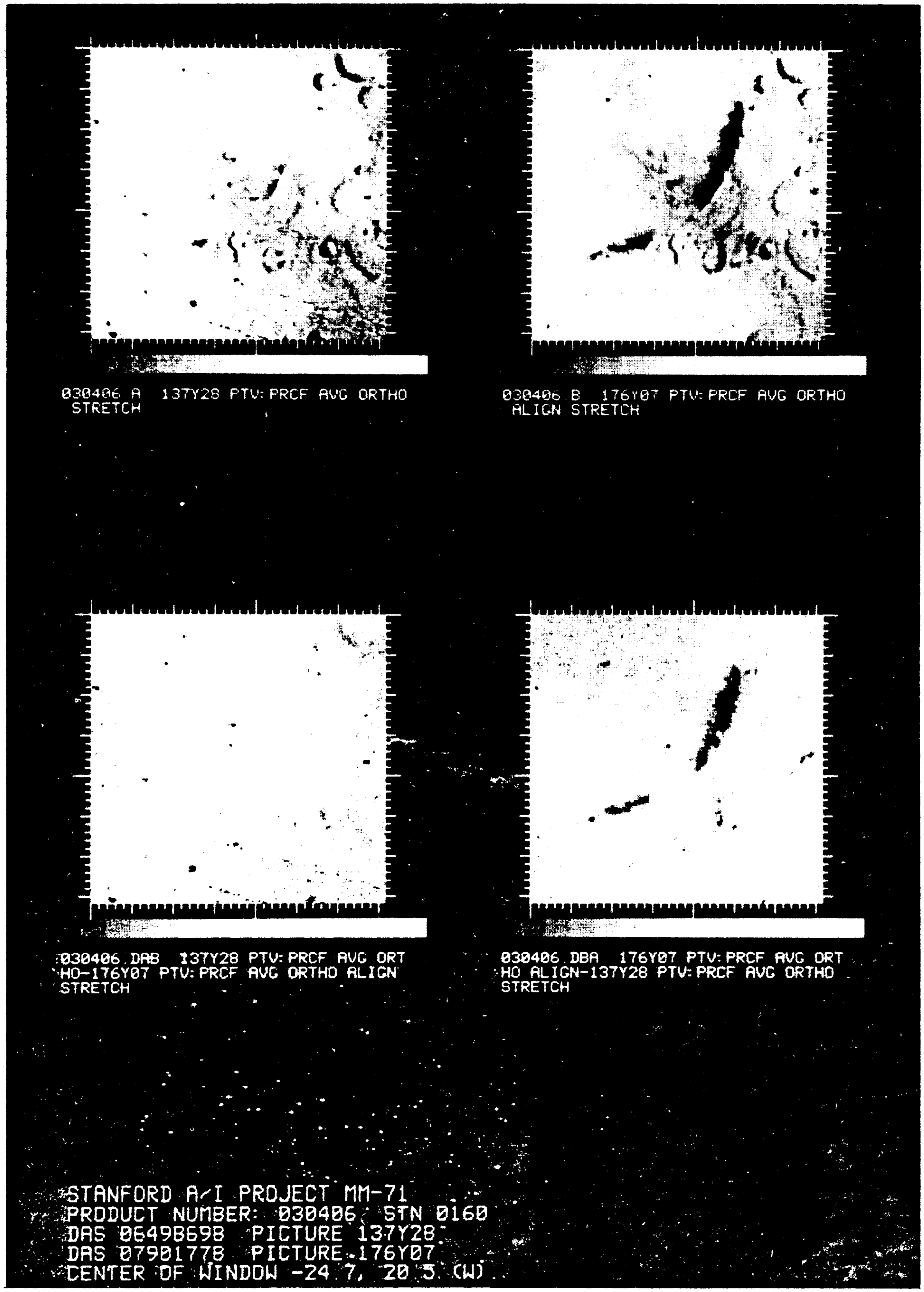
030406 A 137Y28 PTV: PRCF AVG ORTHO
STRETCH
030406 B 176Y07 PTV: PRCF AVG ORTHO
ALIGN STRETCH
030406.DAB 137Y28 PTV: PRCF AVG ORT
HO-176Y07 PTV: PRCF AVG ORTHO ALIGN
STRETCH
030406.DBA 176Y07 PTV: PRCF AVG ORT
HO ALIGN-137Y28 PTV: PRCF AVG ORTHO
STRETCH
STANFORD A/I PROJECT MM-71
PRODUCT NUMBER: 030406 STN 0160
DAS 06498698 PICTURE 137Y28
DAS 07901778 PICTURE 176Y07
CENTER OF WINDOW -24.7, 20.5 (W)

Figure 3

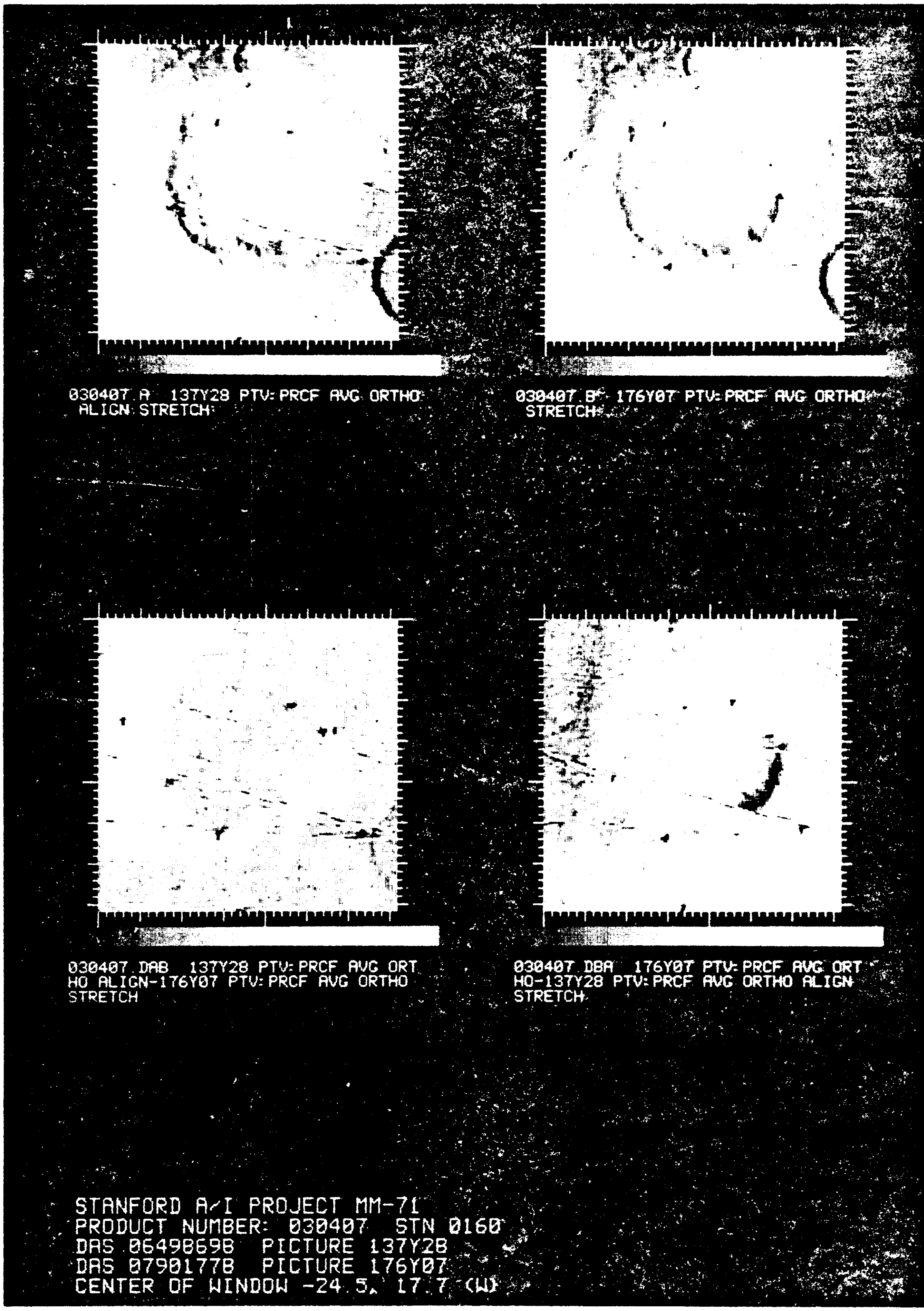
030407.A 137Y28 PTV: PRCF AVG ORTHO ALIGN STRETCH
030407.B 176Y07 PTV: PRCF AVG ORTHO STRETCH
030407.DAB 137Y28 PTV: PRCF AVG ORTHO ALIGN-176Y07 PTV: PRCF AVG ORTHO STRETCH
030407.DBA 176Y07 PTV: PRCF AVG ORTHO-137Y28 PTV: PRCF AVG ORTHO ALIGN STRETCH
STANFORD A/I PROJECT MM-71
PRODUCT NUMBER: 030407 STN 0160
DAS 06498698 PICTURE 137Y28
DAS 07901778 PICTURE 176Y07
CENTER OF WINDOW -24.5, 17.7 (W)

Figure 4

D. VIKING LANDER IMAGERY INVESTIGATION

The effort has been devoted to problems associated with the extraction of near-field ranging information from lander camera stereo image pairs. This latter work, which has used the facilities of the Stanford Artificial Intelligence Laboratory, is described below.

Figure 1 illustrates a stereo pair of images recorded of a portable terrain model by a lander Camera prototype. The figures displayed in this and the following illustrations are reproductions of Polaroid pictures taken of the output of a computer driven video-synthesizer. The upper and lower images represent left and right views, respectively, of the model as recorded by a single lander camera prototype located at two different positions 100 cm apart, approximately 135 cm above the model and at roughly 100-200 cm horizontal range. The model consists of an undulating surface of dark sand upon which have been placed four conspicuous rocks and some smaller pebbles. Since the horizontal range is comparable to the inter-camera displacement, henceforth called the "baseline", the two views of the scene appear appreciably different. It has been determined experimentally that it is impossible to visually fuse such disparate images, a prerequisite to visual depth perception.

Figure 2 indicates the same pair of images upon which have been overlayed two smooth curves. These curves have been constructed as follows. A Plane, henceforth referred to as the "camera-centers plane", is defined to contain the two effective camera-center positions. This plane has been rotated about the baseline until it passes through the selected point of interest in the left view at the center of the prepositioned box located directly beneath the central rock. This plane projects into the left and right camera views as the curves shown. The fact that the plane projects as a curve rather than a straight line is a consequence of the scanning geometry of the facsimile camera and of the associated display format. Specifically, the camera scans the scene, point by point, in uniform polar and azimuthal steps. The display format is linear in polar and azimuthal angles, and hence represents a linear mapping of the original recording. The lowest point in each of these curves corresponds to an azimuthal viewing direction perpendicular to the baseline at the camera location in question. Scene points lying on the camera-centers plane and common to both views must necessarily lie on the projections of this plane in each image.

Figure 3 illustrates the same image pair overlayed with image point location boxes, used in the scene ranging mode. The box in each view has been visually positioned to a scene point that is common to both views and is to be ranged. The positioning procedure is as follows. The box is first interactively centered about a point of interest in the left camera image. A camera-centers plane is then tilted to contain this pointing direction. The physical point of interest in the right view now must lie on the projection of the plane in that image. Thus, only a single degree of freedom remains for the definition of the corresponding box location in the righthand view. This latter parameter has been arbitrarily chosen to be the x-coordinate of the point, in image coordinates. The location of the matching point in the right view is determined visually. The box is then brought to position by the adjustment of the x-coordinate. The associated value of the y-coordinate of the point is then immediately evaluated, remaining parameter.

Once the corresponding pointing directions in the two scenes have been established, we have sufficient information to evaluate the spatial location of the selected point relative to the cameras. The ranging information is promptly computed and recorded and/or displayed.

Following development of the above programs we moved onto a familiarization study of some of the characteristics of the unique data format, preparatory to investigating automation of ranging and contour map production.

A first step in this process is indicated in Figure 4, which contains additional overlays to those discussed above. The oscillatory curves appearing in these views represent plots of the scene intensity scanned along the camera-centers plane and parameterized linearly in the x-coordinate. It will be noted that the lines of constant phase for the sand ripples running across the lower lefthand portion of the left view are roughly orthogonal to the baseline. These same waves, when viewed from the righthand camera position are observed obliquely and consequently exhibit foreshortened projected wavelength. The relative distortions associated with the grossly different perspectives of the the two views pose special picture point correlation problems for automation of ranging.

Figure 5 illustrates a remapping of the intensity plots of Figure 4 into a format that would be more amenable to a one-dimensional correlation of data from the two scenes. The intensity curves in Figure 5 have been obtained by transforming the curves in Figure 4 to the appearance they would have if recorded by a camera located at the mid-point between the left and right camera positions, under the assumption that the scene is perfectly flat, horizontal, and at the nominal value of observed relative elevation. The horizontal scale has been changed to utilize the full width of the screen. The mapping between the scene and the image now is nonlinear and the mutual interrelationship no longer is as readily discernable as in the previous illustrations. It is evident, however, that one-dimensional picture point correlation would be more readily conducted in this transformed image intensity space than in the initial space.

We do not plan to proceed further along the above lines.

Rather, we are considering a more general and powerful approach toward the development of near-field automated ranging, The proposal is to explore the portion of the 3-space mutually accessable through the left and right windows, by correlation of left and right imagery data over a variously tilted and positioned probing planar patch. Once it has been determined, by prescribed correlation criteria, that we have succeeded in "landing" on a surface, we can "crawl" over the surface in any manner desired, accumulating ranging information as we go, Elevation contour lines could for example be generated by instructing the probe to explore at fixed elevation. An automated contour map could thus be constructed,

We also are commmencing various image transformations directed both toward compensating for inherent projective distortions and toward facilitating the comparison both of images taken from the same camera under different recording conditions and of images taken from the two different lander camera positions. FIGURE CAPTIONS

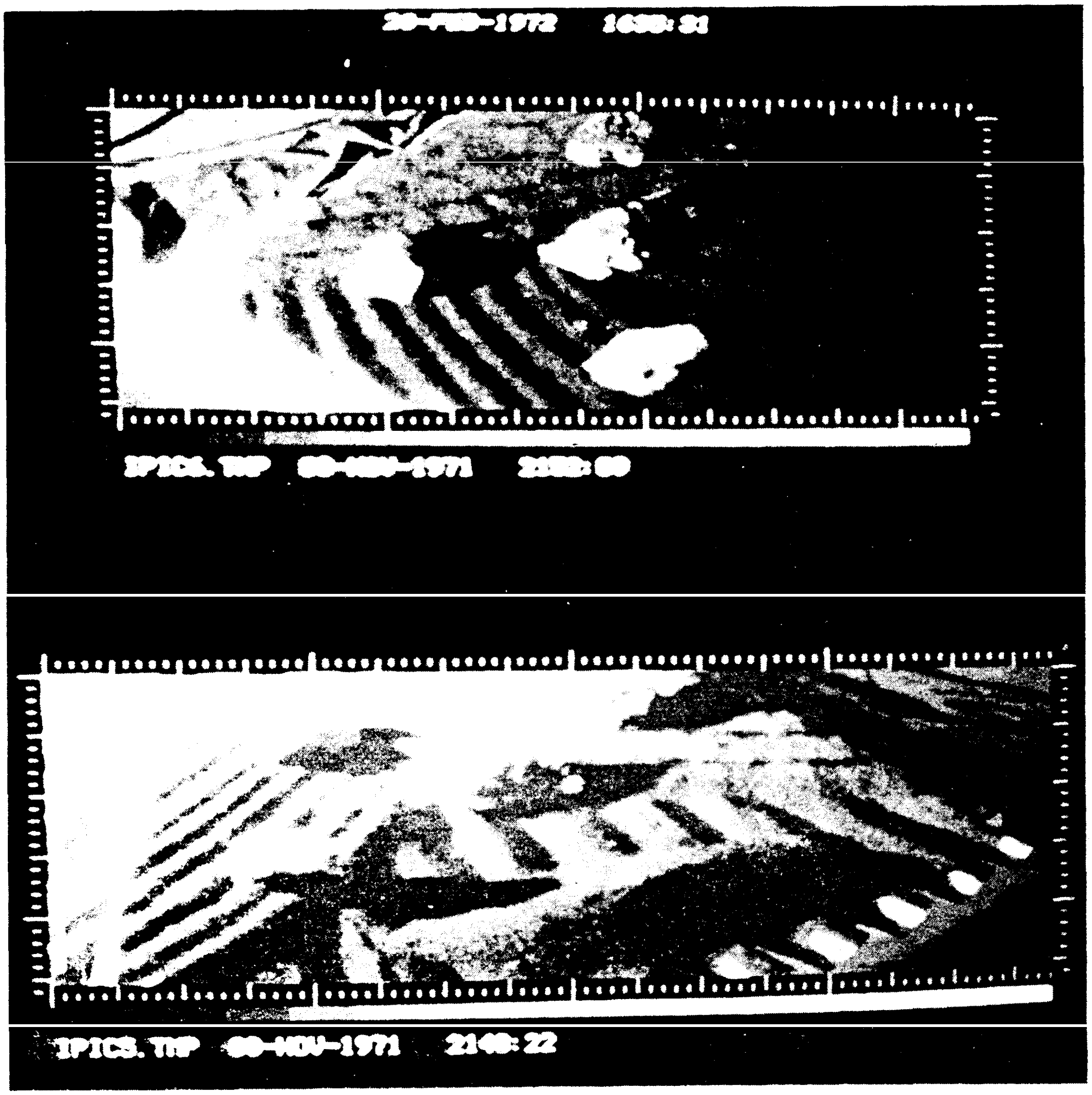

Figure 5. A stereo pair of images recorded of a portable terrain model by a single lander camera prototype located at two different positions. Upper image left view; lower image right view.

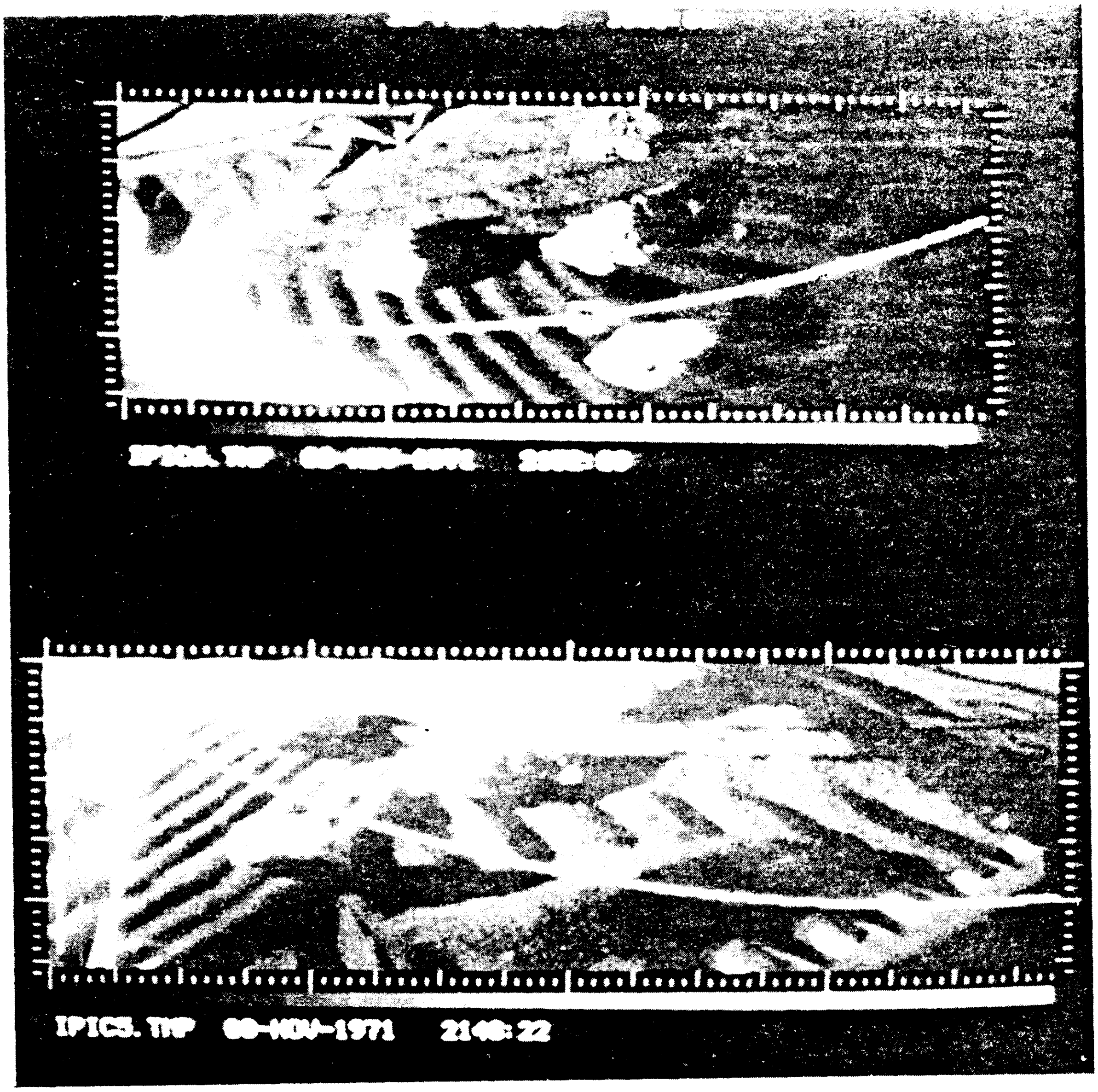

Figure 6. Overlay of a projection of the camera-centers plane onto the stereo images.

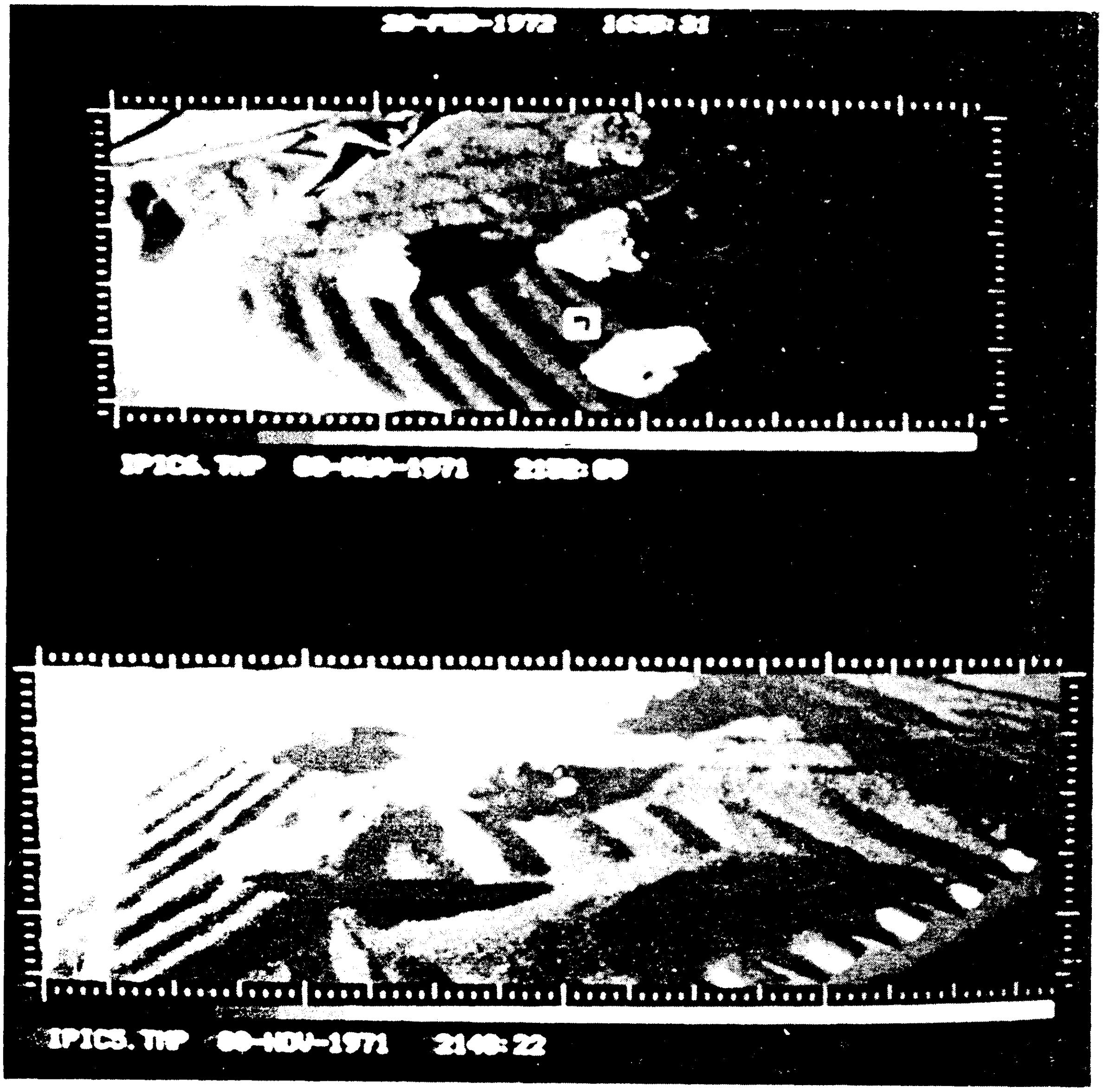

Figure 7. Stereo images overlayed with image point location boxes in the scene point ranging mode.

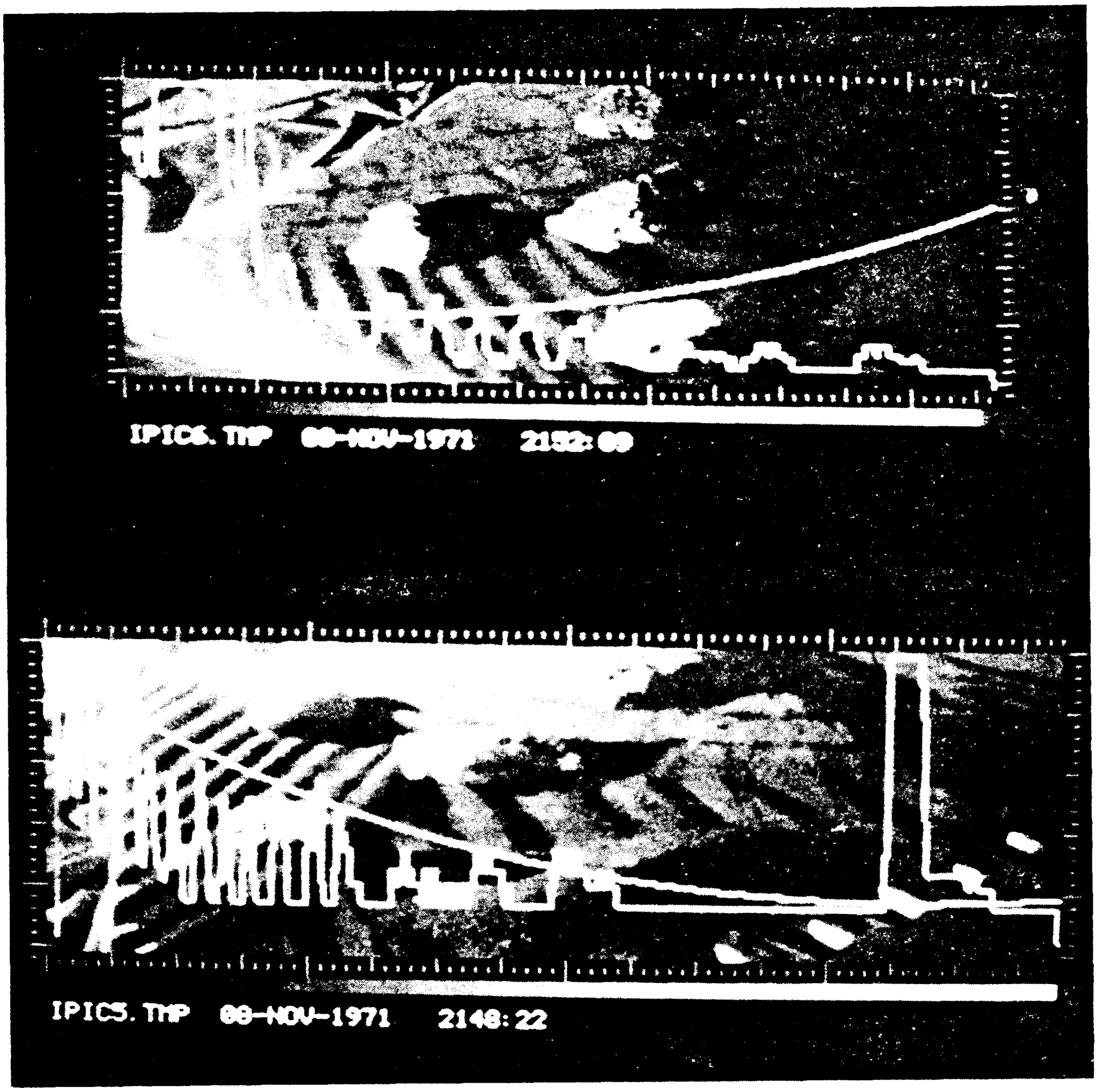

Figure 8. Overlays of the scene point intensity along the camera-centers plane as a function of the x-coordinate of the image point.

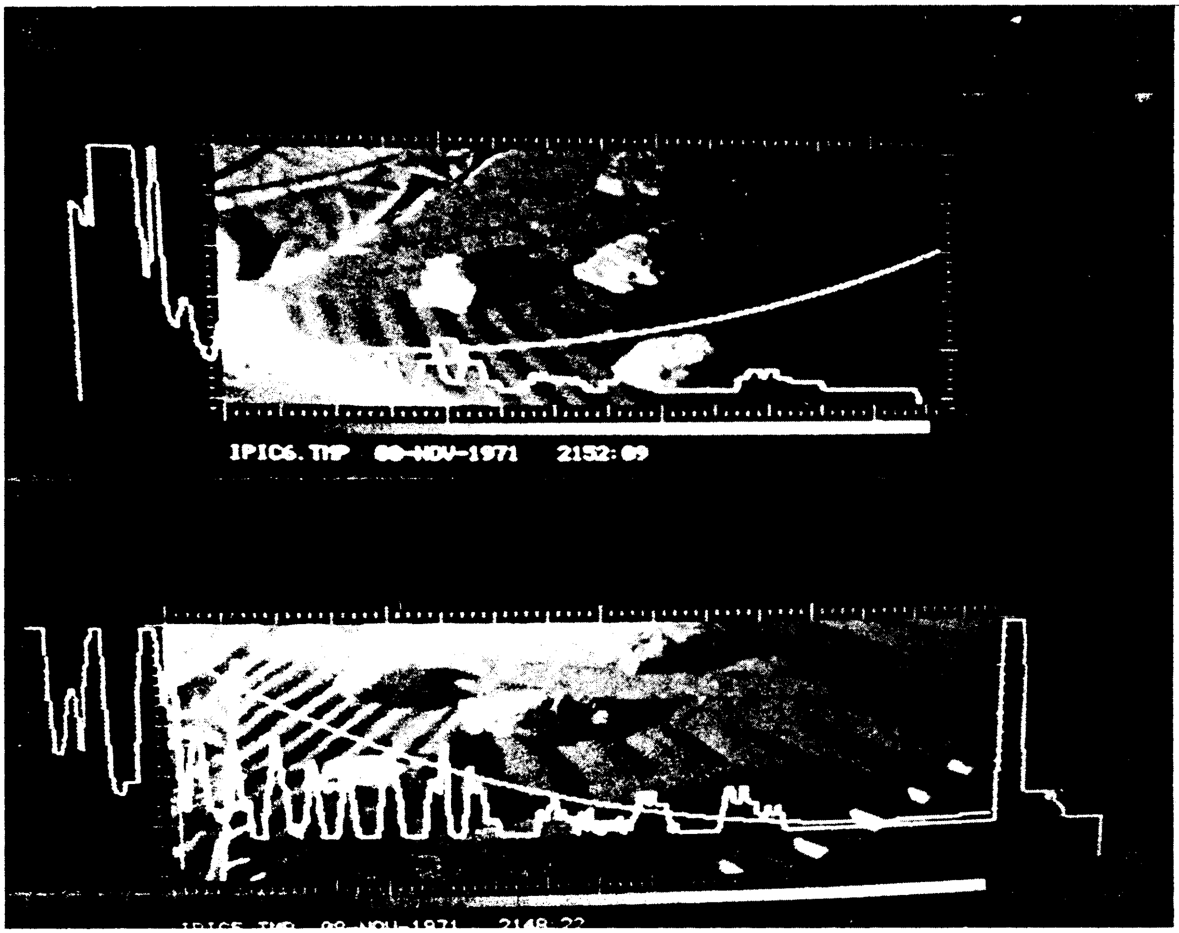

Figure 9. Overlays of transformed scene point intensities.

APPENDIX A

CAMERA MODELS AND STEREO IMAGE PROCESSING

Marsha Jo Hannah

Term Project
CS-293
Autumn Quarter, 1971

PREFACE

This paper reports progress made on a system of programs for processing medium-angle stereo images. The paper takes the form of documentation on what the separate programs written for this project do, with comments as to how they fit together. None of the descriptions are intended to enable the casual reader of this paper to use the programs involved. Anyone desiring to operate any of these programs is advised to contact the author for a demonstration.

INTRODUCTION

Suppose one were given two pictures of the same scene taken from moderately different viewing points. By moderately different is meant that the change in view point causes the pictures to differ by more than an infinitesimal amount but not by so much that an object present in both pictures is not easily recognizable as being the same object. Mathematically, this can be characterized by thinking Of the focal axes of the two cameras as vectors and describing α, the angle between these vectors. For the purposes of this report, $|\alpha|<\pi/8$ is a reasonable approximation to the Phrase "moderately different viewing points".

Given two such pictures, one would like to know how they relate to one another. How were the cameras that took them arranged with respect to each other? What clues are there in the two pictures as to size and position of the objects?

Several things are known to be undecidable given just the information in the pictures. Absolute position, for instance, is not derivable, that is, it is not possible to say precisely where in 3-space one of the cameras was or to give the exact three-dimensional co-ordinates corresponding to a given Point in a picture. Likewise, it is impossible to say exactly how large or how far away a given object is. Both absolute position and absolute size require knowledge not contained in the pictures.

It Is possible, however, to derive relative positions and relative sizes for objects in the pictures. This is done by assigning an arbitrary position and orientation to One Of the cameras and by fixing some distance, usually the baseline distance between cameras. From these starting points, the orientations of the cameras and positions of objects which appear in both pictures can be calculated.

This project, then, was a start toward automating the calculation of said relative orientations and positions given no more then two stereo views and a reasonable guess as to the baseline distance.

THE PROGRAMS

Work on this project was segmented into Separate tasks, each performed by an independent SAIL program. This Segmentation was forced, to some extent, by the fact that the system usually does not live long enough to support one long program. Thus it became advisable to have several small programs which depended on user interaction rather than one large program which would run by itself.

The basic sections and their functions are PARSET, which finds pairs of points using no outside information about the pictures, CAMERA, which uses point-pairs to find approximate camera models, and CAMSCH, which finds pairs of points using camera models.

PARSET

The purpose of this program is to find a set of pairs of points, one point out of each picture, which match. Intuitively, two points match if they both are projections of the same three-dimensional point. Computationally, the criterion for match is that the normalized cross-correlation between the 2n+1 x 2n+1 windows immediately surrounding each of the two points be high enough. Since the computational process can only say that point A matches point B with probability P, this program's purpose is to find pairs of points which match with fairly high probability.

As a preliminary to the matching process, this program segments both pictures into overlapping areas, usually 20 pixels square. It then computes the mean and variance of each area in each picture and sorts each picture's areas by variance, keeping track of where in the picture each area came from.

The matching process begins by selecting an area at random from the top end of the variance list of the first picture, usually the top 25%. This limitation is imposed because the measure of match being used-- normalized cross-correlation-- works best where there is a large amount of information present, which is symptomized by the variance being large.

Since areas which match should have similar variances, the selected area of the first picture is compared with each area of the second picture whose variance is within 20% of that of the area under consideration. (In the following, let the prefixes "first-" and "second-" stand for the modifying phrases "of the first picture" and "of the second picture" respectively.)

Each eligible second-area is initially tested to see if its mean is similar to that of the first-area. If a second-area passes this test, a search is made to find the second-point (some point in or near the second-area under consideration) such that the 2n+1 x 2n+1 window surrounding this second-point is the best match for (has the highest normalized cross-correlation with) the 2n+1 x 2n+1 window surrounding the center point of the first-area. The search strategy used is essentially that used by Quam(1971).

For computational expediency, the above search is carried out using a small window, typically 7 pixels square. As the program proceeds through the second-areas, the N second-areas (where N is usually 5) yielding the highest correlation values are kept track of and are re-searched using a larger area, typically 21 pixels square, to check that the areas do indeed match.

Tests are then made to determine whether the best match found was good enough, First of all, the correlation must be above .5, since a correlation lower than .5 can occur between areas which do not really match. Secondly, the top two correlations must differ significantly. Failure to do so would indicate that more than one match was possible, casting doubt on the validity of either match. Failure in either of these tests causes a first-area to be rejected as having no reliable match, and another first-area is tried.

Note that, when this process is finished, the center point of the first-area has been paired with a second-point which has integer co-ordinates. In practice, however, the proper match for a given first-point will be a second-point with non-integer co-ordinates. Since the only correlation values which are available are those at integer second-points, some form of interpolation is necessary.

Therefore, the final operation on a match is an interpolation. A function of the form EXP(- (A*X↑2 + B*X + C*X*Y + D*Y↑2 + E*Y + F)) is fitted by least squares techniques to the correlation values between the window around the first-point and similar windows around points in the neighborhood of the second-point. Solving this function for a maximum results in either a new match at some non-integer second-point, or in an error if there is no maximum within a one-pixel radius of the matching second-point.

In the latter case, the first-area is said to have no reliable match, and the program continues to another first-area. The most common cause of such failure is a strong linear edge with little information on either side, in which case the chances of error are sufficient to cast any such match in doubt.

If the match passes this final test, it is recorded for use in a later program, and this program proceeds to another first-area.

CAMERA

The job of this program is to find camera models. A camera model consists of seven numbers which specify the focal lengths of the two cameras and the orientation of the second camera with respect to the first.

The first camera is taken to have its focal point at the origin, its focal axis along the z-axis, and its image plane the plane z=F1. (See Illustration 1.) The focal point of the second camera is a point which is described by the baseline distance and two angles. The two angles are the angles by which the first camera must be panned, then tilted, to point at the second focal point. (See Illustration 2.) The focal axis of the second camera is described by two more angles. They are the angles through which the first camera must be panned, then tilted so that its axis parallels the axis of the second camera. The image plane of the second camera is the plane

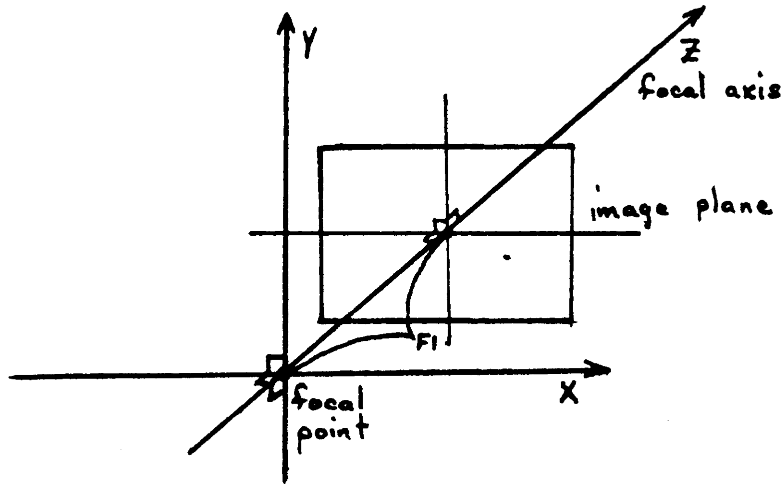

Illustration 1.

Arbitrary co-ordinate system with first camera in place.

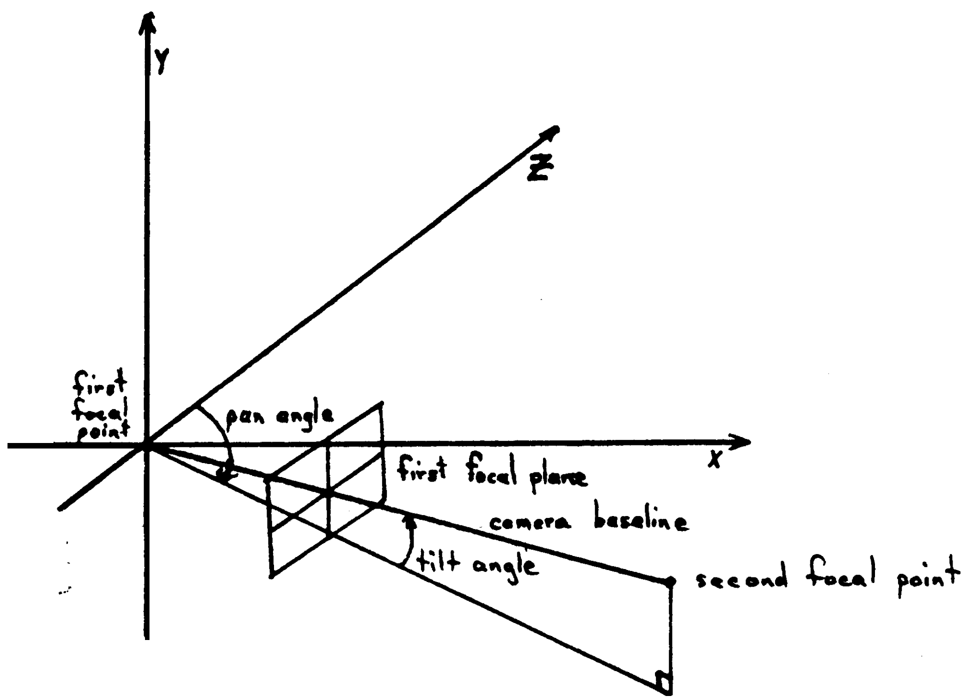

Illustration 2.

Co-ordinate system with first camera panned and tilted to locate focal point of second camera.

perpendicular to the focal axis at distance F2 from the focal point. (See Illustration 3.) The orientation of the second image plane is described by the angle through which the first image plane must roll (after having been panned and tilted to make the axes parallel) in order to bring the two "up" directions into agreement.

This program takes as input a set of pairs of first-points and second-points found to be matches and attempts to find a camera model which would account for these point-pairs. Determination of a model is done by minimizing a measure of camera model error.

The error measure is the average error in match taken over the point pairs. For each point pair, the error in match is determined as follows: Using the camera model, the first-point is projected into space, yielding a ray from the focal point through the image point. This ray is then back-projected into the image plane of

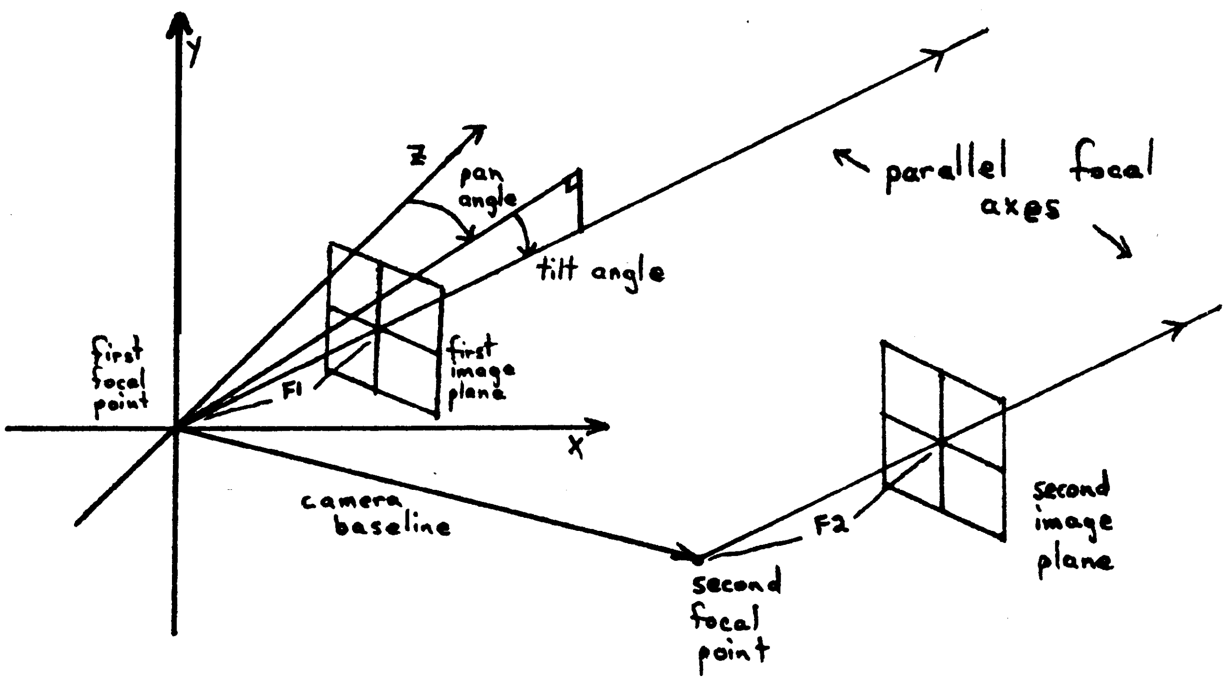

Illustration 3.

Co-ordinate system with second camera in place, first camera panned and tilted so its focal axis parallels that of the second camera.

the second camera, visioing a line segment in the second image. The error is taken to be the square of the distance between the second-point and this line segment, in the usual mathematical sense. (See Illustration 4.)

Actual minimization of the error function is carried out by the independently compiled subroutine MINIMZ.

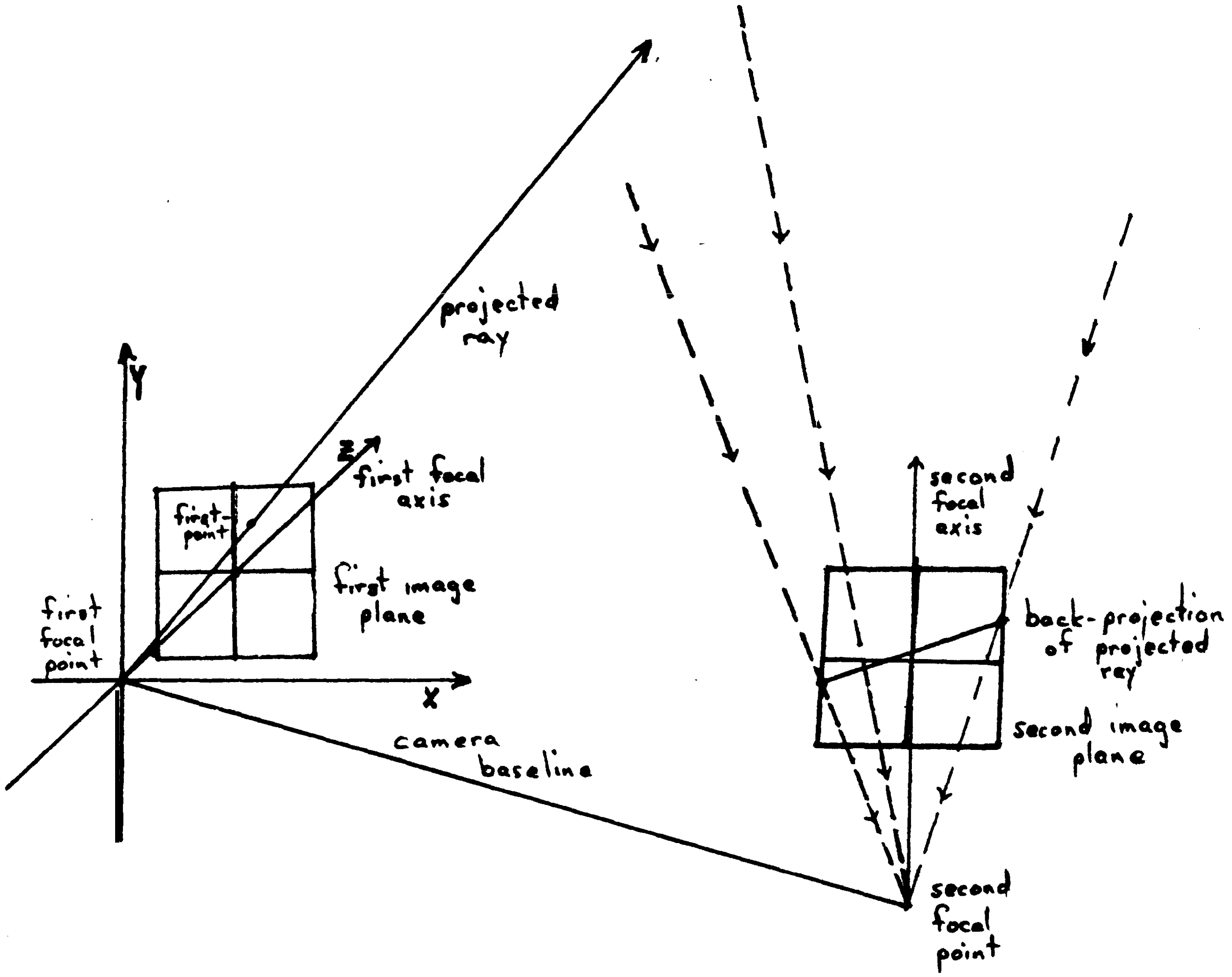

Illustration 4.

Co-ordinate system with both cameras in place, showing a first-point projected into space and the resulting ray back-projected into the second image.

MINIMZ

This sub-program is a function minimizer which uses no derivative information in seeking a minimum. Not using derivative information was a constraint forced by three considerations--the fact that the derivatives of the camera model error function are discontinuous since the function has been truncated to avoid floating point overflows in the calculations, the fact that the locations of these discontinuities are not precisely known, and the fact that the camera model error function itself does not obey any of the constraints (monotonicity, lack of local minima, etc.) usually placed on functions to be minimized by derivative methods.

The workhorse function of this sub-program takes an n-dimensional vector (in the camera model case, n = 7) and finds 3 starting points along this vector such that an upward-facing parabola can be fitted to the three points. The inner loop fits the parabola, finds the minimum of this parabola, evaluates the function at the parabola minimum, and chooses which of the four available points are "best" to fit another parabola to. This continues either until the specified number of cycles (usually 10) have been completed or until successive parabola fits yield the same function value, within a tolerance (usually .00001).

The outer loop of this sub-program constructs a set of orthonormal vectors (starting with the co-ordinate axes) and calls the workhorse function described above along each of these vectors. When this set of vectors is exhausted, the outer loop then calculates the vector difference between the starting point for that round and the final point found. This vector is then used in constructing a new set of orthonormal vectors. This iteration continues until the difference in starting and finishing function values is less than the given tolerance (as above) or until the function value drops below some pre-set limit.

Like all minimizers, this one can get stuck at a local minimum. This is unfortunate, because the camera model error function appears to have a large number of potential local minima near the actual minimum. However, it also appears that having more points available for use reduces the number and depth of local minima, reducing the chance of a spurious minimum being taken as the actual [illegible] This suggests iterating CAMERA with the following program, CAMSCH, to get better and better camera models based on more and more points.

CAMSCH

This program takes as input a camera model either derived by CAMERA or supplied by some other means. To match a designated first-point, this program projects the first-point into space, back-projects the ray so produced into the seoond image, then searches along the resulting line. (See Illustration 5.)

The actual search consists of stepping along the line, starting from the infinity point (the point of the line corresponding to the point on the projected ray which is farthest away from the first camera, but still visible to the second camera). At each step along the line, the correlation between the window around the first-point under consideration and the corresponding window around the second point currently under scrutiny is calculated. As stepping continues, the best N such correlations (again, N is usually 5) are kept track of and a local search for maximum correlation is done in the neighborhood of each such second-point.

Testing whether or not the match is sufficiently good and interpolation are similar to the same processes described for PARSET.

One search-pruning heuristic used in both PARSET and CAMSCH needs, perhaps, to be justified. In PARSET, if the center point of a given second-area does not show a positive correlation with the

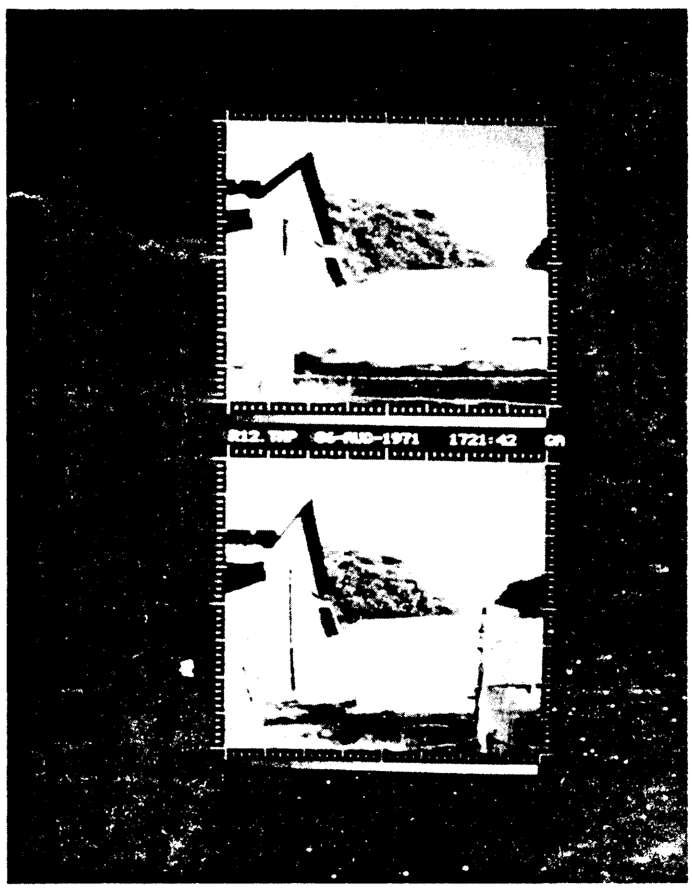

Illustration 5.

A pair of stereo views showing a point and its surrounding correlation window overlaid on the top image and the line that the point projects to overlaid on the lower image. Polaroid picture taken of Data Disc while CAMSCH was in operation.

center point of the first-area, the second-area is rejected as being an improbable place to look for a match. CAMSCH, Instead of examining every point along the line projected into the second image, examines every N-th point, where N is half the radius of the correlation window being used. Thus, in both programs, decisions are made on the basis of the value of the correlation function at some second-point near (but not at) the matching second-point. This, It turns out, is a reasonable thing to do, Taking any cross-section of the correlation function will yield a graph shaped somewhat like EXP(-X↑2). Because of this, a correlation at a second-point near the match will be fairly high--at least, above the noise level. Hence a very low correlation value can be taken to mean that there is no match in the vicinity of the point under consideration, and the computation necessary to search that area can be avoided,

It is interesting to note that different programs require different degrees of accuracy from their camera models.

CAMSCH, for instance, with its local search strategies, can get by with rather inaccurate camera models. Any camera model which will put CAMSCH in the right ballpark is good enough to produce matches for most points. Experimentation has shown that almost any camera model having an average squared-error below .25 pixels is good enough for matching at least 50% of the points tried, This would suggest not spending great amounts of time minimizing the first camera model, as has been done in most cases tried so far, Instead, an inaccurate model can be derived quickly, CAMSCH can use this to find more points, which can be used to derive a better camera model, etc.

On the other hand, programs which do depth modelling require extremely accurate camera models. For instance, on one pair of pictures, about 25 different camera models were derived. Their squared-error varied from .2 to .0004. Depths given for one point at measured distance 320 feet from the camera ranged from 25 feet to 2500 feet. In general, models with smaller squared-errors were better than those with larger errors. However, the best model was not the one with the lowest error! This would indicate that the models being found all represent local minima on the error function; the true model has yet to be found,

This result led to further modification of the minimizer, In hopes of being able to get closer to the real minimum. As yet, this is not possible. Accurate depth ranging remains a hit-or-miss thing, depending on whether or not a model can be found which is good enough.

There are several other minor programs intended for demonstrations which fit into this set of programs.

WINSHO simply takes a file containing point-pairs and displays them, pair at a time, as overlays on pictures shown on Data Disc.

DEPTH works similarly to WINSHO, except that it also asks for a camera model, and calculates the distance from the first camera to the three-dimensional point. It then displays this distance on the Data Disc screen with the appropriate overlays representing the two points. At the end, this program rounds each depth to the nearest unit of distance and displays these distances as overlays at tha corresponding points in the first picture. (See Illustration 6.)

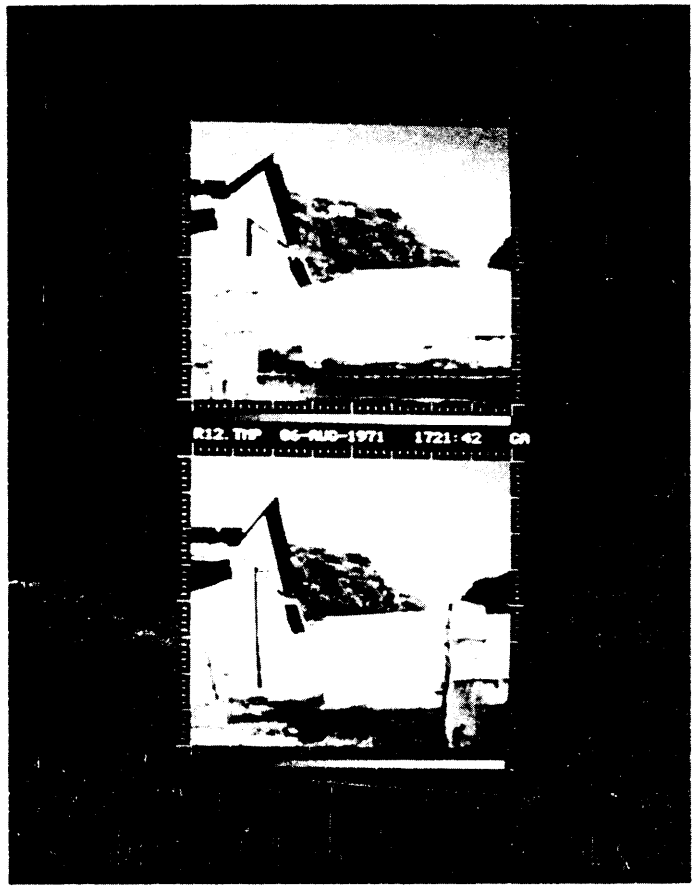

Illustration 6.

Photograph from Data Disc of overlaid depths as generated by DEPTH.

CONCLUSION

There are a number of possible variations on this set of programs.

One possible alteration would be to incorporate color information into the matching process. This change would require using three aligned color-filter pictures, and modifying the correlation function so that it uses a vector of information at each picture point rather than a single scalar number. This technique would give more information at each point, making gross mismatches less likely.

Another interesting idea would be to use the camera model to shape the correlation window, making correlation less sensitive to distortions of an object due to the differences in projection. Thus, a camera model having most of the change in the horizontal direction would be made to cause a correlation window which was tall and thin--using more information in the direction in which little distortion occurs, less in the distorted direction. (See Illustration 7.)

Of course, much can be done to automate these programs. At present they often stop and ask for guidance when there is doubt as to whether or not a match is good.

In PARSET, where all matches must be good, strengthening the criterion for a match is one way to do this--when in doubt, throw it out. Another method of insuring that CAMERA gets only good point-pairs is to allow CAMERA to weed point-pairs given it. If, when a minimum of sorts is found, one or more of the point-pairs are found to contribute abnormally high (or low) errors, these pairs could be rejected, and a re-minimization done.

Theoretically--that is, given enough time and some low cleverness--it is possible to use CAMSCH to find a matching point for every first-point which has a match. Under this assumption, the technique of parallaxing (creating mappings from one picture to another, and finding parallax edges is now feasible. Assuming good camera models, the possession of such a mapping makes depth modelling and the location of depth edges possible.

It seems, then that the next moves on this project will be in the direction of improvements to existing programs. Specifically, the low running necessary to form the matches needs to be developed. The programs documented here need to be optimized so they can run in an amount of time agreeable to the machine. Most of all, the camera model programs need to be improved so that more reliable camera models are possible.

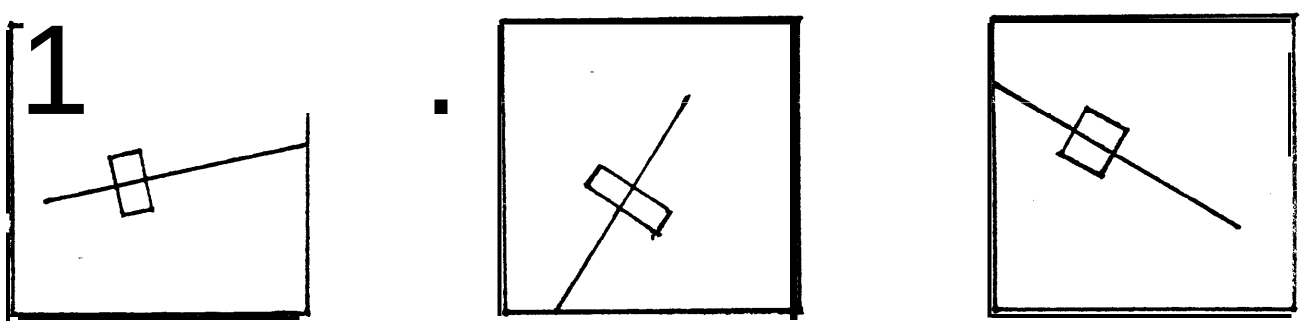

Illustration 7.

Various "second-pictures" showing back-projected first-points and the appropriately shaped correlation windows (size exaggerated).

BIBLIOGRAPHY

Quam, Lynn H. [1971], "Computer Comparison of Pictures", Stanford Artificial Intelligence Project Memo No, 144,

OTHER REFERENCES

Sobel, Irwin [1970], "Camera Models and Machine Perception", Stanford Artificial Intelligence project Memo No. 121,

Temes, G. C., and Calahan, D. A. [1967], "Computer Aided Network Optimization--The State of the Art", Proc. IEEE, Vol. 55, No. 11, Nov. 1967, pp1332:63.

COLOR INFORMATION IN STEREO IMAGE PROCESSING

Marsha Jo Hannah

Term Project
CS-293
Winter Quarter, 1972

INTRODUCTION

My project for this quarter was to start implementation of a system for processing color stereo pairs, similar to my system for processing black and white stereo images (see Hannah, 1971, for details of that system). Since the black and white system was built around the idea of using normalized cross-correlation as a measure of match between two points, the first thing that was needed for the new system was some equivalent measure of match for color images. (Actually, correlation is a measure of match between two areas; the two points refered to are the centers of the areas. In the following, the phrases "between two points" and "between two areas" will be used interchangeably in referring to correlation).

Basically, I had the choice of somehow altering correlation for use with color information or creating an entirely different measure of match. Having had little luck in an earlier attempt to find a new measure of match for the black and white case, I chose to modify correlation.

This document reports the derivation and implementation of color correlation, and describes a program, NEWPTS, which finds initial point-pair matches in either color or black and white stereo pairs.

COLOR CORRELATION

It is generally recognized that color consists of three components. A child learns in grade-school art that all colors can be made from red, yellow, and blue pigments. In high-school physics, he is told that all colors result from red, green, and blue light. In college psychology courses, color is discussed in terms of intensity, hue, and saturation.

Ignoring for the moment the thorny questions of what the components of color "really" are, we shall admit only that there are three such components. Since the color images we currently are working with were obtained by digitizing three black and white pictures which resulted from photographing an ordinary color slide under red, green, and blue filters, respectively, we shall refer to the components as R, G, and B.

It is somewhat more convenient (as well as more mathematical!) to think of a color picture as one array of vector-valued points (r,g,b) instead of three separate arrays of scalar-valued points r, g, and b. This suggests regarding the text-book-version of normalized cross-correlation

```
              SUM( (x-MEAN(x)) * (y-MEAN(y)) )
COR = ---------------------------------------------
      SQRT( SUM( (x-MEAN(x))↑2 ) * SUM( (y-MEAN(y))↑2 ) )
```

(where small letters denote sample elements, SUM is the sum over some set of such elements, MEAN is such a sum divided by the number of elements summed over, SQRT is the square root function, and * denotes multiplication)

as the one-dimensional case of a vector function

```
                SUM( (X-MEAN(X)) . (Y-MEAN(Y)) )
VCOR = -------------------------------------------------------
       SQRT( SUM( |X-MEAN(X)|↑2 ) * SUM( |Y-MEAN(Y)|↑2 ) )
```

(where capital letters denote vectors, • is vector dot product, and |A| is the norm of the vector A),

Considering only the factor SUM((X-MEAN(X)) • (Y-MEAN(Y))) (since SUM(|X-MEAN(X)|↑2) and SUM(|Y-MEAN(Y)|↑2) are both special cases, of this SUM with X substituted for Y in the first case and Y substituted for X in the second) and letting X be (xr,xg,xb) and Y be (yr,yg,yb), we have

```
SUM( (X-MEAN(X)) • (Y-MEAN(Y)) )

   = SUM( ((xr,xg,xb)-MEAN( (xr,xg,xb) )) •
                         ((yr,yg,yb)-MEAN( (yr,yg,yb) )) )

   = SUM( (xr-MEAN(xr),xg-MEAN(xg),xb-MEAN(xb)) •
                         (yr-MEAN(yr),yg-MEAN(yg),yb-MEAN(yb)) )

   = SUM( (xr-MEAN(xr))*(yr-MEAN(yr))+(xg-MEAN(xg))*(yg-MEAN(yg)) +
                         (xb-MEAN(xb))*(yb-MEAN(yb)) )
```

If we cleverly notice that all three terms within this sum are the same in form and combine them into one term under a summation which sums over all components as well as all elements of components, we get

```
   = SUM( (x-MEAN(x)) * (y-MEAN(y)) )
```

which is the representative factor of the formula for ordinary correlation (see above formula for correlation).

It is convenient (It somewhat embarassing) to have color correlation turn out to be a dressed up form of ordinary correlation, for this means that color correlation has all of the mathematical properties of ordinary correlation, this, in turn, will be particularly useful in the interpolation of correlation values at non-integer points in the picture, since previously developed techniques for such interpolation need not be justified again.

Since both vector and Scalar correlation have the same form, save for the number of components which need be summed over, the two brands of correlation have been implemented as one subroutine, CORLAT, in the SAIL load-module SCOREL. Which calculation is done depends on the global flag COLOR, For expediency in computation, the' coefficient is calculated as

$$= \frac{(n * SUM(x * y) - SUM(x) \cdot SUM(y))^2}{(n * SUM(x^2) - SUM(x)^2) \cdot (n * SUM(y^2) - SUM(y)^2)}$$

that is, with sums arranged so that only one pass need be taken to calculate ail sums, Note, too, that the square of the correlation is used (as It was in the black and white case), trading a multiplication for a call on SQRT.

No modifications (other than a small amount of optimizing) have been made on the functions MATCH and MAXCOR which call CORLAT and live In SCOREL. A separate program, NEWPTS, has been created to serve the function of PARSET in the color case and replace PARSET in the black and whit8 case, It is described below as if it operated in color mode, only,

NEWPTS

The purpose of this program is to find a set of pairs of points, one point out of each picture, which match, Intuitively, two points match if they both are projections of the same three-dimensional point, Since the computational process can oniy say that point A matches Doint B with some probability, this program's purpose is to find pairs of points which match with fairly high probability.

As a preliminary to the matching process, this program segments both pictures Into overlapping areas, usually 20 pixels square, It then computes the mean and variance of each area in the first component of each ooior picture (only one component is used to expedite computation) and sorts each picture's areas by variance, keeping track of where in the picture each area came from,

The matching process begins by selecting an area at random from the top end of the variance list of the first picture, usually the top 25%. This limitation is imposed because the measure of match being used works best where there is a large amount of information present, which is symptomized by the variance being large,

Since areas which match should have similar variances, the selected area of the first picture is compared with each area of the second picture whose variance is within 20% of that of the area under consideration, (In the following, iet the prefixes "first-" and "second-" stand for the modifying phrases "of the first picture" and "of the second picture" respectively.)

Each eligible second-area is initially tested to see if its mean is similar that of the first-area, If a second-area Dassts this test, a search is made to find the second-point (some point in or near the second-area under consideration) such that the 2n+1 x 2n+1 window surrounding this second-point is the best match for (has the highest normalized cross-correlation with) the 2n+1 x 2n+1 window surrounding the center point of the first-area. The search strategy used is essentially that used by Quam (1971).

For computational expediency, the above search is carried out using only the first component of the color picture. As the program proceeds through the second-areas, the N second-areas (where N is usually 5) yielding the highest correlation values are kept track of. Later, a second search is done on these areas using color correlation to determine which of the areas that matched on the basis of the one-component search match best in color.

Tests are then made to determine whether the best match found was good enough. First of all, the correlation must be above .5 (calculated square of the correlation above .25), since a correlation lower than .5 can occur between areas which do not really match. Secondly, the top two correlations must differ significantly. Failure to do so would indicate that more than one match was possible, casting doubt on the validity of either match. Failure in either of these tests causes a first-area to be rejected as having no reliable match, and another first-area is tried.

Note that, when this process is finished, the center point of the first-area has been paired with a second-point which has integer co-ordinates. In practice, however, the proper match for a given first-point will be a second-point with non-integer co-ordinates. Since the only correlation values which are available are those at integer second-points, some form of interpolation is necessary.

Therefore, the final operation on a match is an interpolation. A function of the form EXP(- (A*X↑2 + B*X + C*X*Y + D*Y↑2 + E*Y + F)) is fitted by least squares techniques to the color correlation values between the window around the first-point and similar windows around points in the neighborhood of the second-point. Solving this function for a maximum results in either a new match at some non-integer second-point, or in an error if there is no maximum within a one-pixel radius of the matching second-point.

In the latter case, the first-area is said to have no reliable match, and the program continues to another first-area. The most common cause of such failure is a strong linear edge with little information on either side, in which case the chances of error are sufficient to cast any such match in doubt.

If the match passes this final test, it is recorded for use in a later program, and this program proceeds to another first-area.

ADDITIONS AND IMPROVEMENTS

Our present practice of using red, green, and blue as the three components of the color picture has its drawbacks. One would like to calculate the means and variances of the average intensities in the three components for the purpose of narrowing the searches for the initial point-pair matchings, To do this under the present scheme, one must either calculate the averages as one goes-a rather slow process--or keep around an extra Daft of pictures, the intensity pictures--a scheme which enlargens (hence slows) one's job excessively, A solution to this Problem would be to have the intensity picture be one of the three components of the color picture,

There are at least two schemes of color representation which have intensity a 9 one component, The best known, perhaps, is the intensity, hue, and saturation scheme, Commercial television uses the different, although related, scheme of intensity, x-, and y-chromaticity, Both of these are besed on the idea of a color wheel, ie, colors arranged circularly around a hub, The hue-saturation Scheme corresponds t o using polar co-ordinates to nocate a color Point on the wheel, The x-, y-chromaticity soheme corresponds to using a (not necessarily rectangular) Cartesian co-ordinate system to locate the color point,

Implementing one of these systems seems to be desirable, Precisely which one to implement and how to derive these components from the given red, green and blue components is a matter for further study,

Once the representation question has been settled, there are a number of statistics which can be oaloulated over the segmented pictures, In addition to the mean and variance of the intensity, one could calculate the mean and variance of the color, the mode (most frequently occuring) color, possibly the next most frequent color, etc, Each area in each picture would then be associated with a vector of such statistics, and searches for a matohlng second-area could be constrained to those second-areas whose vector distance from the first-area is within some tolerance,

A more thorough investigation of the properties and representations of color seems to be in order, It is in this direction that this project will Proceed.

BIBLIOGRAPHY

Hannah, M. J. [1971], "Camera Models and Stereo Image Processing", Term Project Report, CS-293, Stanford University, Fall, 1971.

Quam, Lynn H. [1971], "Computer Comparison of Pictures", Stanford Artificial Intelligence Project Memo No. 144,