General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

(NASA-CR-147117) COMPRESSION OF CONTOUR DATA THROUGH EXPLOITING CURVE-TO-CURVE DEPENDENCE (Brown Univ.) 21 p HC $3.50 CSCL 09B

N76-22959

Unclas

G3/63 26841

COMPRESSION OF CONTOUR DATA THROUGH EXPLOITING CURVE-TO-CURVE DEPENDENCE

Nese Yalabik[†]

David B. Cooper[†]

MAY 1976
RECEIVED
NASA STI FACILITY
INPUT BRANCH

[†] Division of Engineering, Brown University, Providence, RI 02912

Work supported by NASA Grant NSG 5036

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

Abstract

Many classes of line drawings such as contour maps appearing in weather data representation (e.g., isobars and isotherms), seismograph data etc., exhibit structure where 2, 3, or even more adjacent curves are very similar to one another. These pictures thus exhibit strong two dimensional dependences--along curve dependencies which show up as smooth curves with slowly changing curvatures, and the afforementioned curve-to-curve dependencies. In this work we discuss an approach to exploiting these dependencies in order to achieve high data compression. We take as a starting point one of the approaches to date of along curve compression through use of cubic spline approximation and extend this by investigating the additional compressibility achievable through curve-to-curve structure exploitation. We report on one of the models under investigation, namely that in which a curve is fairly well approximated as a linear transformation of an adjacent curve. This transformation includes any possible combination of a curve translation, two dimensional stretching, and rotation about an arbitrary axis. Hence, this coding might in concept be thought of as a linear least mean square error predictive coding in which rather than predicting a new data point based on past data in a time series, the system is predicting a new curve based on preceding curves in a region. In practice formidable diff~~er~~i~~ences~~culties exist because of lack of an obvious indexing parameter and lack of any usable statistical stationarity. Much weather map data consists of nested curves (representing hills), and for typical examples of this data we show that compression ratios of 1.5 to 2.0 above that achievable with spline approximations to individual curves are possible. This compression is achieved through use of curve-to-curve dependencies in encoding knot points and no additional error is introduced. If some additional error is permitted, greater compression is possible.

I. INTRODUCTION

Image data compression has become an extensive research area since computers have been more and more available for automatic storage, processing and transmission of pictorial data. This includes TV pictures used in space program, biomedical data, weather maps and cloud pictures in meteorology etc. Conventional method of scanning an image for transmission (for storage) generates a data sequence which usually contains considerable redundancy. So here the data compression problem is that of representing two dimensional data in a most efficient way without loosing any necessary information.

In most of the work done on the subject, data sequences generated by line scanners are compressed by methods like run-length coding, transform coding, predictive coding, etc. [1-3]. For all of the above methods, statistical dependencies between successive symbols in line scan is exploited. For many applications, it is possible to use the theory very successfully because data can be well defined as a stationary stochastic process.

It is possible to get much higher compression ratios if two dimensional structures inherent in pictorial data are recognized. In that case, data sequence to be encoded can be a set of parameters representing lines, curve segments, etc. [4], or it can be the sequence of coordinates of points in contour lines in a picture [5]. Recently, Pavlidis [6] represented certain kinds of pictures and maps by piecewise linear surfaces.

In this paper a two dimensional predictive coding method is applied to a class of line drawings which consist of groups of similar looking smooth curves such as appear in Weathermaps, seismographs etc. (Section II). The essential information for representing each curve is a number of knot points and the remainder of each curve is then reconstructed through use of cubic splines. In sections III and IV, we discuss a compression method applied to a group of closed curves (Figures 3, 4). Instead of encoding each knot point on each curve by picture coordinates, it's location is predicted by exploiting the similarities between curves. It should be mentioned that, even though real data does not everywhere consist of closed curves as discussed in sections III and IV, the method can easily be extended to open curves. In case of single curves which do not look like any other curve in the picture, only the information contained on the same curve can be used for prediction.

Predictive coding method used here is not a standard theoretical procedure (such as prediction using an autocorrelation function) since for data such as we deal with, standard stationary data sequence methods do not appear to be applicable because the number of degrees of freedom of these patterns is too small for the concept of stationarity to be meaningfully applied. However, linear transformations (section III) can be thought of as playing a structural description role here. Furthermore, since the linear transformation which is appropriate differs from one region of a picture to another, we have a situation analogous to that of a quasistationary time sequence where the autocorrelation function is different for different intervals and must be described to the receiver if it is to be used for purposes of data compression.

Since a picture contains regions exhibiting a number of different types of data structures --- the nested contours treated in this paper being one type --- pattern recognition plays the important role of segmenting the picture field into regions each of which can then be encoded in the most appropriate way.

II. WEATHERMAPS

2.1 Previous Work on redundancy removal for weathermaps

Figures 1 and 2 are typical examples of the meteorological maps that are constantly transmitted throughout weather stations for forecast purposes. The usual data sequence to be transmitted for these pictures is a sequence of binary symbols representing a line by line scan of a two gray level picture. There has been some work on efficient coding of these strings such as [1], where constant output bit rate was used, and [2] where run-length coding schemes for source coding were used. The best reduction factor is about in these schemes.

More recently, conic curves have been used to describe and transmit weather maps. The contours of constant pressure and temperature were approximated by conic curves such as a segment of a parabola or ellipse [3]. The parameters corresponding to these curve segments are encoded. Such a system is claimed to be capable of compressing the data substantially, often by two orders of magnitude.

2.2 Picture Structure

The structure that is observable in Figures 1 and 2 arises in some other applications as well, such as in geographic elevation mapping, cartography and seismology. In all of these drawings, there is more structural redundance than a contour being representable by a number of conic curves: a) By being smooth and slowly changing, a piece of a single curve carries information about the other parts of the same curve. That fact has been used before [4] for redundancy removal purposes. b) There are groups of nested curves that show very similar structures. (Some examples are circled in Figures 1 and 2). Once the properties of one of these curves is known, behavior of others in the same group can be predicted with small error. The goal in this paper is to utilize the property(b) above together with (a) to achieve greater compression by exploiting the regional

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

(group) dependencies as well as single curve dependencies.

III. PROCESSING OF DATA

3.1 Use of computer graphics

An acoustic data tablet attached to the graphics facility was used to input the curves from weathermaps to appropriate memory . The graphics facility was also used to manipulate the curves interactively, as will be explained in 3.2. Stored data was further processed by APL because vector and matrix operations can be programmed very quickly, resulting in easy experimentation and debugging.

3.2 Linear Transformations

The Graphics facility was used to explore the relations between the curves within a group. (A group is roughly defined in Section 2.2.) The similarity of these curves suggests that a simple combination of linear transformations such as relocation, scaling and rotation can transform one curve in the group to another. In other words, with the use of a few parameters two curves can be made to look quite similar. Figures 3 - 6 show two examples of groups of closed curves and their re-shaped versions. (Uniformly scaled, rotated and shifted.) The purpose of the re-shaping operation is two-fold:

1. The similarities and differences between these curves can be observed more easily.

2. Re-shaping can be a part of the coding process. That is, if only a small number of transformation parameters is enough to transform one curve to another, it can be worthwhile to do the encoding in the following way: First, encode the suitable transformation parameters and one of the curves. Then encode only the differences between one curve and transformation of another. This is in fact what is done in this paper and we will return to this in Section 4.1.

To be able to realize the second purpose above, it is necessary to define a measure of distance between contours and to set up an algorithm for determining suitable parameters for transformations.

3.3 Transformation Algorithms

Define the distance between two curves, referred to as "1" and "2" (Figure 7) as follows: Choose an appropriate number, say n, of points (spaced according to the variations in the curve; this matter will be discussed at the end of the section) on curve 1. For each selected point, find the closest point on Curve 2. The sum of the squares of distances between corresponding points is an appropriate distance measure. Then, the problem of interest in this section is to find a linear transformation (a set of parameters) for curve 2 such that the distance between the transformed curve (call it 2^*) and curve 1 is minimum.

Let (x_i^1, y_i^1) denote the i'th point on curve 1 and (x_j^2, y_j^2) denote the j'th point on curve 2. The transformation of interest is:

$$\begin{bmatrix} {x_i^2}^* \\ {y_i^2}^* \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \end{bmatrix} \begin{bmatrix} x_i^2 \\ y_i^2 \\ 1 \end{bmatrix} \quad \text{or} \quad {P_i^2}^* = M\, P_i^2 \qquad (1)$$

where ${P_i^2}^* = [{x_i^2}^* \; {y_i^2}^*]$, ith point on the transformed curve 2^*,

$$P_i^2 = [x_i^2 \;\; y_i^2 \;\; 1]^T ,$$

and M is the 2 x 3 matrix above.

Find parameter matrix M to minimize

$$D = \left| \sum_{i=1}^{n} ({x_i^2}^* - x_i^1)^2 + \sum_{i=1}^{n} ({y_i^2}^* - y_i^1)^2 \right|$$

Generalize equation 1 to n points

$$\begin{bmatrix} x_1^{2^*} \\ y_1^{2^*} \\ x_2^{2^*} \\ y_2^{2^*} \\ \cdot \\ \cdot \\ \cdot \\ x_n^{2^*} \\ y_n^{2^*} \end{bmatrix} = \begin{bmatrix} x_1^2 & y_1^2 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_1^2 & y_1^2 & 0 & 1 \\ x_2^2 & y_2^2 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_2^2 & y_2^2 & 0 & 1 \\ \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & x_n^2 & y_n^2 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_4 \\ m_5 \\ m_3 \\ m_6 \end{bmatrix} \quad \text{or} \quad P^{2^*} = A \times MV \qquad (2)$$

where MV is the parameter vector $[m_1\ m_2\ m_4\ m_5\ m_3\ m_6]^T$.

The least square solution to MV is given by

$$\widehat{MV} = (A^T A)^{-1} A^T P^1 \qquad (3)$$

where $P^1 = [x_1^1\ y_1^1\ x_2^1\ y_2^1\ \ \ldots\ \ x_n^1\ y_n^1]^T$

As it can be observed, the parameter vector MV provides for operations of scaling, rotation and translation. The four parameters m_1, m_2, m_4, and m_5 provide for a combination of scaling and rotation operations. m_3 and m_6 are translation parameters for the x and y coordinates respectively following scaling and rotation.

The choice of points selected for distance measure will effect MV and p^{2^*}. If points are chosen closer together at the regions where curves are rapidly changing (and possibly not very similar) and if they are chosen far apart at the slow changing regions where curves look quite alike, it is possible to get a better fit. The same effect can be achieved by weighting the errors more heavily in some regions than in others, causing parameters to define a transformation in favor of heavily weighted regions.

3.4 Spline Representation

In this work, instead of representing curves by conic sections, parametric cubic spline approximations are used because they are found to be convenient to work with since spline fitting algorithms have nice properties such as automatically guaranteed slope continuity at the knot points, and smoothness of spline functions.

For periodic splines (closed curves) the only information to be coded is the coordinates of the knot points. For non-periodic splines, additional 2 end-point slope information is required. (See appendix). If the same number of knot points are assigned to two similar curves, it is possible to express one curve in terms of the other. This is discussed in the next section.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

IV. CODING PHILOSOPHY

4.1 Discussion of coding method

Consider one typical group of closed curves, Figure 4. In this section we discuss an adaptive predictive coding method for the knot points on one of the curves, given all knot points on the adjacent one, and we roughly estimate the costs (code length in bits) involved in the coding process. The problem of predicting knot points on a single curve in terms of other knot points on the same curve can be handled by known methods (such as curve fitting) and will not be discussed here. It will be enough to say that a reduction factor of 1.4 is possible over coding each knot point individually for the weathermap data in the examples shown.

Figure 8 shows the result of one global transformation applied to 3 inner most curves of figure 4. The transformation minimizes the sum of the errors in approximating each curve by a transformation of an adjacent curve. 2^*, 3^*, and 4^* are the transformed versions of 2, 3, and 4. The problem of interest is to encode the differences between 1 and 2^*, 2 and 3^* etc. in a most efficient way. The reasons for using one transformation above instead of using optimal transformations for 1 and 2 and for other pairs separately is a compromise between better overall fits and the cost of encoding transformation parameters. We will return to this later in the section where overall costs and reduction factors are discussed.

Consider curves 1 and 2^*. The set of points to be encoded on 1 are shown by '0''s. They are determined by spline algorithms. The knot points for 2^* are determined by a procedure that will be discussed later in the section. The points to be encoded on 2^* are shown by 'X''s in the figure. (Curve 2 is recovered from 2^* at the receiver through use of inverse transformation applied to 'X''s, then by the use of spline algorithms on these transformed points.) As expected, although most parts of the two curves are very close, portions of them are significantly different due to local variations which cannot be avoided by a single linear transformation. So, variability in local behavior suggests a coding method adaptive to local characteristics. That can be obtained by partitioning the curves into sets of regions and defining different predictors for different sets.

In figure 8, the field is divided into two regions S1 and S2. S1 is the region where curves 1 and 2^* are quite close, and S2 is the region where the two curves are very dissimilar. In general, S1 and S2 will be unions of seperated sets, as is the case in figure 8, where A is one such subset of S1.

When referring to a predictor for any point i on curve 2^*, we will in theory mean the conditional expectation of point i = $(x_i^{2^*}, y_i^{2^*})$ whose conditional probability density function can be expressed by:

$$P(x_i^{2^*}, y_i^{2^*} | x_1^1, y_1^1, \dots x_n^1, y_n^1, x_1^{2^*}, y_1^{2^*}, \dots, x_{i-1}^{2^*}, y_{i-1}^{2^*})$$

where $S_n^1 = \{x_k^1, y_k^1, k = 1, \dots n\}$ is the set of all knots on curve 1

and $S_{i-1}^{2^*} = \{x_k^{2^*}, y_k^{2^*}, k = 1, \dots i - 1\}$ is the set of knots on 2^* that are already known.

Then, the predictor $m_i^{2^*}$ is defined as

$$m_i^{2^*} = E(x_i^{2^*}, y_i^{2^*} | S^1, S_{i-1}^{2^*})$$

For S1 above, we observe in the examples that the effect of (x_i^1, y_i^1) on $m_i^{2^*}$ is much greater than is the effect of the other points on $m_1^{2^*}$. It is reasonable to assume that

$m_i^{2^*} = (x_i^1, y_i^1)$, that is, point $(x_i^{2^*}, y_i^{2^*})$ is most likely to be equal to the corresponding (x_i^1, y_i^1) on 1.

Furthermore, for S1, ($x_i^{2^*}$, $y_i^{2^*}$) is taken to be the intersection of curve 2^* with the perpendicular line to the spline approximation to curve 1 at point (x_i^1,y_i^1). Then only 1 error variable, the distance from (x_i^1,y_i^1) to ($x_i^{2^*}$,$y_i^{2^*}$), need be encoded and transmitted.

For S2 it is more difficult to decide which particular subsets of S^1 and $S_{i-1}^{2^*}$ have the greatest interdependence with $m_i^{2^*}$. We will assume that $m_i^{2^*} = f(((x_{i-1}^1 - x_{i-1}^{2^*})^2 + (y_{i-1}^1 - y_{i-1}^{2^*})^2)^{1/2}, x_i^1, y_i^1, x_{i-1}^{2^*}, y_{i-1}^{2^*}, \ldots x_{i-k}^{2^*}, y_{i-k}^{2^*})$ that is, $m_i^{2^*}$ is a function only of (x_i^1, y_i^1), the distance between (x_{i-1}^1, y_{i-1}^1) and ($x_{i-1}^{2^*}$, $y_{i-1}^{2^*}$), and the last k points that are already known on 2^*.

For S2, choosing knots on 2^* by using perpendiculars from 1 does not always result in suitable points because of fast changes in either one of the curves. Our approach is to choose knots on 2^* such that chordlengths between corresponding pairs of knots on each of the two curves are of constant ratio, which is equal to the ratio of the total lengths of curve segments over this particular region. (($\bar{P}_{0,3}/\bar{P}_{0^*,3^*}$) = ($\bar{P}_{0,1}/\bar{P}_{0^*,1^*}$) = r in figure 9.) $\bar{P}_{i,j}$ is the length of the segment between points i and j. For successive points, $\bar{P}_{i,j}$ can be approximated by the distance between these points.) In that case, the receiver has to know the ratio r, prior to receiving information about knots on 2^* for that region.

Then, the predictor $m_i^{2^*}$ will be somewhere on the circle CR with radius r and center located at ($x_{i-1}^{2^*}$, $y_{i-1}^{2^*}$). It might be determined by one of the following algorithms: (1) Fit a cubic curve C to the points $\{x_{i-1}^{2^*}, y_{i-1}^{2^*}, \ldots, x_{i-k}^{2^*}, y_{i-k}^{2^*}\}$. Find the intersection of C with CR, (2) Find the intersection of circle CR with circle with center at (x_i^1, y_i^1) and radius $((x_{i-1}^1 - x_{i-1}^{2^*})^2 + (y_{i-1}^1 - y_{i-1}^{2^*})^2)^{1/2}$. In any case, assume that the receiver wants to locate point P^{2^*} in figure 9, while points P^1, P^{1^*}, and P^2 are known. Assume that $m_i^{2^*}$ is found to be the point $P^{\hat{2}}$. The transmitter has to encode only the difference between the lengths of chords P^2 $P^{\hat{2}}$ and $P^2 P^{2^*}$. Since $|P^2$ $P^{\hat{2}}|$ is known by the receiver, $|P^2$ $P^{2^*}|$ is calculated. Since $|P^{1^*}$ $P^{2^*}|$ and $|P^2P^{2^*}|$ are known also, the angle $P^{1^*}P^2P^{2^*}$ is established and hence P^{2^*} is determined.

In figure 8, eight of the $n = 15$ points fall in S1 and seven fall in S2. We assume that each picture is quantized into 128 x 128 binary matrix form. A maximum of three bits per point in S1 and 5 bits per point in S2 are required for encoding each d_i. Another cost involved is the identification of each point belonging to one of the sets. That can be efficiently performed by first identifying a region belonging to one of the sets, which requires only one bit in our example, then encoding the number of knots belonging to the region,

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

requiring roughly four bits per region. Obviously, the above procedure should be performed before encoding the points for each region. The cost of encoding curve 2^* is ∿73 bits, including the cost of encoding r.

The number of bits required to encode the differences for 2 and 3^* is found by first defining the spline knots on 2 (which is reconstructed from 2^* at the receiver) by an algorithm known by both receiver and transmitter. Then the same procedure that is used for 1 and 2^* is used for encoding.Total cost of encoding 3^* and 4^* is ∿107 bits.

There are 6 transformation parameters to be encoded, 2 of them representing translation. The translation parameters require 8 bits each (including sign). The other 4 are related to scaling and rotation, requiring approximately 8 and 7 bits respectively for error-free reproduction at the receiver. Total cost for transformation parameters then is 46 bits.

Would it be beneficial to apply a different transformation for each pair of curves and then to encode the differences between sets of transformation parameters? Figure 8 shows the result of a transformation applied to curve 4 to make it look like curve 3 (4^{**}). Encoding the difference between transformation parameters is estimated to require about 16 bits, but newfit 4^* gives rise to a saving of ∿ 8 bits. Thus, we conclude that it does not appear to be worthwhile to use different transformations for adjacent curves, but rather one optimal transformation for the whole group.

A rough indication can be given of the saving in bits provided by this coding over encoding the knot points without using compression and with using single curve compression. Hence, we compare the cost of encoding for 4 curves in figure 8 using our method with the other two mentioned above. a) The cost of encoding 4 curves by encoding the knot points without using any compression is calculated by assuming that encoding each knot requires 14 bits (2^7x2^7 matrix). Spline approximation requires a total of ∿44 points, resulting in a total cost of ∿616 bits. b) Encoding the knot points using dependence along individual curves require 10 bits for each of 44 points, requiring a sum of 440 bits. c) Total cost for our method is the sum of: 1. Coding curve 1 by itself, 2. Cost of transformation parameters, 3. Cost of encoding curves 2^*, 3^*, and 4^*. Total cost for 'c' is ∿380 bits.

We conclude that a reduction factor of ∿1.6 is obtained for figure 8 over encoding knot points individually. It must be mentioned that in the example above, 3 innermost curves are transformed to adjacent outermost ones. That requires the number of knot points to be largerthan the case where 3 outermost curves would be transformed to adjacent inner ones. (Total wordlength would be shorter for the second case.) It would be possible to get compression ratio of ∿1.8 for the second case.

4.2 Results and Conclusions

In this section, we will summarize the suggested coding method step by step and discuss the results. A code might be constructed as follows"

(1) Divide the whole region so that groups of similar curves are formed.

(2) For groups that contain only one curve, encode each curve separately, by estimating each knot point using knots that are already known on the same curve. Here, each knot will require about 10 bits.

(3) For each of the other groups:

(a) First encode one of the curves as in (2) above

(b) Encode the linear transformation parameters, requiring about 46 bits total for one transformation

(c) Encode the other curves in the group with respect to the one encoded in (a) and the transformation in (b). On the average, each knot point can be specified by ∿5 bits including the specification of the region it belongs.

The code constructed above is able to compress weathermap or similar data with a ratio between 1.5 and 2 compared to the coding of individual knot points.

The discussed compression ratio can be improved if Huffman coding is used for encoding the symbols. Also, the relations between groups in one picture can be used for compression, such as exploiting the similarities between transformation parameters.

REFERENCES

[1] Huang, T. S., "Run Length Encoding and it's Extensions", in Picture Bandwidth Compression, Gordon and Breach, New York, 1972, pp. 231-264.

[2] Wintz, P. A., "Transform Picture Coding" IEEE Proc., vol. 60, No. 7, July 1972, pp. 809-820.

[3] Connor, D. J., Pease, R. F., Scholes, W. G., "Television Coding Using Two Dimensional Spatial Prediction", Bell System Technical Journal, vol. 50 No. 3, March 1971, pp. 1049-1061.

[4] Cooper, D. B., "Feature Selection and Super Data Compression for Pictures Occuring in Remote Conference abd Classroom Communications", The Record of Second International Conference on Pattern Recognition, Lyngby, Denmark, August, 1974, pp. 111-115.

[5] Graham, D. N., "Image Transmission by Two Dimensional Contour Coding", Proc. IEEE, vol. 55, March 1967, pp. 336-346.

[6] Pavlidis, T., "Techniques for Optimal Compaction of Pictures and Maps", Computer Graphics and Image Processing, No. 3 (1974) pp. 216-224.

[7] Preuß, D., "Facsimile Source Encoding with Constant Output Bit Rate", NTZ Communications Journal, May 1974, no. 27, pp. 198-202.

[8] Musmann, H. G., Preuß, D., "A Redundancy Reducing Facsimile Coding Scheme", NTZ Communications Journal, No. 26 (1973) pp. 91-94.

[9] Villalobos, L., "Compression of Contour Map Data with Conographic Curves", Memo by Hughes Aircraft Company.

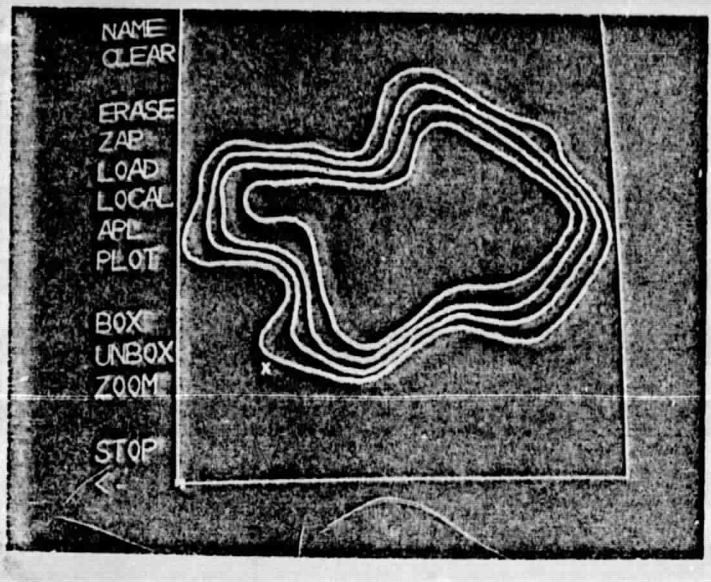


Figure 1

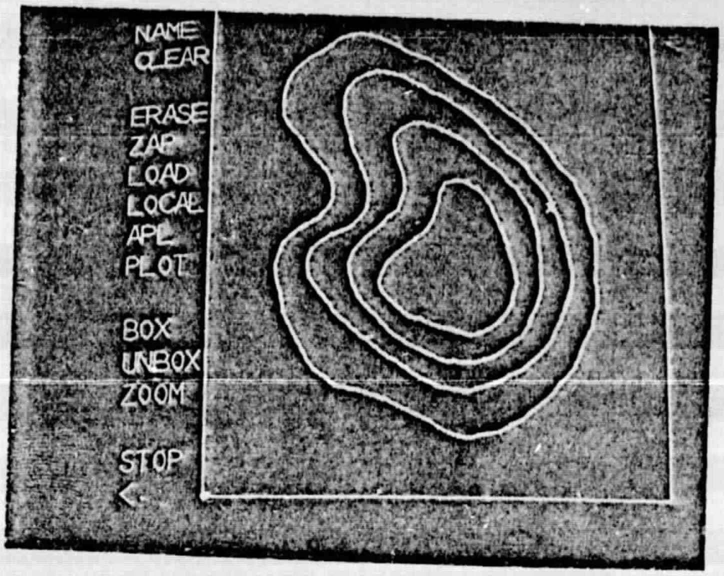
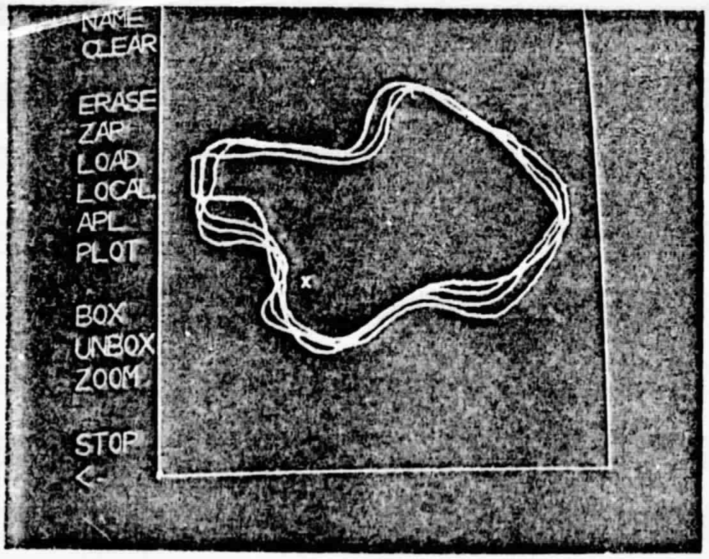
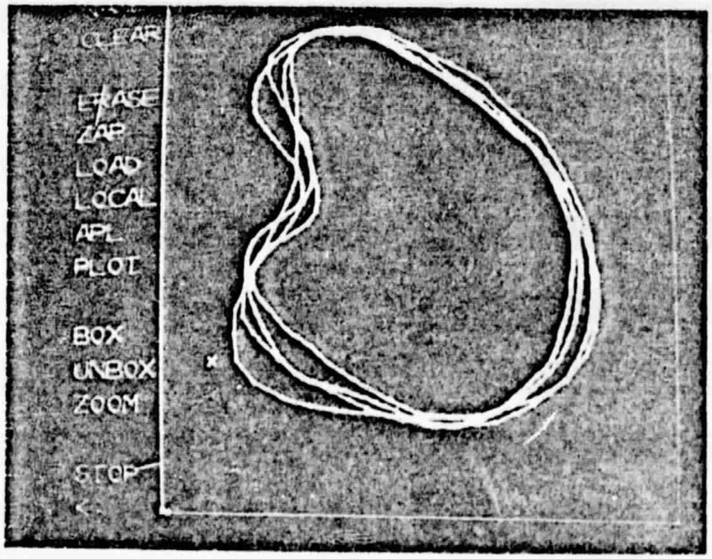


Figure 2



Curves of Fig 1 uniformly scaled, rotated and superimposed

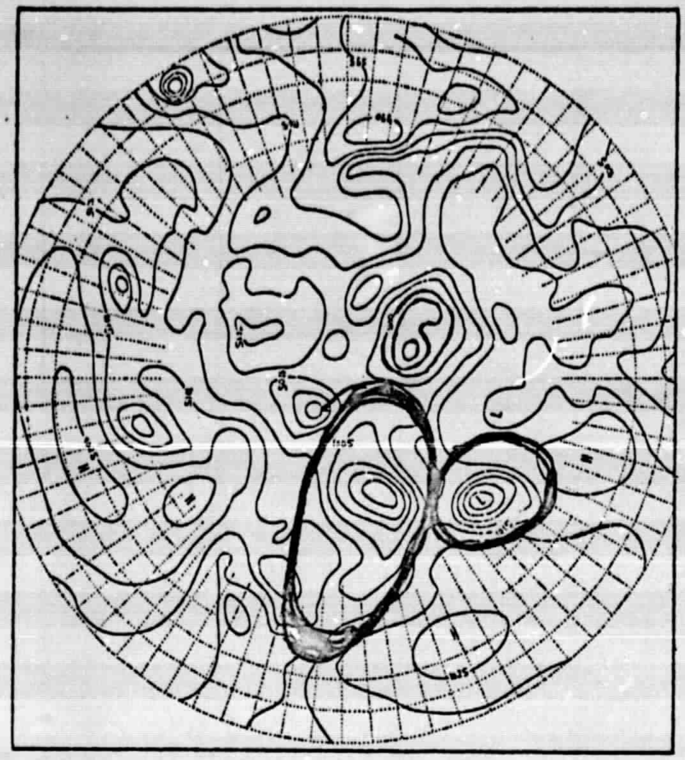
Fig 3



- Curves of Fig 2 uniformly scaled, rotated, and superimposed.

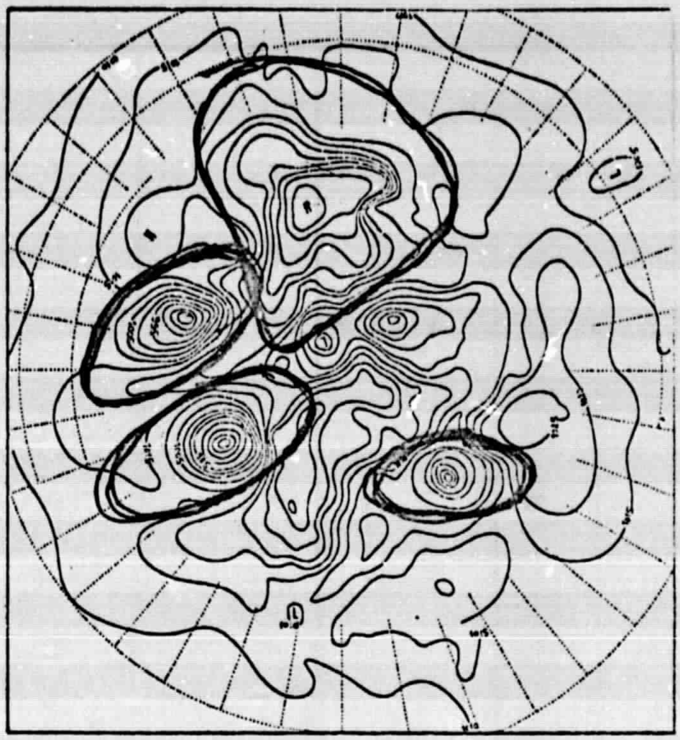
Fig 4.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR



International Geophysical Year world weather maps
Part I - Northern Hemisphere - 15 July, 1958.
1200 GMT. Sea Level

Figure 5



Synoptic weather map. Northern Hemisphere.
January 15, 1959
1230 GMT. Sea Level

Figure 6

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

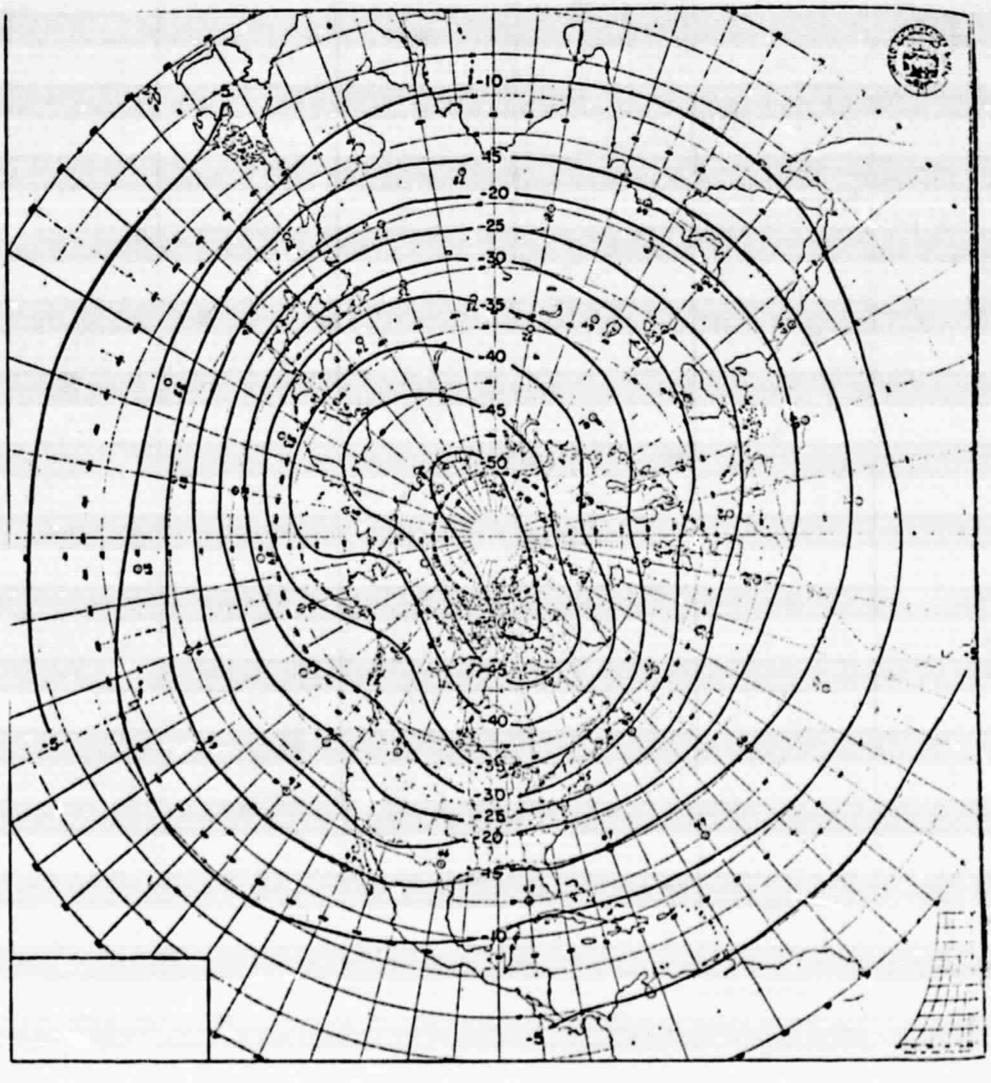


Figure 7

Fig. 9. Mean temperature at the 20,000-ft level, January (°C).

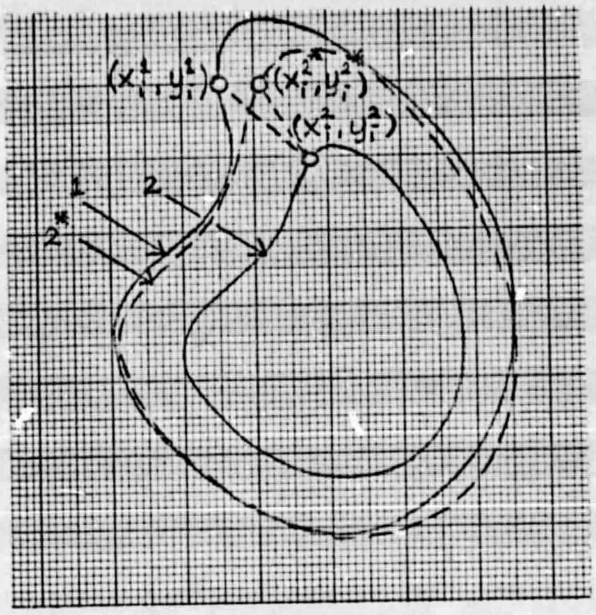


Fig.7. Location of points (x_i^1, y_i^1), (x_i^2, y_i^2) and (x_i^{2*}, y_i^{2*}) on curves 1, 2 and 2^*.

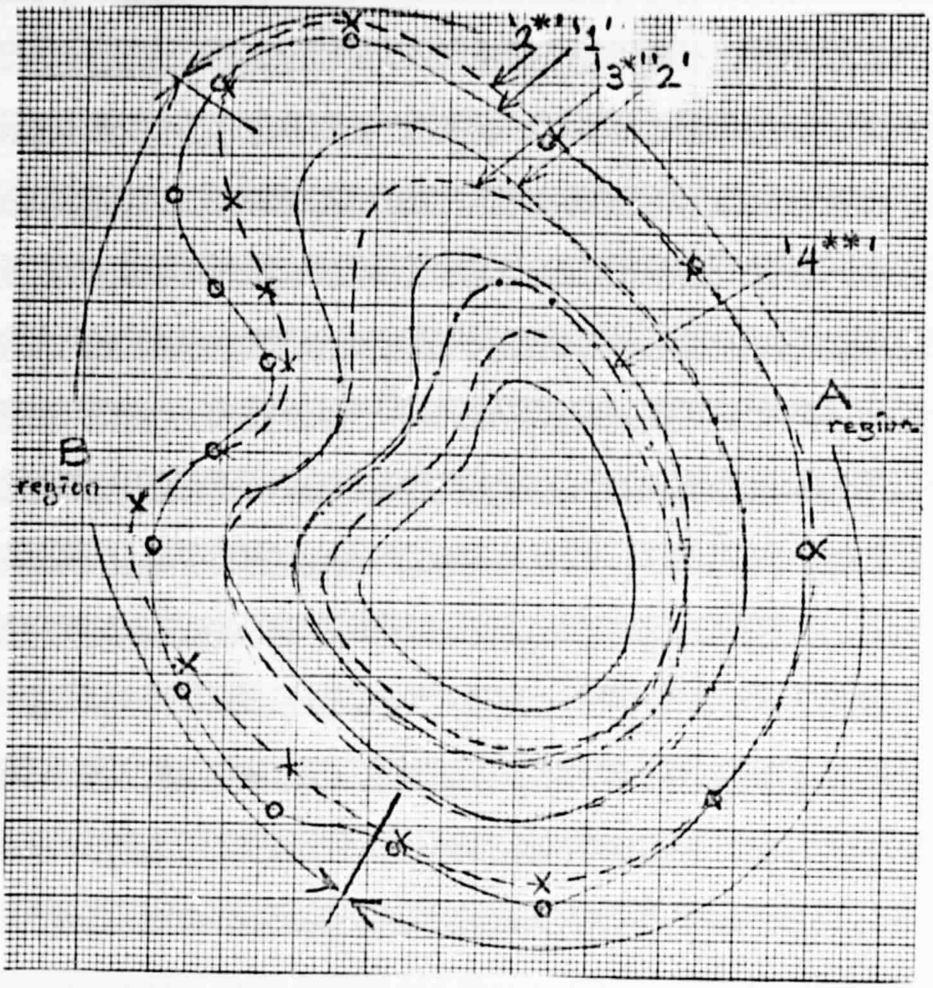
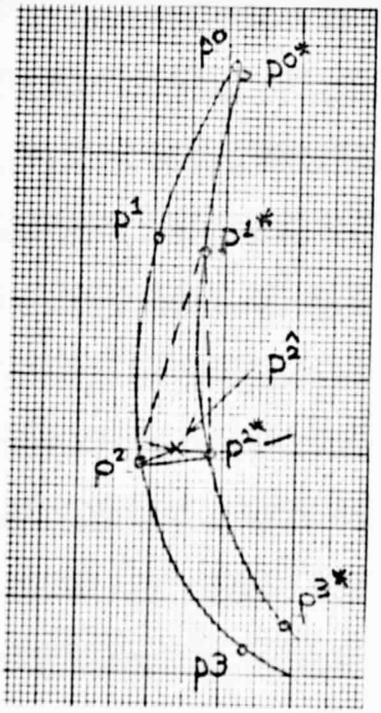


Fig. 8. A group of 4 curves and their transformations.



REPRODUCIBILITY OF THE
PAGE IS POOR

Fig 9. The determination of p^{2*} when p^1, p^{1*} and p^2 are known.