[redacted]

31 August 1964
WT:bb:jg-405
997-112

To: [redacted]

From:

Subject: Calibration of Neutral Density Filters

CC: [redacted]

Introduction

This memo presents the results of the calibration of glass neutral density filters manufactured by [redacted] [redacted] The densities of the filters tested, as determined by [redacted] were .21, .69, 1.06, 1.51, and 1.83.

Experimental Procedure

The glass filters were tested using the [redacted] Spectrometer, employing a 931A phototube. The illuminating source was a tungsten filament. Each of the filters tested was placed in front of the entrance slit to the spectrometer. Care had to be taken to prevent the phototube from saturating. The spectrometer then automatically scanned from approximately 3790 Å to 7860 Å and recorded the intensity as a function of wavelength. A tungsten response curve with no filter present was also recorded. It was not possible to test densities greater than 1.83 due to the large amount of noise present in the output when high gain was used on the spectrometer recorder. It is believed that most of the noise was due to erratic line voltage fluctuations caused by the machines in the shops. The tests were repeated several times and the results averaged to reduce the errors caused by noise.

Results

Three independent functions of the wavelength, λ, are present in this test; (1) the intensity of the tungsten source, $I(\lambda)$, (2) the transmission of the filters, $T(\lambda)$, and (3) the response of the phototube, $R(\lambda)$. The recorder output, $W(\lambda)$, can be written as:

$$W(\lambda) = I(\lambda)\, T(\lambda)\, R(\lambda)$$

If no filter is present, the tungsten response curve will be:

$$W_o(\lambda) = I(\lambda)\, R(\lambda)$$

Thus,

$$T(\lambda) = \frac{W(\lambda)}{W_o(\lambda)}$$

can be determined by dividing point for point the filter response curve by the tungsten response curve. The results of $T(\lambda)$ vs. λ for each filter are plotted in Figures 1 and 2.

It is seen that $T(\lambda)$ is not a constant. It is therefore necessary to obtain an "average" density as indicating the "neutral" density. The simplest "average" to define is $\bar{D}_1$, which is found by calculating the mean transmission, $\bar{T}_1$, and converting this value to density

$$\bar{T}_1 = \frac{\int_\lambda T(\lambda)\, d\lambda}{\int_\lambda d\lambda}$$

and $\bar{D}_1 = \log_{10}(\bar{T}_1)^{-1}$.

Simple numerical approximations to these integrals can be defined as:

$$\bar{T}_1 = \frac{\sum_{m=1}^{M} T(\lambda_m)}{M}$$

where the average is taken over $M = 12$ transmission readings for λ running from 3790 Å to 7860 Å. The results of these calculations are tabulated in Table 1 and the values of $\bar{T}_1$, are indicated on Figures 1 and 2.

TABLE 1

Labeled Density D	Labeled Transmission T	Mean Transmission $\bar{T}_1$	Density $\bar{D}_1$
.21	.62	.58	.24
.69	.20	.18	.74
1.06	.087	.074	1.13
1.51	.031	.031	1.51
1.83	.015	.015	1.83

The above averaging process is not, in general, characteristic of the way that a densitometer averages the transmission of the light in making a density measurement. A densitometer will measure the transmittance of a filter as a weighted average, where the weighting factor for each wavelength is the system response.

Since both the Ansco Model 4 Microdensitometer and the MacBeth Densitometer use tungsten sources and 931A phototubes, it is reasonable to take the response to be $W_o(\lambda)$. The data recorded by the spectrometer for the filter transmissions, $W(\lambda)$, is the product of the filter response and the system response, $T(\lambda)\, W_o(\lambda) = W(\lambda)$.

STAT

31 August 1964
WT:bb:jg

Thus, $\bar{\bar{T}}_z$ can be defined as a weighted average of $T(\lambda)$ by:

$$\bar{\bar{T}}_z = \frac{\int_\lambda W(\lambda)\, d\lambda}{\int_\lambda W_o(\lambda)\, d\lambda}$$

Simple numerical approximations of these integrals can be defined as:

$$\bar{\bar{T}}_z = \frac{\sum_{m=1}^{M} W(\lambda_m)}{\sum_{m=1}^{M} W_o(\lambda_m)}$$

where the summations are over the $M=12$ recorded values of the functions. Increasing M to 24 did not significantly change the results of calculations.

The results for the calculation of $\bar{\bar{T}}_z$ for λ running from 3790 Å to 7860 Å are shown in Table 2 and on Figures 1 and 2. $\bar{D}_z$ is defined as $\log_{10}(\bar{\bar{T}}_z)^{-1}$.

TABLE 2

D	T	$\bar{\bar{T}}_z$	$\bar{D}_z$
.21	.62	.65	.19
.69	.20	.20	.69
1.06	.087	.091	1.04
1.51	.031	.034	1.47
1.83	.015	.017	1.77

Conclusions

The results for $\bar{\bar{T}}_z$ are too high, thus making the $\bar{D}_z$ values less than the labeled filter values. In Figures 1 and 2, all experimental curves show a peak near λ = 5540 Å, the same wavelength at which the $W_o(\lambda)$ response curve peaks. It is not known whether or not these peaks are characteristic of the filters or of the apparatus. However, nothing could be done to reduce them. The sample curves of transmittance as a function of wavelength provided by Schott & Gen. do not show these peaks. The sample curves also show the transmittance remaining approximately constant for λ up to 10000 Å. The experimental results in Figures 1 and 2 show a definite falling off of transmittance above 6000 Å.

STAT [redacted]

31 August 1964
WT:bb:jg

Recommendations

In the event that future tests require measurements of the type described above, it is recommended that they be conducted when none of the shops are operating. It might also be worthwhile to check the voltage output versus light intensity relation for [redacted]. STAT

[redacted] STAT

Approved For Release 2005/05/02 : CIA-RDP78B04770A002300030034-8

Fig. (1) TRANSMISSION vs. WAVELENGTH for D = .21, .69, 1.06

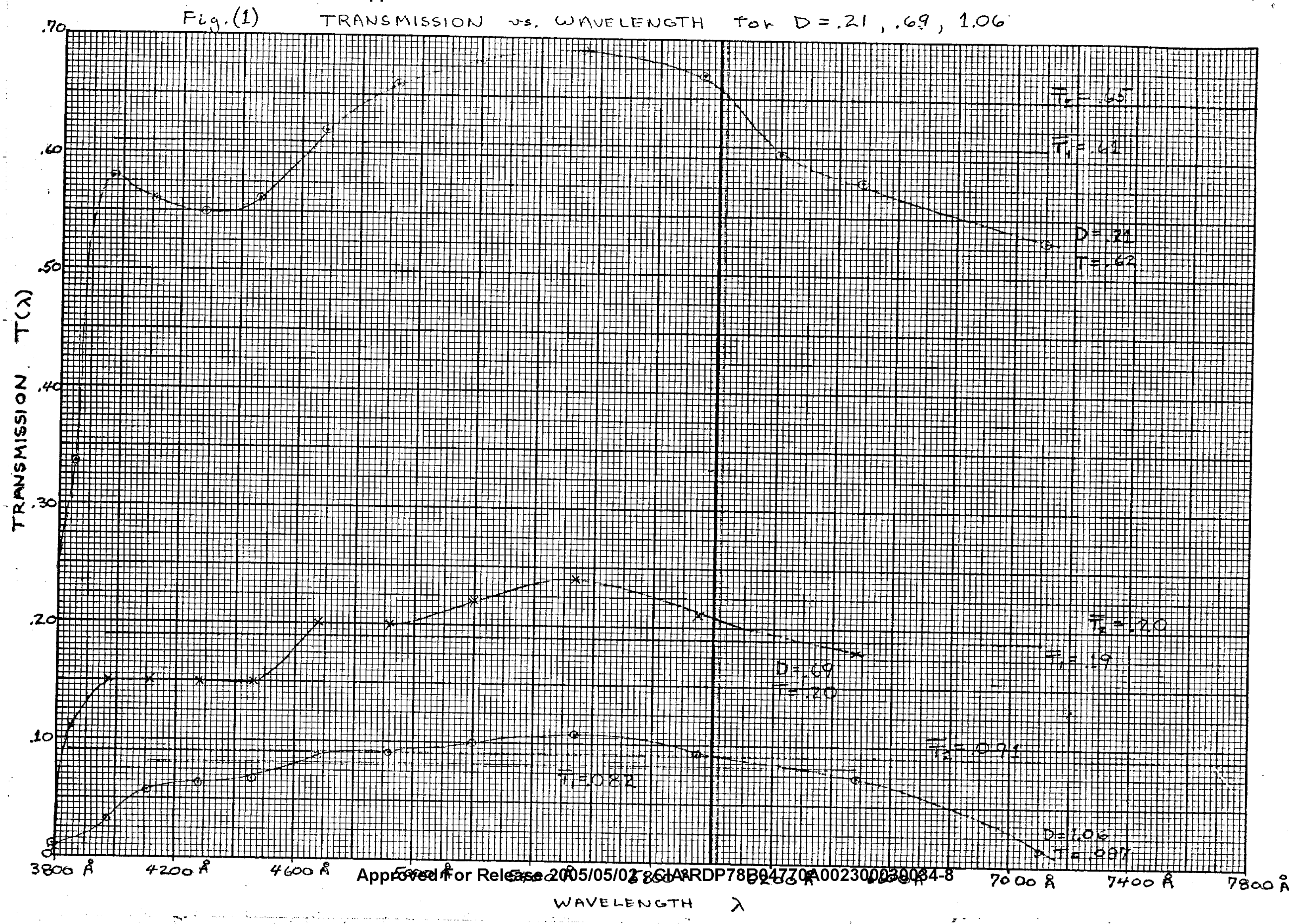
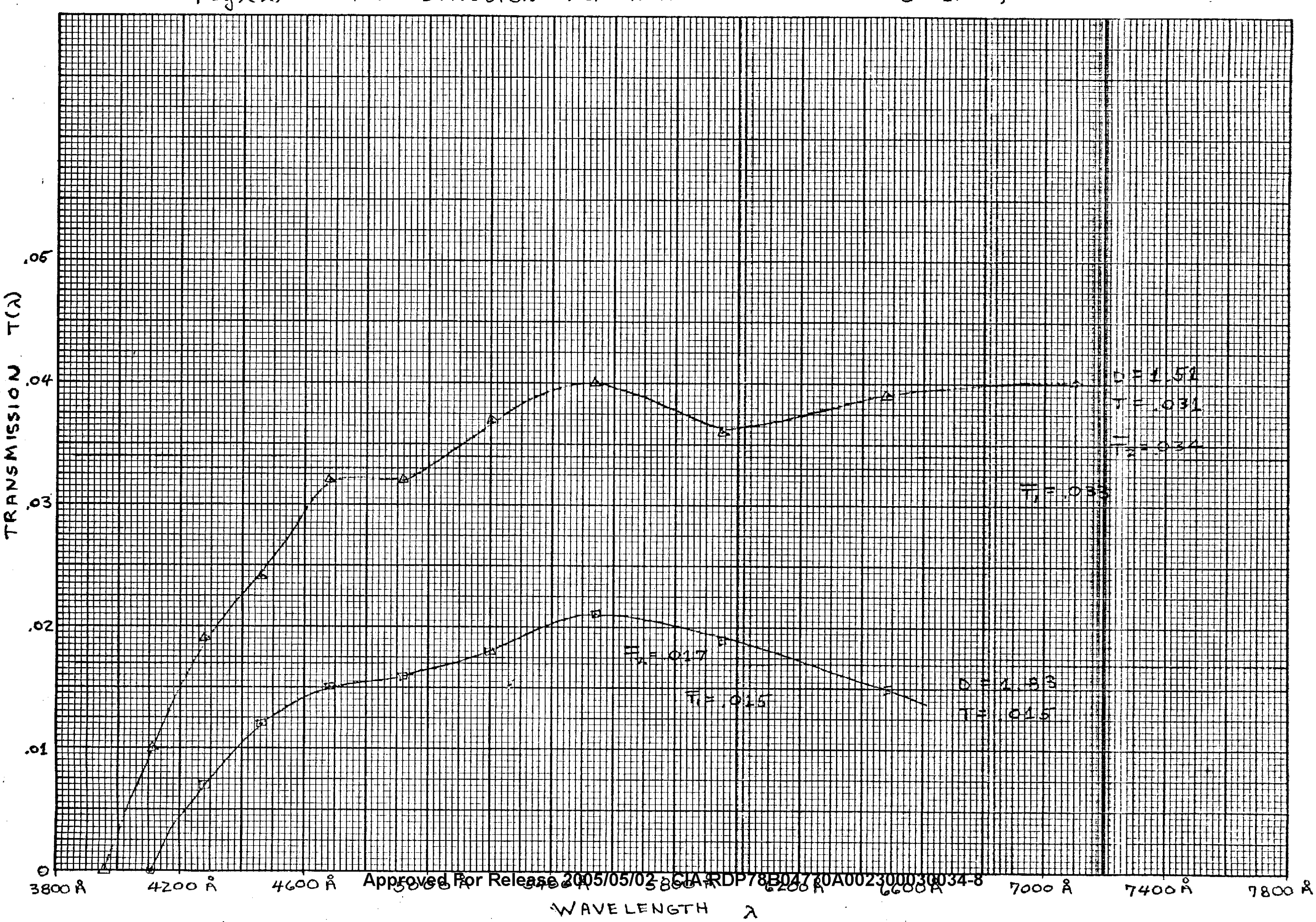


Fig. (2) TRANSMISSION vs. WAVELENGTH for D = 1.51, 1.83

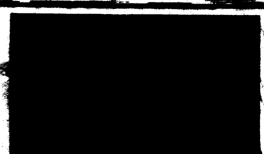


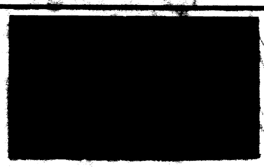
Approved For Release 2005/05/02 : CIA-RDP78B04770A002300030034-8

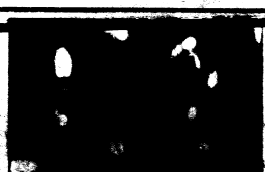
[redacted]

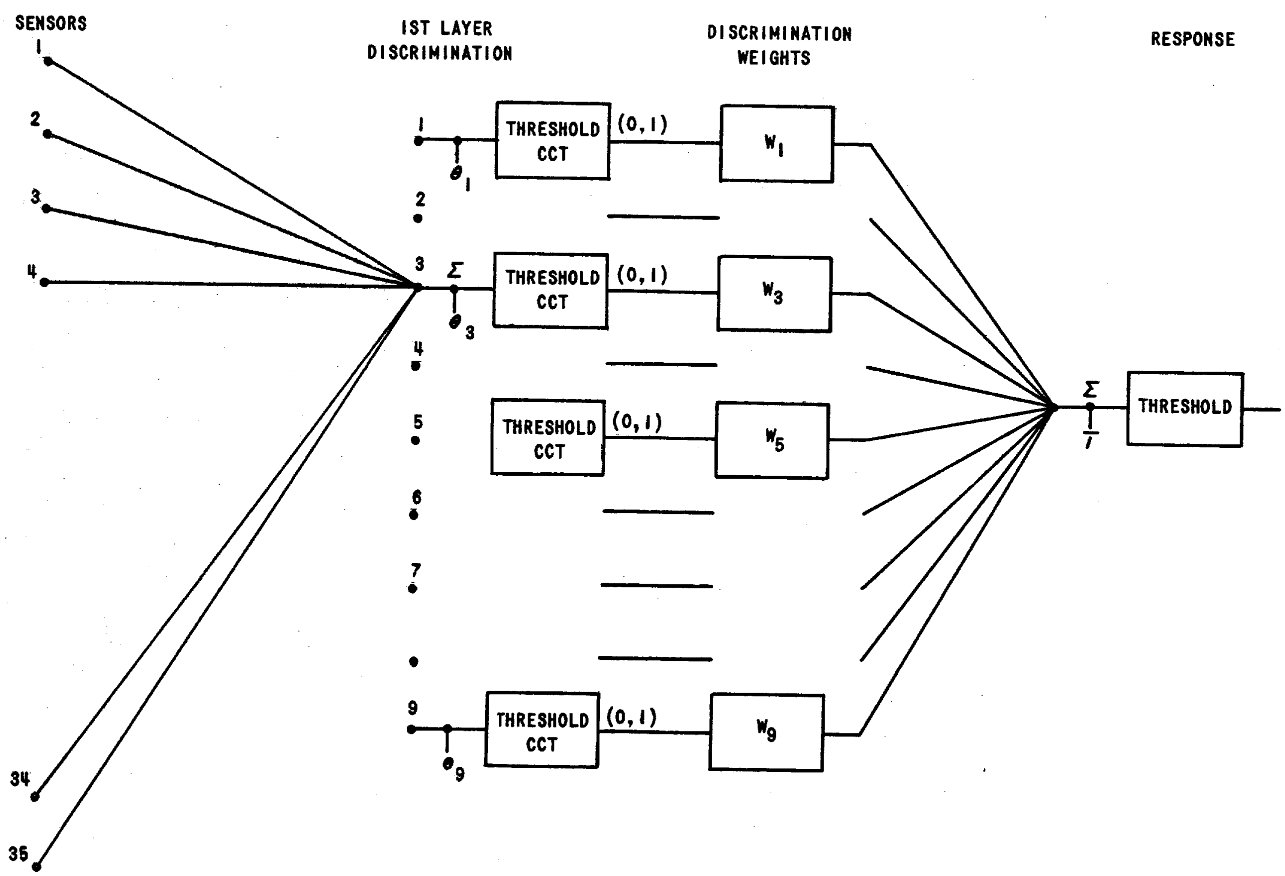
[redacted]

[redacted]









PATTERN RECOGNITION CONCEPTS

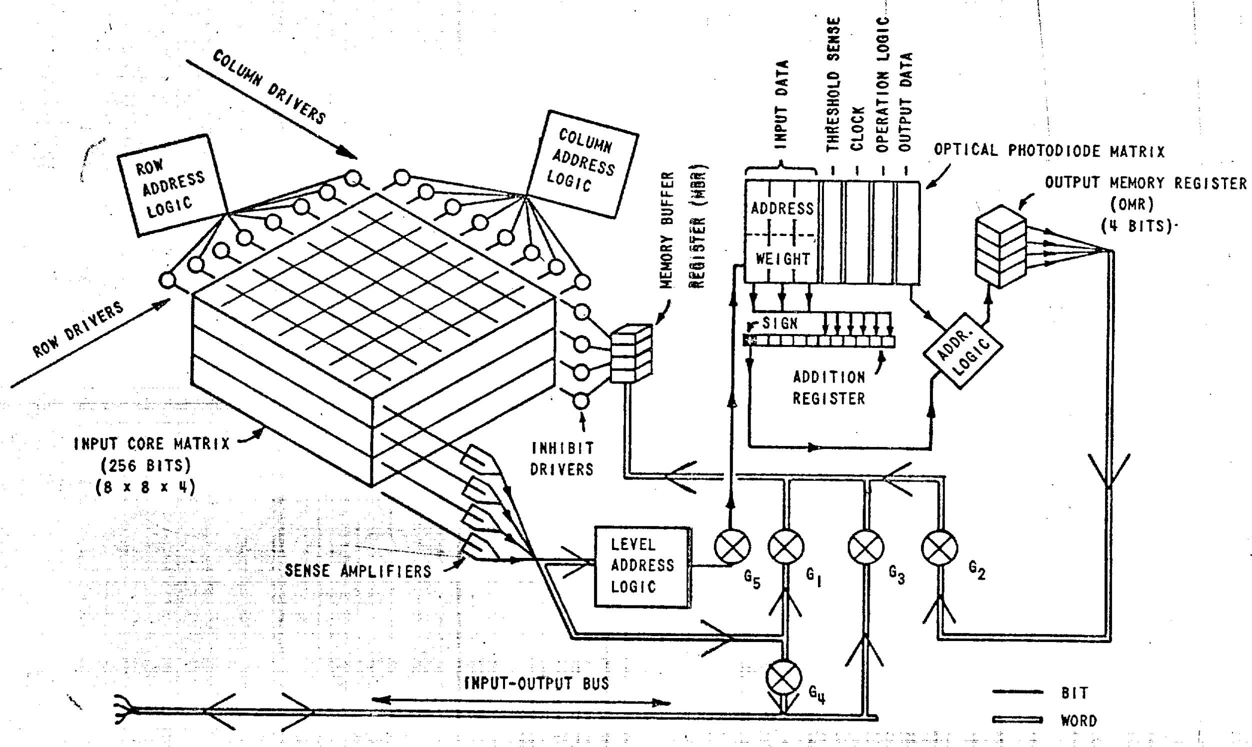


Fig. 21 SYSTEM DATA FLOW DIAGRAM FOR PROPOSED PERCEPTRON IMPLEMENTATION

FILE IN PCPTRN CONTRACT

19 SEP 63

TWO-DIMENSIONAL SPATIAL FILTERING AND COMPUTERS

ABSTRACT

Processing of two-dimensional signals has important applications, for example, in photographic image analysis, but when the weighting function of a two-dimensional linear filter extends over a large area, e.g., smoothing filters, digital realization via a two-dimensional convolution is prohibitively time consuming, and analog realization is extremely difficult. The principal purpose of this paper is to show how a broad and useful class of two-dimensional filtering operations can have notably shortened execution time in the digital case, and be put into a particularly convenient form for electrical filtering. The procedure includes a reduction of dimension from two to one; transformation of two-sided (weighting function extends into both past and future) operations into one-sided (physically realizable) operations; and finally, for the digital case, the transformation of a direct many-term convolution expression into a compact recursive form ideally suited for digital computation. An important class of smoothing filters, with weighting functions approximately Gaussian, is derived and used for illustration. The result is a several-order-of-magnitude reduction in time for digital two-dimensional filters, and some interesting results applicable to one-dimensional zero-phase-shift filters.

I. INTRODUCTION

Analog or digital processing of two-dimensional signals has many important applications, notably in photographic image analysis for military or commercial purposes. Much of this type of processing is special purpose, tailored to the physically significant details within an image. But there are certain basic operations (e.g. high and low-pass filtering) that are extensions of corresponding one-dimensional operations, and which have a similar range of usefulness.

Because in two-dimensional processing by digital means, or by means of electrical filters, storage and processing time requirements are much greater[1] than for one dimension, there is a genuine need for nontrivial methods to alleviate time and storage problems. This need is greatest when each computed point in the output image is affected by values from a relatively large area of the original or input image. Such filters, called area filters, are exemplified by many smoothing operations; they form the principal topic of this paper. In contrast to the area filter is the local filter, in which the value of an output image point depends only upon a small neighborhood of the corresponding input image point.

II. THE DIRECT CONVOLUTION APPROACH

The direct approach to realization of an arbitrary filter operating on a two-dimensional input is explicit convolution of the impulse response of the filter with the input. In the digital case this takes the form:

$$y_{m,n} = \sum_{i=-I}^{I} \sum_{j=-J}^{J} h_{ij}\, x_{m-i,\, n-j} \quad , \qquad (1)$$

where subscripts on y (output) and x (input) give the digitized coordinates of the image point, and h_{ij} is the discrete two-dimensional impulse response or weighting function.

In the case of a local filter the number of terms in this expression is small, and the method

[1] An exception is the optical filter, not considered in this paper.

Approved For Release 2005/05/02 : CIA-RDP78B04770A002300030034-8

is practical and often useful. For example, Kovasznay and Joseph (Ref. 1) describe an interesting analog equivalent to (1) used for outline enhancement. David (Ref. 2) reports a digital application of linear and nonlinear local operations for noise reduction.

In the case of an area filter, the large number of terms (121, for example, if $I = J = 5$) required for each output point makes this approach impractical.

III. THE SIMPLIFICATION STEPS

Equation (1) of the direct method is the starting point for a number of drastic simplifications. The goal is a set of one-dimensional filters, of simple recursive form for digital computation, and of physically realizable form for analog use. The simplification steps are:

(a) Restriction of the weighting function h_{ij} or $h(t_1, t_2)$ to a product form $f_i g_j$ or $f(t_1) g(t_2)$; this special form allows the two-dimensional problem to be decomposed into two one-dimensional problems.

(b) Transformation of the one-dimension filtering operation, which has an impulse response extending into both positive and negative time values, with two filtering operations of the "physically realizable" form, in which output depends upon the "past" (impulse response vanishes for negative argument).

(c) In the digital case, conversion of the one-dimensional filter into a recursive form so that only very few terms appear in the computation of an output value, even though the effective memory extends far into the past.

These items are discussed in the following four sections.

IV. REDUCTION OF DIMENSIONALITY

The first major simplification of (1) is the reduction in dimensionality from two to one, by restricting (with some loss of generality) the weighting function of two variables to be of product form

$$h_{ij} = f_i g_j \text{ (discrete) or}$$
$$h(t_1, t_2) = f(t_1)\, g(t_2) \quad \text{(continuous)}$$

Then the complete double summation of (1) may be replaced by two sets of calculations, each only having a single sum; omitting details, and again writing only the digital equations, we have:

$$\hat{y}_{m,r} = \sum_i f_i x_{m-i,r} \tag{2}$$

$$y_{m,n} = \sum_r g_r \hat{y}_{m,n-r} \tag{3}$$

Here, $\hat{y}_{m,r}$ is an intermediate result, computed from the x's, then regarded as input variable for final computation of the desired $y_{m,n}$ values. The first computation (2) is (for any fixed r) an ordinary one-dimensional filter, operating on one horizontal[2] line of the image; the parameter r identifies which line. Similarly, the second computation (3) is, for any fixed m, an ordinary one-dimensional filter, operating on a vertical line[2] of the y image; the parameter m identifies which vertical line.

To illustrate the general effect in terms of saving computing time, suppose that a 100 x 100 grid of picture intensity values is to be filtered, and assume that the weighting function of (1) extends 10 terms in each of the possible directions ($I = J = 10$). With edge effects ignored, the application of (1) directly would require 21 x 21 = 441 operations (an operation consisting of a multiplication and addition) for each processed point, for a total number of 4,410,000 operations. The corresponding double application of one-dimensional filters, according to (2) and (3), would require 21 + 21 = 42 operations per processed point, for a total number of 420,000 operations -- less than 1/10 as many.

---

[2] We arbitrarily identify horizontal lines with fixed second subscript, varying first subscript, and vertical lines with contrary conditions, in analogy with the continuous form $x(t_1, t_2)$, where the first variable normally gives the abscissa value and the second gives the ordinate value.

V. SIMPLIFICATION OF THE ONE-DIMENSIONAL FILTERS

The following approach towards simplifying the one-dimensional filters of (2) and (3) gives results definitely useful in originally one-dimensional filtering problems, as well as in the present context as intermediate aids for two-dimensional filtering.

It is convenient to drop the redundant double subscript, and begin with the generic form (open-loop, double-sided) form that is implied by (2), (3) and the preceding discussion:

$$y_n = \sum_{k=-K}^{K} h_k x_{n-k} \quad \text{(discrete)}$$

$$y(t) = \int_{-\infty}^{+\infty} h(\tau)\, x(t-\tau)\, d\tau \quad \text{(continuous)}, \qquad (4)$$

and it is convenient to regard a subscript as a time value (e.g., y_n is the value of y at the quantized time value n).

Equation (4) could be used in its existing form for digital computation. But for the case where the number of terms is large, there is a more efficient method most conveniently derived by orienting our terminology towards the continuous-time situation, so that we can simultaneously develop a method suited to electrical filtering and of a form that can be converted to an efficient recursive digital filter.

Write $h(\tau) = h_+(\tau) + h_-(\tau)$, where h_+ vanishes for negative time, and h_- vanishes for positive time. The h_+ part could be the impulse response of a realizable filter, assumed to have rational transfer function $F_1(s)$ with its poles in the left half plane. The h_- part may be achieved with a realizable filter by <u>reversing</u> the time variable, for example, by storing the input on magnetic tape, then playing the tape backwards. A reversed-time signal passed through a realizable filter with transfer function $F_2(s)$, gives a result that is formally equivalent (in the sense of a bilateral Laplace transform) to passing the original forward-time signal through a filter with transfer function $F_2(-s)$. Thus we may speak of filters with poles in the right half plane, with the understanding that they refer to a physically realizable filter driven by a time-reversed version of an input signal.

Thus, if the original picture is processed with filter having transfer function $F_1(s)$, then the original picture is filtered independently with filter having transfer function $F_2(-s)$ (actually accomplished by running the input signal backwards through a filter with transfer function $F_2(s)$), and the results are added, the resultant impulse response will be desired $h_+ + h_-$, and the total effective transfer function will be

$$G(s) = F_1(s) + F_2(-s)$$

The rational functions $F_1(s)$ and $F_2(-s)$ may be combined into a single multiplicative expression, of the form

$$G(s) = G_1(s)\, G_2(s)$$

where G_1 contains all of the poles in the left half plane and G_2 contains all of the poles in the right half plane. Thus, we have a cascade form: rather than filter the original picture twice and add results, as in the previous paragraph, one could filter the original picture with G_1 , then filter that result with G_2 . In the digital case, the latter has the advantage of eliminating the need for duplicate storage space, by means which, if not obvious, are simple. For analog filtering the cascade form is more convenient and avoids various practical synchronization difficulties.

VI. DIGITAL ONE-SIDED RECURSION FILTERS

An <u>open-loop</u> <u>one-sided</u> digital filter has the form

$$y_n = h_0 x_n + h_1 x_{n-1} + h_2 x_{n-2} + \cdots,$$

the expression possibly being infinite in extent. "Open-loop" refers to the fact that y is expressed only in terms of x's (inputs); "one-sided" refers to the fact that only present (time n) and past values of input are used to determine output. For example, the one-sided open-loop digital filter corresponding to an exponential impulse response (simple RC lag filter) is

$$y_n = x_n + \alpha x_{n-1} + \alpha^2 x_{n-2} + \alpha^3 x_{n-3} + \cdots. \qquad (5)$$

If, as in this example, the number of terms is infinite, then digital computation can be performed only by approximating with a finite number of terms. But if α in the expression above is fairly close to unity, (e.g., 0.99), it is possible that several hundreds of terms would be required for satisfactory approximation.

A one-sided feedback digital filter equivalent to (5) has the form

$$y_n = \alpha y_{n-1} + x_n \tag{6}$$

where α is the same one as in (5). This form is of course particularly suited to digital computation, since it contains only two terms. "Feedback filter" here refers to the fact that output depends explicitly on prior output, as well as the x input; this type of filter is also commonly called a recursive (or recursion) filter.

VII. RECURSION FORMULAS FROM TRANSFER FUNCTIONS

The general problem of converting a transfer function (such as our $F_1(s)$, if additive form is used, or $G_1(s)$ if cascade form is used) into a digital recursion relationship is easily solved by means of a method presented at the 1961 NEC (Ref. 3, 4), which is described here only in necessarily very sketchy form. In the expression for a transfer function $G(s)$, make the substitution $s \rightarrow \frac{2}{T}\frac{1-z}{1+z}$, clear of extraneous fractional forms and normalize numerator and denominator into the following form:

$$G \rightarrow \frac{a_0 + a_1 z + a_2 z^2 + \cdots + a_m z^m}{1 + b_1 z + b_2 z^2 + \cdots + b_r z^r} \tag{7}$$

(z can be interpreted as the delay operator, with Laplace transform e^{-sT}).

Then the recursion formula becomes

$$y_n = a_0 x_n + a_1 x_{n-1} + \cdots + a_m x_{n-m} - b_1 y_{n-1} - b_2 y_{n-2} - \cdots - b_r y_{n-r} . \tag{8}$$

The quantity T is a time scaling parameter; in this application, it relates the time variable of the Laplace transform to the separation between adjacent sample values.

To illustrate, we use a transfer function that will later be used to illustrate another aspect of the two-dimensional filtering problem;

$$G(s) = \frac{2\sqrt{3}}{s^2 + 3s + 2}$$

The impulse response has time constants of the order of one second in real time. Suppose we desire that over one of these time constants there would be 10 picture elements (in a rough sense, the memory covers ten picture elements, if memory is taken to be a nominal time constant). Then $T = 0.1$. Use of the substitution described above gives for the final recursion formula:

$$y_n = 0.00849(x_n + 2x_{n-1} + x_{n-2}) + 1.951y_{n-1} - 0.9106y_{n-2}$$

VIII. ILLUSTRATIVE EXAMPLE

In many two-dimensional filtering applications, it is desired to have an impulse response that is circularly symmetrical. The purpose of this section is to show how one class of such impulse responses can be approximately realized with the multiple application of one-dimensional filters as described above.

Circular symmetry requires that the filter impulse response h be a function of $t_1^2 + t_2^2$. Thus the one-dimensional filter $g(t)$ which is to be applied in each direction must be chosen such that

$$h(t_1, t_2) = g(t_1^2)\,g(t_2^2) \quad \text{with} \quad g(t_1^2)\,g(t_2^2) = g(t_1^2 + t_2^2) . \tag{9}$$

A solution of this functional equation is

$$g(t) = e^{-t^2} \tag{10}$$

In accordance with the previously prescribed procedure, we now attempt to find a one-sided, one-dimensional filter whose impulse response is

$$f(t) = e^{-t^2} \quad t > 0$$
$$= 0 \quad t < 0 \tag{11}$$

There is no filter whose Laplace transform is a rational function of s which has the impulse response (11). However, there is a class of filters with transfer functions rational in s, and with impulse responses approximately (11) in the vicinity of $t = 0$. These can be found by expanding (11) in a Taylor series about the origin, transforming the result, and equating the coefficient to the large s expansion of a rational function. Thus, for

$$f(t) = 1 - t^2 + \frac{t^4}{2} - \frac{t^6}{6} + \cdots \text{ as } t \to 0 \qquad (12)$$

we want

$$F(s) \sim \frac{1}{s} - \frac{2}{s^3} + \frac{12}{s^5} - \frac{120}{s^7} + \cdots \text{ as } s \to \infty \quad (13)$$

The first few of a family of approximations to (13) are:

$$F_3(s) = \frac{s + a}{s^2 + as + 2}$$

$$F_5(s) = \frac{s^2 + as + 4}{s^3 + as^2 + 6s + 2a}$$

$$F_7(s) = \frac{s^3 + as^2 + 10s + 4a}{s^4 + as^3 + 12s^2 + 6as + 12}$$

Here $F_n(s)$ is an approximation to (13) good through terms in s^{-n}. All of the transfer functions are stable for any positive value of the parameter a. A numerical example of the two-dimensional filter related to $F_3(s)$ with $a = 3$ has been worked out. The results are presented as contours of constant impulse response in Fig. 1. Note that in the vicinity of the origin the contours are approximately circular, as predicted. Two cross sections of the "impulse mountain" are shown in Fig. 2.

The example above can be continued to display explicitly the transfer function to be used for each separate filtering operation. The one-sided one-dimensional transfer function is

$$F(s) = \frac{s + 3}{s^2 + 3s + 2} \qquad (14)$$

The two-sided one-dimensional transfer function is then (in summation form)

$$\frac{s + 3}{s^2 + 3s + 2} + \frac{-s + 3}{s^2 - 3s + 2} .$$

We can combine the terms to obtain

$$\frac{12}{(s^2 + 3s + 2)(s^2 - 3s + 2)} = \frac{\sqrt{12}}{s^2 + 3s + 2} \cdot \frac{\sqrt{12}}{s^2 - 3s + 2} .$$

It can thus be seen that the required transfer function to produce the two-dimensional impulse response of Fig. 1 is $\frac{\sqrt{12}}{s^2 + 3s + 2}$. This transfer function is used for four cascaded filtering operations: positive t_1 direction, negative t_1 direction, positive t_2 direction, negative t_2 direction.

Examination of the two-sided transfer function shows that this filter has a high frequency attenuation of 24 db per octave and no phase shift at any frequency.

IX. CONCLUSION

When a two-dimensional image is to be processed with a filter for which the response at an output point depends upon a large region of the input image, the methods of this paper find their principal utility. When the methods are applicable, there is a potential for decreasing execution time by several orders of magnitude in the digital case, or for putting the processing steps into a form suited for physically realizable filters in the analog case.

These simplifications are achieved by (a) restricting the two-dimensional impulse response to the product of two one-dimensional impulse responses -- a loss of generality that is satisfactory in many practical applications if one judiciously chooses the proper one-dimensional functions -- and (b) simplifying the one-dimensional operations so that recursive filters may be used in digital computation, and ordinary electrical filters may be used in the analog case. One of the important features is the time-reversal step, accomplished by routine programming (digital) or by using a reversible storage medium such as a tape recorder (analog). The time reversal permits impulse responses which are nonrealizable in the ordinary sense. In particular symmetrical impulse responses become possible.

REFERENCES

1. L. S. G. Kovasznay and H. M. Joseph, "Image Processing," Proc. I. R. E., Volume 43, pp. 560-568, May 1955.

2. E. E. David, Jr., "Digital Simulation in Research on Human Communication," Proc. I. R. E., Volume 49, pp. 319-329, January 1961. See especially pp. 327-8.

3. W. D. Fryer and W. C. Schultz, "Digital Computer Simulation of Transfer Functions," Proc. Nat. Elec. Conf., Volume XVII, pp. 419-420, 1961.

STAT 4. [redacted] "Digital Computer Simulation of Transfer Functions,"
STAT [redacted]
To be published, 1962.

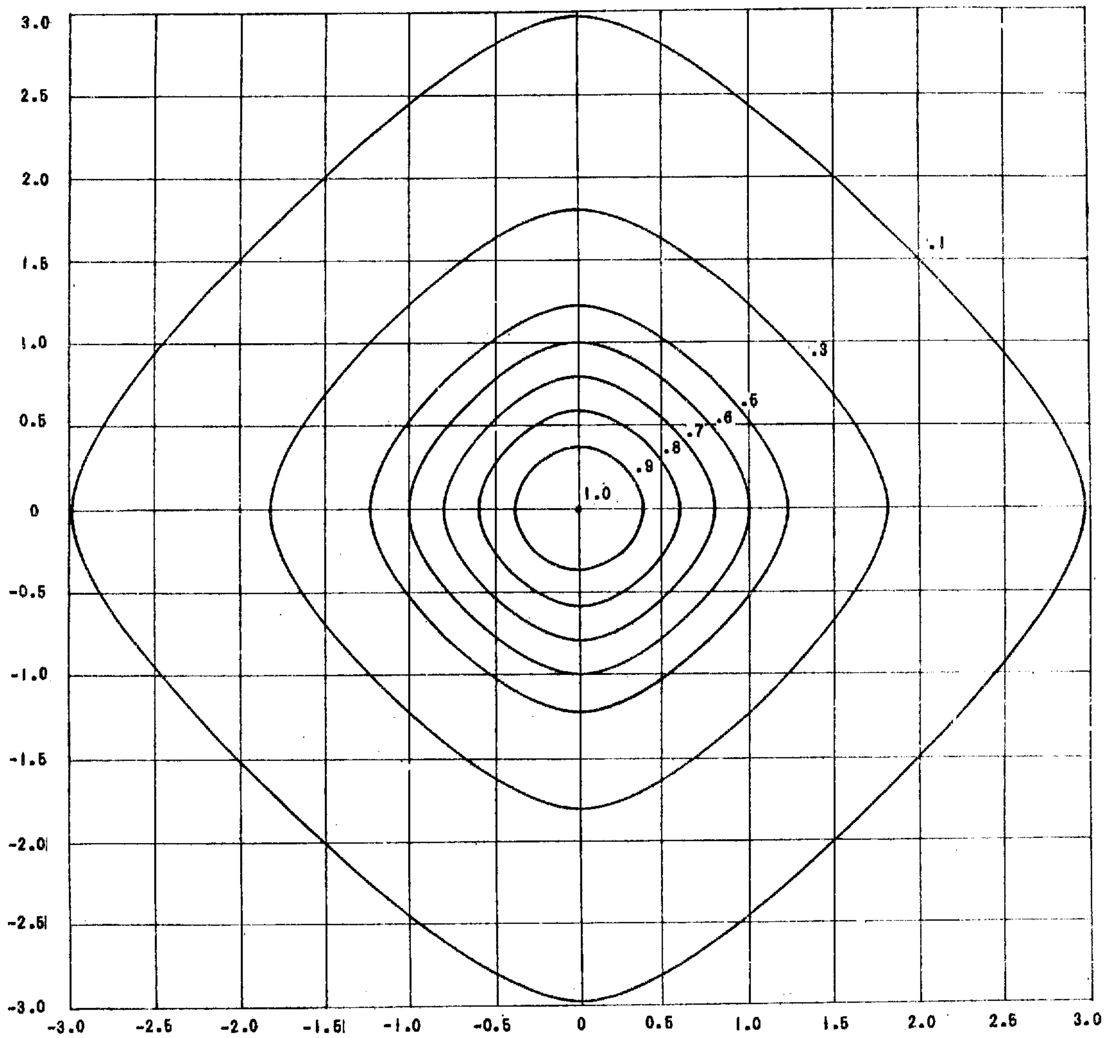


Figure 1 CONTOURS OF CONSTANT IMPULSE RESPONSE FOR THIRD ORDER APPROXIMATION TO GAUSSIAN FILTER

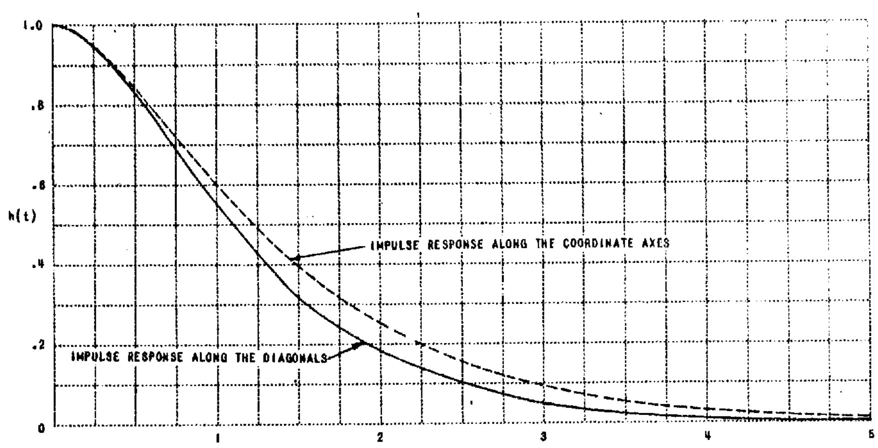


Figure 2 IMPULSE RESPONSE ALONG THE AXES AND DIAGONALS FOR THE SAME FILTER AS FIGURE 1