

ESTONIAN ACADEMY OF SCIENCES
DIVISION OF SOCIAL SCIENCES

S Y M P O S I U M

MATHEMATICAL PROCESSING
OF CARTOGRAFIC DATA
(TALLINN, DECEMBER 18 – 19, 1979)

SUMMARIES, pp. 63-66

T a l l i n n 1979

MONOTONIC SYSTEMS ¹ IN SCENE ANALYSIS

J.E. Mullan, L.K. Vyhandu

In the now classic book by Duda and Hart [1] there are three main methods to clean up a scene – gradient finding, space smoothing and regional analysis. The stopping rules they use are purely intuitive and do not have a sound mathematical background.

The primary problem dealt in this paper is the following. Given a two-dimensional array of gray-level values (scene) we try to separate undefined objects from the background (to clean the scene). The problem of an exact identification of objects is not touched upon in this paper.

We all know that a precise description of the noise and distortion process, which defines the mapping between the objects and their image in the scene, allows us to employ statistical decision theory techniques optimizing some objective criterion (e.g. minimum error, minimum risk). However, in most practical problem situations, the required noise and distortion model is not available, nor is it feasible to attempt to construct one.

We introduce a simple language to describe a scene as a set W of elements. In terms of this language scene cleaning is the same as extracting such subset of W that its elements do reflect their intrinsic dependence more efficiently than any other subset.

¹ Monotonic Systems has been first introduced in by J. E. Mullan, “On The Maximum Principle for some Set Functions,” *Tr. Tallin. Politech. Inst., Ser. A*, No. 313, 37-44 (1971), <http://www.data laundering.com/download/modular.pdf>. Detailed description, J.E. Mullan, “Extremal Subsystems of Monotonic Systems, I,II,III,” *Automation and Remote Control*, 1976, 37, 758-766, 37, 1286-1294; 1977, 38. 89-96, compatible with the notation vocabulary of current paper is available for download from <http://www.data laundering.com/mono/extremal.htm>.

1. Monotonic systems. Kernel

Let us suppose there is a system W with a finite number of elements. Each element has a numerical measure of its weight (influence) in the system. Further let us suppose that for every element $\alpha \in W$ there is a feasible discrete operation, which does change as well as the weight of α and the weights of any other element β of the system. If the elements in W are independent, then it is natural to suppose that a change in the weight α does not change the value of another element β .

System W is called monotonic if the operation upon any element $\alpha \in W$ brings about a change in the weight levels of other elements only in the same direction as α itself is changed.²

To use the method of monotonic systems we have to fulfil two conditions.

1. There has to be a function π , which gives a measure (weight) $\pi(\alpha)$ of influence for every element α of the monotonic system W .
2. There have to be rules f ³ to recomputed the influences of the elements of the system in case there is a change in the weight of one element.

These conditions leave a lot of freedom to the scientist to choose the influence functions and rules of influence change in the system. The only constraint we have to keep in mind is that the functions f and π have to be compatible in the sense that after eliminating all elements α of the system W the final weights of $\alpha \in W$ must be equal to zero.

² Eliminating (deleting) an element from the system is an example of such type of actions. Noted by JM

³ Also, reasonable example of rules f and the like may be found in Appendix I belonging to J. Mullat, see. E. Ojaveer, J. Mullat, and L. Võhandu, "A study of infraspecific groups of the Baltic east coast autumn herring by two new methods based on cluster analysis," *in: Proceedings of Estonian Studies According to International Biology Program VI* [in Russian], Isd. Akad. Nauk ESSR, Tartu (1975), available for download <http://www.data laundering.com/download/herring.pdf>. Noted by JM.

A monotonic system of the scene represents a system for which an extraction of an arbitrary element of the two-dimensional array decreases the weight level of all other elements.

If we assume the set \bar{H} ⁴ to be extracted elements of the two-dimensional array, we can construct the weight levels in the remaining set H by introducing four values as follows:

$$\begin{aligned} r_i &= \sum_{k=1}^m a_{ik} \quad (i = 1, \dots, n), \quad v_j = \sum_{k=1}^n a_{kj} \quad (j = 1, \dots, m), \\ d_\ell &= \sum_{i+j=\ell} a_{ij} \quad (\ell = 1, 2, \dots, n+m), \\ s_p &= \sum_{p=m+i-j} a_{ij} \quad (p = 1, 2, \dots, n+m-1). \end{aligned}$$

(i.e., row sums, column sums and sums of diagonals in both directions – up and down the array).

We define the weight of the element in the row i and column j on the set H as

$$\pi_H a_{ij} = (r_i - a_{ij}) \cdot (v_j - a_{ij}) \cdot (d_\ell - a_{ij}) \cdot (s_p - a_{ij}) \cdot a_{ij} \quad (1)$$

One could easily test that $\pi_H a_{ij}$ has the needed property to decrease with the extraction of any other element (different from a_{ij}) in accordance with the definition of a monotonic system [2].

We define a kernel of a two-dimensional array W as a subset H of its elements on which the global maximum of the function

$$F(H) = \min_{a_{ij} \in H} \pi_H a_{ij}$$

is reached.

⁴ $\bar{H} = W \setminus H$, noted by JM.

2. Kernel splitting of two-dimensional arrays

We describe the algorithm by steps. Let $n \times m$ two-dimensional array $A = \left\| a_{i j} \right\|_n^m$ have integers from the interval 0 to 255.

A1. Compute four tables of sums for the array A : R -, V -, D - and S -table (row sums r_i ($i = 1, \dots, n$), column sums v_j ($j = 1, \dots, m$), sums of up-diagonals d_ℓ ($\ell = 1, 2, \dots, n + m$), sums of down-diagonals s_p ($p = 1, 2, \dots, n + m - 1$).

A2. Define the element α of the monotonic system W as the element $a_{i j}$ of the array A and compute the weight $\pi(w) = \pi_w a_{i j}$ for the element $w = a_{i j}$ using (1).

A3. Find two numbers

$$L = \min_{w \in W} \pi(w) \text{ and } U = \max_{w \in W} \pi(w).$$

A4. Apply the following stratum splitting process with a given threshold u^o ($L \leq u^o \leq U$).⁵

Compare the weights the weights $\pi(w)$ throughout the array A with the threshold u^o . If $\pi(w) \leq u^o$, then the compared element w has to be extracted from the system W and the weights of all other remaining elements have to be recalculated by making the needed changes in the sums of tables R , V , D and S .

This comparing process (we call it *stratum*(u^o)) with a given threshold level u^o and recalculation of weights will be stopped only when the inequality $\pi(w) > u^o$ is satisfied for all not yet extracted elements.

A5. After each stratification step there are two possibilities:

- a) *stratum*(u^o) extracts all elements of the monotonic system W ;
- b) not all elements of W are eliminated.

In case a) put $U = u^o$ and return to the step A4 for a new pass through the data. In case b) find from the set of non-extracted elements the element with the minimal weight (denoted by *inf*(u^o)). Clearly $\text{inf}(u^o) \geq u^o$. On the set of non-extracted elements the procedure *stratum*(*inf*(u^o)) is repeated (Step 4A.).

⁵ Stratum algorithm was first described by J. Mullat in *Avtomatica i Telemekhanika*, No. 10, pp. 84-94, October, 1979, submitted October 3, 1978. Its original name was not stratum but a sequence of elements W in concord with respect to the threshold u^o .

It can be seen from the description of the algorithm that the main part of it tries to find an exact value of the maximal threshold \tilde{u} , which extracts all elements of the array A , but the algorithm do not extract the final set Γ_p from the array with $\tilde{u} - \varepsilon$ (ε – arbitrary small positive number). This property of the set Γ_p is equivalent to the property of kernel. In order to prove this we consider a notion of a sequence of subsets of elements belonging to the array A , see [2].

Let $\langle \alpha_0, \alpha_1, \dots, \alpha_{n \cdot m - 1} \rangle$ be an ordered sequence of distinct elements of the array A , which exhaust the whole array. From the sequence $\langle \alpha_0, \alpha_1, \dots, \alpha_{n \cdot m - 1} \rangle$ we construct an ordered sequence of subsets of A in the form

$$\langle H_0, H_1, \dots, H_{n \cdot m - 1} \rangle$$

with the help of the following recurrent rule

$$H_0 = A, H_{i+1} = H_i \setminus \alpha_i, i = 0, 1, \dots, n \cdot m - 2.$$

A sequence $\langle \alpha_0, \alpha_1, \dots, \alpha_{n \cdot m - 1} \rangle$ of A is called a defining sequence relative to the weight function $\pi_H(\alpha)$ if there exists in sequence $\langle H_0, H_1, \dots, H_{n \cdot m - 1} \rangle$ a subsequence of sets $\langle \Gamma_0, \Gamma_1, \dots, \Gamma_p \rangle$, such that

- a) $\pi_H(\alpha_i) < F(\Gamma_{i+1})$ for every element $\alpha_i \in \Gamma_i \setminus \Gamma_{i+1}$;
- b) in the set Γ_p there does not exist a proper subset L , which satisfies the strict inequality $F(\Gamma_p) < F(L)$.

One can easily see that the set $W = A$ of all elements of the array A and the set extracted by the algorithm with the maximal threshold (denoted by Γ_p), e.g. $\langle A, \Gamma_p \rangle$ ⁶, represent a defining sequence, in accordance with its definition. Therefore by the theorem proved in [2] the set Γ_p is a kernel of the two-dimensional array A .

After we have found a kernel for a given two-dimensional array, we can simply delete the elements belonging to the kernel (as missing values) and repeat the whole procedure of the algorithm so many times as necessary.

REFERENCES.

1. R. O. Duda, P.E. Hart (1973). Pattern Classification and Scene Analysis, New York, Wiley.
2. J. E. Mulla, "Extremal Subsystems of Monotonic Systems, I," *Avtom, Telemekh.*, No.5, (1976), pp. 130–139, <http://www.data laundering.com/download/extrem01.pdf>.

⁶ Actually the algorithm constructs a bit longer sequence $\langle A = W = \Gamma_{i_0=0}, \Gamma_{i_1}, \dots, \Gamma_{i_p=p} \rangle$ of subsets each time it must pass the step A5 while computing the *stratum*(*inf*(u^0)). It is a simple exercise to check that this sequence satisfies p. b) property of the defining sequence [2]. Noted by JM