Urszula Stańczyk
Silesian University of Technology
Institute of Informatics

Properties of image processing logic functions

Transformations operating on a finite window can be expressed by a logic function, whose variables correspond to points from the neighbourhood defined by the processing window [2]. Assuming operating on binary images, pixels values can be directly interpreted as variables values of a logic function, allowing for evaluation of the output. Basic morphological transformations of dilation and erosion can be expressed respectively as logical sum and logical product.

Logic functions can possess properties that are useful in their transformations or implementation. These properties can be detected either at a logic synthesis stage or, by some auxiliary algorithms applied to formal representation types of a function. Among important functional properties there are such as degeneracy, symmetry, monotonicinity, and decomposability.

Properties a logic function specifying image transformation can be viewed either from the point of theory of logic circuits or image processing. Due to the fact that independent variables of the function correspond to considered pixels of an image, contained in a processing window, these variables acquire some geometrical meaning apart from inherent algebraic and logic.

An important characteristic of these functions is the fact that they are always completely specified, which results from necessity to produce the output value for all input combinations. In switching circuit theory do not care conditions are assumed for situations that are not valid for a considered problem. In image processing none of input combinations can be called invalid—images can contain all kinds of data, as long as they are within specified class. For all patterns in the processing window there is always defined an answer giving the value of the output pixel.

Degeneracy property of logic functions means that a function of some N variables actually depends on some fewer variables. Degeneracy is useful because it helps to reduce the amount of data to be stored—there should be stored only those variables the function actually depends on, which can be a significant difference.

However, in the case of image transforming Boolean functions there is no room for degeneracy. Each variable of a function support corresponds to some point in a processing window. If the function was vacuous in any of the variables, it would mean that the point the variable corresponds to should not be contained in the processing window in the first place, since the sole change of its value never causes the function to change as well. Thus a properly defined processing window ensures the function to be non-degenerate, and instead of testing the function for this property it is better to check whether the window contains only essential points.

Functional values in symmetric functions depend only on the number of variables equal zero or one, no matter which variables are equal zero and which are equal one. Totally symmetric functions are encountered less frequently than partially symmetric functions, in which only in some subset of variables a function is symmetric.

Function symmetry in image processing can cause misunderstanding with the geometrical symmetry that is often encountered, the two of which, significantly different, use the same name of property but in two different fields. Thus it is necessary to consider a function symmetry by detailed analysis of the definition. For a symmetric function the output depends only on the number of ones and zeros and not their distributions among variables. In image processing there is a group of operations with such approach to pixels in the processing window and they are rank order operators [3]. They are described by logic threshold functions among which there can be named special cases of minorative, majorative and median functions. These functions are totally symmetric and their symmetry comes from the image transformation definition, which

means that once again instead of testing the function for symmetry more efficient approach is to analyse the transformation and since such analysis is necessarily performed at the synthesis of a logic function stage, at this very stage the symmetry property for the function is known.

In the context of image processing monotonicity is an important concept, especially when it is limited to increasing transformations. Increasing operations maintain containment relationships in processed images. If a transformation is increasing, the logic function expressing it also possesses this property. From morphological transformations these that are not increasing are Hit-or-Miss Transformations and all those using them as building blocks. Increasing logic functions have so-called *stacking* property and they are used to describe *stack* operators, the special case of which constitute rank order operators, among which there are median operators.

Monotone functions constitute a class of functions that is closed under superposition, which is verified by transformations defined with such functions. For example, dilation and erosion operations are increasing. Performing first erosion, then dilation of the erosion result should also maintain containment relationships, that is it should be increasing and it really is, because opening transformation, which corresponds to such superposition of dilation and erosion, is increasing. In the same way closing, which also is a superposition of dilation and erosion, is increasing. Superposition of openings gives an opening, superposition of closings gives a closing, necessarily preserving monotonicinity of operations.

Increasing logic functions are characterised by the unique sum of products form in which none of variables is complemented, which is why they are often called positive functions.

The class of monotone functions can be enlarged by considering functions that are monotone increasing for some subset of variables and monotone decreasing for the rest. Such functions are called mixed monotone or unate if partition of a support set is non-trivial and disjoint. To this class belongs a logic function expressing Hit-or-Miss transformation and the partition of variables is provided by the definition of its composite structuring element, whose constituent two sub-elements must be necessarily disjoint thus partitioning the processing window pixels corresponding to function variables.

Testing monotonicity can relay on the definition of this property, that is checking whether there is preserved partial ordering in the functional value corresponding to ordered strings of variables values. Exhaustive tests obviously are time-consuming and to avoid that there can be introduced some limitation, for example to local tests verifying whether monotonicity is maintained for randomly chosen pairs of strings that are logically adjacent. Another option is to obtain minimal sum of product form of the function and check for complementations on variables. If none of them is complemented, the function is positive and by that monotone increasing. If all variables are complemented the function is monotone decreasing and if there exists a disjoint and non-trivial partition of variables into two subsets, the variables in one subset being uncomplemented and the variables in the second complemented, the function is unate.

Decomposability property of logic functions is very useful in their implementations, since, if existing, it allows for producing the functional value with lowered realisation costs through construction of some number of simpler sub-functions. Testing for decomposability in fact corresponds to some attempts to find non-trivial decompositions for a given function with respect to some choices of variables. Thus the process of analysis of functional properties for some function becomes the process of simplification of this function.

References

[1] R. M. Haralick, S. R. Sternberg, X. H. Zhuang, *Image analysis using Mathematical Morphology*, IEEE Transactions on Pattern Analysis and Machine Intelligence 9, No. 4, July 1987.

[2] H. J. A. M. Heijmans, *Mathematical Morphology: a modern approach in image processing based on algebra and geometry*, SIAM Review 37, No. 1, March 1995.

[3] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, Inc., New York 1991.