Philips Research Laboratory
Ave. Em. van Becelaere 2, Box 8
B–1170 Brussels, Belgium

Report R.469

REAL-TIME DETECTION OF CONNECTED COMPONENTS IN BINARY PICTURES

Christian Ronse and Pierre A. Devijver

February 1983

Abstract: Separation of objects from their background is a major problem in pattern recognition and scene analysis. It finds applications in many fields ranging from optical character recognition to biomedical engineering. It is often associated with other processes like thinning, vectorization, etc.

Besides the question of their mathematical validity, algorithms for extracting connected components in real-time are subjected to implementation constraints resulting from the actual state of technology, namely,

- the limited size of the memory: if we wanted to store, say, a whole A4 sheet of paper digitized at facsimile resolution in computer memory, we would need storage space for about 4×10^6 bits, which is frequently much too large with regards to the amount of meaningful information contained in it;
- the need for real-time processing: in a number of applications, features (e.g., connected components) should be extracted at the time they are encountered in order to avoid memory congestion and permit a smooth cooperation of cascaded processes.

In virtually every application, it is normally expected that the detection of any connected component be accompanied by some concise description of that component. We start from the premise that one should not overlook the amount of information which is made available at the output: For pattern recognition purposes, it is preferable to receive a minute description of the component structure rather than, say, the sole chain-code of its borders.

In this report, we outline an algorithm—up to the code level—which satisfies these requirements to a great extent.

In terms of memory requirements, we assume that the input image consists of an arbitrary number of rows of given length, and is read in a single raster scan. The program maintains essentially two types of memories: a working memory whose size is linear in the size of the rows, and an object memory containing *i)* partial descriptions of connected components under detection, and *ii)* surrounding relations between them.

The algorithm detects the completion of the scan of any connected component at the time the last pixel of that component is being read. Concomitantly, the description of that component is transfered to an output buffer in a time linear in the size of that component. Thus, the algorithm operates in *real-time* at the component level.

For what concerns component descriptions, the algorithm performs a decomposition of components in 2-dimensional structural elements called *blocks* and *hinges*. Geometrical information about components is encoded in the geometry of these elements as well as adjacency relations between them. Topological information is encoded in terms of surrounding relations between *outer-edges* of connected components of both the figure and the background.

The program offers a number of options, *e.g.,* 4- or 8-connectivity. The two major options concern the level of detail of the geometrical and topological descriptions desired in the output. They are called *full* or *restricted adjacency*, and *full* or *restricted surrounding* respectively. They can be activated interactively and are fully orthogonal in the sense that any combination of both is feasible. Thanks to these options, our approach is quite versatile, and the algorithm can be readily adapted to the needs of a wide variety of possible applications.

The algorithm is presented in successive stages through the chapters of this report. Every attempt was made to keep the presentation self-contained. The theoretical background is recalled whenever needed, and the discussion is carried up to the darkest corners of the implementation. The complete program can be found in the appendices. The algorithm was implemented in *Pascal*, and was tested on the VAX 11/780 computer at the Philips Research Laboratory Brussels.

Contents