# A NOVEL FEATURE EXTRACTION METHOD BASED ON SEGMENTATION OVER EDGE FIELD FOR MULTIMEDIA INDEXING AND RETRIEVAL

*Serkan Kiranyaz, Miguel Ferreira and Moncef Gabbouj*

Institute of Signal Processing, Tampere University of Technology, Tampere, Finland
{serkan.kiranyaz, miguel.ferreira, moncef.gabbouj}@tut.fi

## ABSTRACT

In this work we focused on visual content extraction. Digital images have some basic features, such as texture, colour and shape. Through low-level bit processing, and using colour intensity changes across pixels, we try to achieve some form of mid-level meaningful description of the image. Starting from the well-known Canny Edge algorithm we convert the image map to an edge map. From the edges, and performing analysis based on perceptual rules, we achieve a segmentation of the image, which is proved sound for content-based indexing and retrieval purposes.

## 1. INTRODUCTION

The shape of objects is one of the most important semantic information sources and therefore it is a powerful descriptor. Shape features can be obtained from regions of uniform properties or from abrupt changes in these properties between two uniform regions in the picture. Evidence shows that the human visual system (HVS) focuses on edges and tends to ignore uniform regions. This process, known as *Lateral Inhibition*, is similar to the mathematical *Laplacian*.

Visual content extraction was focused in this work. From basic visual features we aim to achieve, through low-level bit processing, some form of mid-level meaningful description based on the segmentation information extracted from the edge field. Apart from the biological motivation, the reason we used an edge-based approach is that it brings generality, as edges are generally present across any type of visual content, coloured or not, textured or not. Another advantage over pixel-based segmentation algorithms is that they usually suffer from local colour and texture changes. This results in over-segmentation or under-segmentation of images, into such segments, which have no use for retrieval purposes. The underlying object might lie in several local regions where colour or texture behaviour is merely static, or it might be merged with background sharing similar colour or texture.

Furthermore, our analysis uses image edges since they mostly represent boundaries of object shapes. However, these edges might also be incomplete, missing, noisy or saturated. In this work certain semantically meaningful analytical steps are performed over the edge field in order to achieve the extraction of object boundaries. The edge pixels are transformed into *sub-segment*s that are then become the subject of further analysis performing extraction of closed-loop (CL) and non-closed-loop (NCL) edge segmentations. The analysis will focus on perceptually meaningful merging of *sub-segment*s, according to local rules such as proximity, good continuation of edges and according to the global concern of finding objects, i.e. closure. CL segments represent the objects (their boundaries) or the part of the objects that are present in the image whereas NCL segments are for background scene or incomplete (overlapped) objects. Finally feature extraction (*FeX*) is performed over both types of segments in order to achieve an edge-based shape description of the image. This description will discriminate both types of segments, by providing contour-based shape description for CL segments, and background information description for NCL segments. Forming the whole process as a *FeX* module into MUVIS framework [1], [2], allows us to test the overall performance in the context of multimedia indexing and retrieval area.

The whole process is automatic (i.e. without any supervision, feedback or training involved) and adaptive, which, if needed, iteratively tunes the default parameters set by the user. This paper is organized as follows: in Section 2 a generic overview of the method and particularly CL/NCL segmentation will be introduced. We discuss the integration of the proposed method in the context of shape-based multimedia indexing and retrieval in Section 3. Section 4 presents the experimental results and Section 5 concludes and suggests topics for future research activity.

## 2. CL/NCL SEGMENT ANALYSIS OVER EDGES

Our visual system perceives objects and extracts their shape information according to intensity contrasts between different regions. These contrasts, or intensity changes, can be identified and marked algorithmically. We call these borders between different intensity regions as edges. An efficient shape analysis can then be performed according to these edges if edge field provides visual accuracy and perceptually meaningful boundaries.

Therefore, the proposed approach mainly consists of three processing: First an efficient edge field is extracted. Afterwards *sub-segment*s are extracted from the edges and finally the segments are formed from the *sub-segment*s. Detailed illustration is shown in Figure 1.

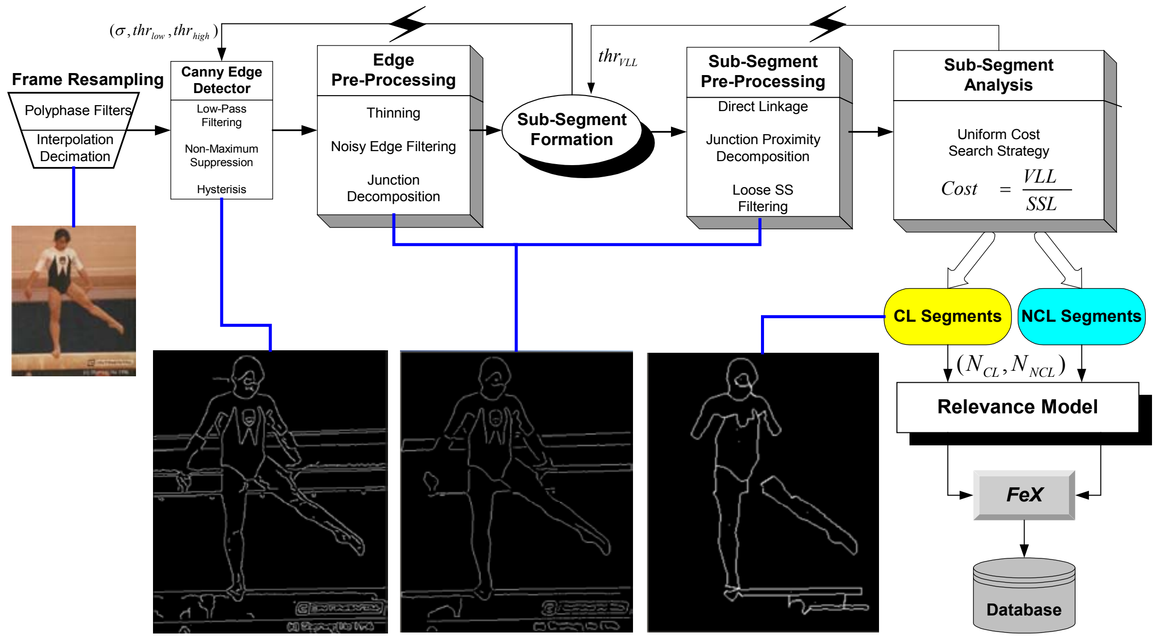


**Figure 1: Overview of the proposed segmentation based *FeX* algorithm.**

## 2.1. *Sub-Segment* Analysis Over Edge Field

John Canny developed an algorithm on edge detection, which has been widely used since its publication in 1986 [3]. The generality and visual accuracy of this method are particularly suitable for our purposes.

An interesting issue raised is how to choose the starting elements for edge analysis towards grouping. Pixel-level analysis does not lead to clear identification of image structures, since HVS does not perceive individual pixels (or photons) separately. When looking at a digital edge image, how exactly does our perception combine together the edge pixels, to form natural shape outlines? The first low-level structures HVS is able to perceive clearly are continuous, connected, series of pixels. However, abrupt direction changes in a connected series, can lead to the impression of two different edge sections, which are considered separately in mental grouping. We start from the connectedness to build the basic elements, which will be the starting subject to our analysis. We call these elements as *Sub-Segments*, through which we achieve both forms (CL and NCL) of *Segmentation*. A *sub-segment* can basically be defined as the series of connected edge pixels with two end-points and they are formed from the following pre-processing steps directly applied onto edge field:

- **Thinning**
- **"Noisy Edge" Filtering:** While finding and tracing our *sub-segments*, a small number of edge pixels is not to be considered as a *sub-segment* and we eliminate such edges from the map for further reducing noisy edges.
- **Junction Decomposition:** At a junction we count how many valid branches originate from it. In order to achieve a better segmentation from the *sub-segment* analysis, junctions need to be decomposed into their branches (separated *sub-segments*). For that, our algorithm breaks the tracing procedure of a segment whenever we have more than one valid branch to follow at some edge pixel location. This pixel location is considered as en endpoint to that *sub-segment* traced and removed from the edge map.

## 2.2. CL and NCL Segmentation from *Sub-Segments*

Both types of segmentation are performed over *sub-segment* list extracted in the previous step. Due to space limitations NCL segmentation is not described in this paper. For CL segmentation, a search process links one or more individual (and distinct) *sub-segment* using an abstract linkage element, so-called *virtual link,* according to a cost function based on HVS perception rules. There is however a need for applying some pre-processing filtering before the search process in order to prevent infeasible search times and inaccurate segmentation due to saturated *sub-segment* field.

### *2.2.1. Pre-Processing over Sub-Segments*

This pre-processing step is similar to one presented before, but this time it is directly performed over *sub-segments* within three levels:

- **Direct Linkage:** The *sub-segments* with particular endpoints, which have only one other endpoint to link to, can be immediately linked. In case they belong to two different segments, the gap between their endpoints is created as a straight line and they are merged into one segment.
- **Junction Proximity Decomposition:** As we mentioned earlier, the problem that Canny edge field usually presents when dealing with junctions. What the Canny operator frequently does, is to output a T-junction as a normal continuous edge between two branches, with the third branch of the junction being separated by a small gap. Similar to the argument made earlier, if such

composition is detected then the connected branches should be decomposed into two distinct *sub-segments*. Therefore, the problem is solved by *breaking* the single segments at points, which lie within a close neighbourhood of any other segment's endpoint. The breaking distance is determined by the virtual link threshold ($thr_{VLL}$), which is chosen as the maximum linkage length allowed.

- **Loose *Sub-Segment* Filtering:** In case there is one endpoint that cannot reach any other within virtual link threshold, its *sub-segment* cannot be a part of a CL segment. In order to reduce the number of states that the search algorithm will have to process, these "loose" *sub-segments*, which will not lead to a solution, are removed beforehand.

In Figure 1, the effect of the pre-processing steps is shown. Note that the main perceptual cues do not change.

*2.2.2. Search Strategy for CL Segmentation*

Considering that we have *N sub-segments*, with two endpoints each, we find $N_{CL}$ objects retrievable from the edge map. A CL segment, $S_{CL}$, consisting of $l$ *sub-segments* ($ss_i$) can be defined as:

$$S_{CL} = \sum_{i=0}^{l} ss_i \quad \left| ss_0 = ss_l \right. \qquad (1)$$

We define a state space, which will consist of these *2N* endpoints. Each state is represented by its respective endpoint coordinates and the segment identification number it belongs to. By choosing one of the endpoints of a particular sub-segment as being the starting state, we automatically set the end state as its other end-point. It becomes clear that this particular problem fits well in the graph-based search approach. In the general case, there are multiple paths, which lead to the solution node.

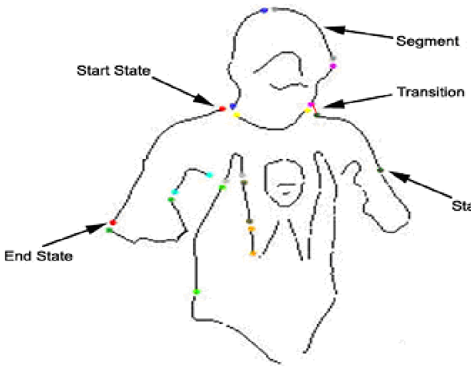


**Figure 2: State space for a given *sub-segment* layout.**

From each of the *2N* endpoints, we can link to each of the remaining *2N-1* endpoints. These transitions are virtual connections to other endpoints (states) as shown in Figure 2. They are reversible, i.e. valid for both directions. In formulating our search problem we try to apply all the domain knowledge we can, in order to increase its performance and accuracy.

HVS perceptual rules indicate that consecutive segments forming an object are usually within a close distance in the image. Adding this constraint to the search space, candidate v*irtual links* are created between the endpoints. V*irtual link* threshold (*VLink*) establishes the maximum distance for two endpoints to be virtually linked, and it is obtained empirically from the image dimensions and *sub-segment* length statistics. Its purpose is to reduce the number of state transitions, as to make our problem more tractable. These virtual links will be the operators for node expansion. They supply the transition network of our state space.

The search will start with the biggest segment in the segment list. The start and end states are defined as the two endpoints of this segment. A cost function is assigned to transitions according to perceptual rules explained earlier. The (selected) final path closing the initial segment, in case multiple paths exist, will be the one with less cost. The cost function ($C(e_1, e_2)$), to jump from an endpoint $e_1$ belonging to segment $s_1$ to an endpoint $e_2$ in segment $s_2$ will be expressed by:

$$\boxed{C(e1,e2) = \frac{VLL\,(e1,e2)}{SSL\,(s2)}} \qquad (2)$$

where *VLL* is the virtual link length and *SSL* represents the *sub-segment* length operators. Obviously this cost function favours shorter virtual link (jumps) to a neighbouring endpoint belonging to a longer segment.

Uniform cost search strategy is used for its completeness and optimality, and also for pruning in the search tree therefore converging faster to better paths. Once a (candidate) closed-loop path is detected, its cost is also used to prune the next search and we use and update the minimum cost value so far obtained for pruning the rest of the search operations. This drastically reduces the time complexity of the search algorithm. Once all the search operation is completed, the path giving minimum cost value defines a CL segment. After concluding the search and whether a closed path is found or not, we remove both states of the segment from the state set.

## 3. INDEXING AND RETRIEVAL

For a MUVIS multimedia database, indexing process involves the feature extraction applied onto visual items such as key-frames of the video clips or frame buffer of the images. A particular *FeX* module formed from the proposed algorithm provides both feature extraction of visual frames and also the similarity distance function for retrieval. The proposed *FeX* module parameters such as Canny Edge Detection parameters, number of CL/NCL segments ($N_{CL}, N_{NCL}$) to be extracted, the feature vector dimension, etc. are set by the user depending on the database content and type and then might be tuned by the algorithm adaptively whenever needed. Assuming these parameters are set appropriately, in the next section we will present the feature extraction process from CL segments and due to space limitations feature extraction from NCL segments is not described in this paper.

### 3.1. Feature Extraction from CL Segments

The number of $N_{CL}$ segments having the biggest boundary length with the least cost values are chosen among the whole set of CL segments extracted and each of them is then used for feature extraction.

As shown in Figure 3, we used the histogram of normalised angular first moment calculated from the centre of mass (*CoM*) of a CL segment, as the main feature vector. Apart from unit normalization, this feature vector is further designed to be translation, rotation and size invariant. Translation invariance comes naturally as a result of using the *CoM* as the reference point. Size (area) invariance and unit normalization are achieved by dividing each bin value in the histogram by the area of the CL segment. Rotation invariance is achieved during the similarity distance calculation of the retrieval phase and will be described in the next section.

Once all feature vectors for all CL segments in a frame are calculated, they are indexed into feature file for that particular frame (image or key-frame) and appended into the MUVIS database structure.

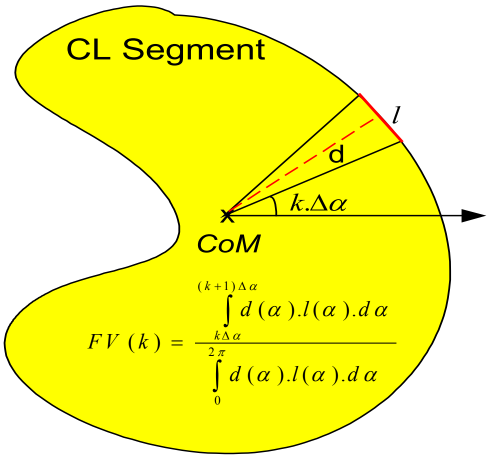


**Figure 3: Feature Vector formation from 1st moment inserted in kth bin of the N-bins angle histogram.**

### 3.2. Retrieval for CL Segments

Retrieval process in MUVIS is involved in a query by example (QBE) operation initiated by the user. The features of the query item are used for (dis-) similarity measurement among all the features of the visual items in the database. Ranking the database items according to their similarity distances yield the retrieval result.

Between the query frame and a particular frame from the database, the feature vectors of all ( $N_{CL}$ ) segments from both of them are used for similarity distance calculation with the following matching criteria: for each CL segment in query frame, a 'matching' CL segment in the compared frame is found. The minimum similarity distance is simply used as the matching criteria. Afterwards the total similarity distance will be the sum of matching similarity distances of all the segments present in the query frame.

In order to achieve rotation invariance, the feature vector of a CL segment (angular histogram bins) is slide one bin at a time with the other vector kept stationary and the similarity distance is calculated per step. The minimum similarity distance is chosen for that particular CL segment of the query frame. Since this sliding operation on the histogram bins represents the feature vector of the rotated CL segment, the rotation invariance is therefore achieved.

## 4. EXPERIMENTAL RESULTS

The experiments are performed to test the segmentation efficiency with respect to HVS perceptive criteria (subjective test) and retrieval performance via query-by-example (QBE) of several images and video clips within MUVIS databases indexed by the proposed *FeX* module.

### 4.1. Visual Evaluation of Segmentation

The classical evaluation of a segmentation method examines how good the object(s) in the scene can be extracted. In our case however it is equally (or even more) important to extract the major parts of the objects rather than the entire object itself. This is mainly due to the fact that the parts of the objects usually provide better similarity measure than the overall object, which might change its shape quite often and hence becomes infeasible to compare among the similar content available. For example there are several images containing the similar semantic content, i.e. "gymnast" or "human body" as one example is shown in Figure 4 (a). Therefore, it is more convenient to extract the parts of the body (the object) such as the legs, the arms, the torso and the head, some of which usually provides a stationary shape information that is perceptually similar to the others (however they probably might be rotated or having zoom/shrink effects among other (similar) examples but the algorithm is designed to be robust to such effects).

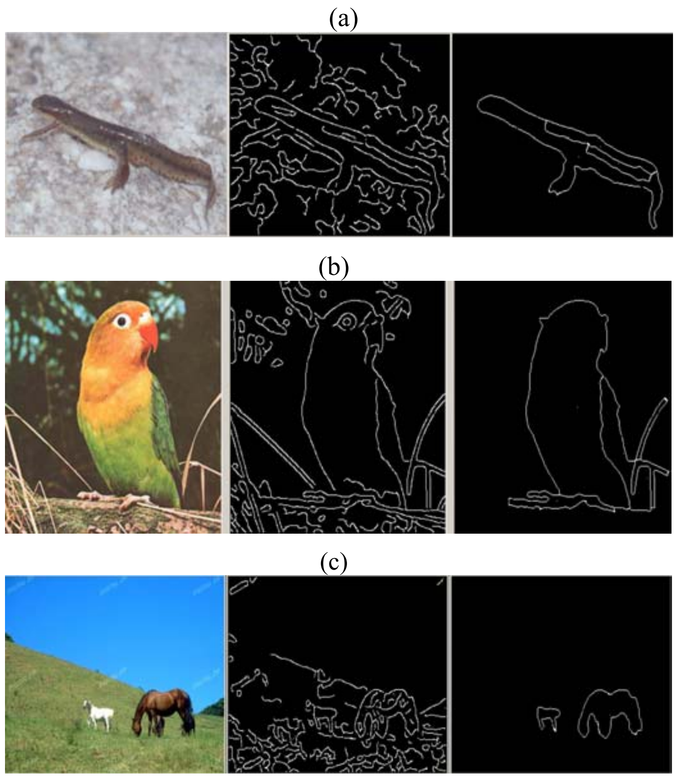


**Figure 4: Three CL segmentation examples, the original image (left), its Canny edge field (middle) and the CL segments (right).**

From several segmentation experiments, we observed that if the Canny edge field captures the boundaries of the object in an acceptable perceptual performance, then the proposed method could extract the significant CL segments belonging to the major parts of the object of interest. Three typical examples are shown in Figure 1 and Figure 4, as $N_{CL} = 7$ for Figure 1, $N_{CL} = 3$ for (a), $N_{CL} = 4$ for (b) and $N_{CL} = 2$ for (c) in Figure 4.

### 4.2. Retrieval Performance

In order to test the retrieval performance we have indexed two image databases using the proposed *FeX* module. The first database is ***binary shape*** database, which contains 1400 binary images. Using this pre-segmented database we examined on the effectiveness of the proposed feature vector and as two typical examples shown in Figure 5, we observed a significant retrieval performance, i.e. on average 85% or higher *Recall* level. The second one (***sports*** database) contains 500 images carrying various sports content (i.e. gymnastics, athletics, ice-hokey, F-1, football, etc.). Performing several retrieval experiments, we observed promising results, particularly on some experiments we achieved such a retrieval performance that can otherwise be retrieved neither by color nor by texture as one typical example shown in Figure 6.

## 5. CONCLUSION

The proposed feature extraction method based on segmentation and *sub-segment* analysis over Canny edge field achieves the following major innovative properties:

- It brings a novel and alternative approach for the automatic (image) segmentation problem by using (object) boundaries instead of pixel-based color and/or texture information.
- The overall algorithm is unsupervised and designed by the HVS perceptual characteristics to yield meaningful segmentation as the final product. The major perceptual fact we refer is “the objects have closed boundaries”.
- The search space is reduced by adaptive pre-processing steps, which mainly employ “simplifications” over the edge field with minimal degradations on the object boundaries. Pruning the trace by the cost of the last minimum path further optimizes the search operation.
- The cost function ensures the optimal paths (the object boundaries) giving the biggest possible segmentations with minimum virtual link usage.
- Finally, the features extracted from the overall segmentation scheme provide an effective indexing and retrieval performance via visual query.

Current and planned research work includes: design a better and more semantically meaningful feature vector from CL/NCL segments, apply an adaptive merging operation to obtain global object(s) from the parts and test the proposed algorithm with the existing segmentation algorithms and MPEG-7 edge descriptors. By using the proposed *FeX* module integrating the “query by sketch” capability into MUVIS framework is also considered.

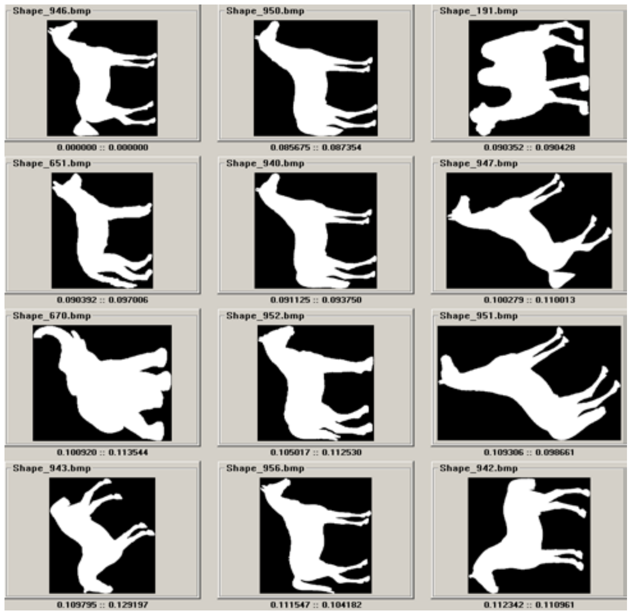


**Figure 5: Typical (12-best) retrieval examples from *binary shape* database. Left-top image is queried.**

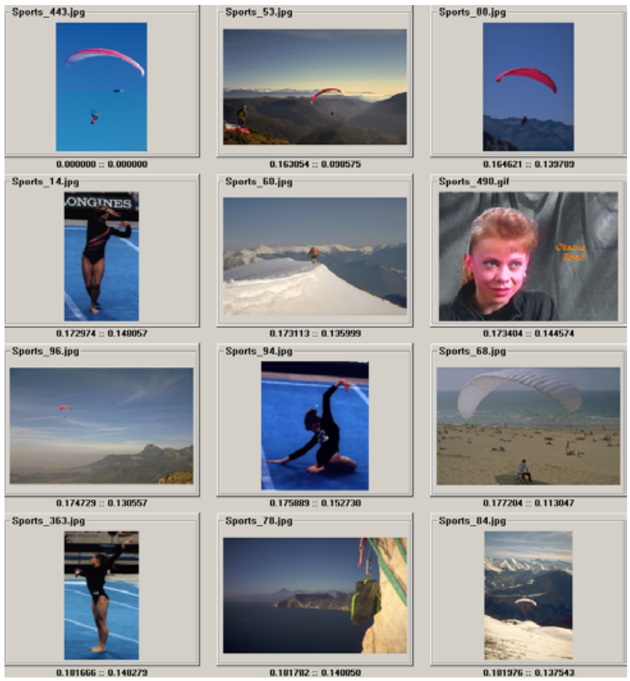


**Figure 6: A retrieval example from *sports* database.**

## 6. REFERENCES

[1] S. Kiranyaz, K. Caglar, O. Guldogan, and E. Karaoglu, “MUVIS: A Multimedia Browsing, Indexing and Retrieval Framework”, Proc. of Third International Workshop on Content Based Multimedia Indexing, CBMI 2003, Rennes, France, 22-24 September 2003.

[2] MUVIS. http://muvis.cs.tut.fi

[3] John Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679-698, November 1986.