Fuzzy Curve Pyramid*

Horst Bischof Walter G. Kropatsch
Dept. for Pattern Recognition and Image Processing
Technical University Vienna, Treiltlstr. 3/1832
A-1040 Vienna, e-mail: bis@prip.tuwien.ac.at

Abstract

This paper describes an extension of the binary curve pyramid to curves with strengths. In particular we propose fuzzy relations to represent curve strength. We show how fuzzy relations can be processed in a pyramidal framework. The advantage gained by this method is that the properties of the binary curve pyramid are preserved, and that we gain some additional properties, which can be used in the pyramid construction phase.

1 Introduction

Curves convey important information about an image, they may describe either a

- boundary of a region, or
- the skeleton of an elongated thin region

Typically recognition of curves involves 2 steps. First, local instances of the curves are detected (e.g. edge detection). Second, these elementary segments are connected to longer curve segments that may be matched with higher level curve models.

This paper deals with the second step of the curve recognition process. In particular we present a hierarchical method to find long curves in images in a logarithmic number of steps. This is achieved with a symbolic pyramid structure. The symbols represented in the pyramid are curve relations, which connect the sides of a pyramidal cell. In order to be able to handle also curve strength (i.e. which is typically delivered by an edge detector) we will introduce the notation of fuzzy curve relations. The pyramidal process will be extended to these fuzzy curves. The advantage of this approach is that we can, additionally to the length of the curve, take also its strength into account. In particular, we will extend the $2 \times 2/2$ curve pyramid introduced in [2, 3] to handle also fuzzy curve relations.

*This work was supported by the Austrian National Fonds zur Förderung der wissenschaftlichen Forschung under grant S7002MAT.

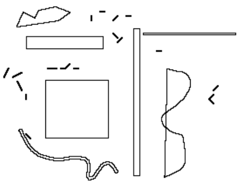
2 $2 \times 2/2$ Curve Pyramid

The basic idea is that linear structures of images are represented by curve relations. A cell of the pyramid is considered as an observation window through which the curve is observed. A single curve intersects this window twice. Only the intersection sides (N, E, S, W) are stored in the cell (i.e. a curve relation). We denote a curve relation by AB, where $A, B \in \{N, E, S, W, F\}$ (F is the special end code when the curve ends in a cell). The basic routines of building the next level of the pyramid are (Fig. 1):

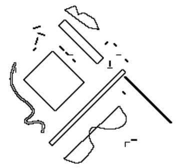
1. Split - subdivision of the cells contents by introducing a diagonal.
2. Transitive closure - the curve relations of four cells are merged by computing the transitive closure of all relations (i.e. $AB, BC \Rightarrow AC$).
3. Merge - the curve relations of the new cell are selected.

The $2 \times 2/2$ curve pyramid has several interesting properties. A curve remains connected until it is completely covered by one cell (then it disappears). Very important is the *length reduction property* (i.e. the number of curve code elements decrease after every reduction step), see [2]. This important property relates the highest level up to which a curve is still represented to the area that is traversed by the curve. This implies that short curves will disappear after a few levels, and only long curves will be represented in higher levels of the pyramid (see Fig. 1d).

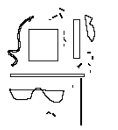
This property has been used for *structural noise filtering* [3], which first builds the curve pyramid up



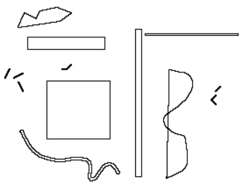
(a) Level 0



(b) Level 1



(c) Level 2



(d) Curve relations down projected from level 6 to original resolution

Figure 1: $2 \times 2/2$ binary curve pyramid

to a certain level and then deletes all curves which are not represented in the levels below.

3 Fuzzy Curve Pyramid

The (binary) curve pyramid as described above has the disadvantage that it can only represent the presence/absence of a curve segment. This binary decision has several disadvantages:

- We must decide already at the base level about the presence/absence of a curve segment. This typically involves the determination of a threshold for an edge detector. This is a problematic step, because either we miss curve segments which cause disconnected curves (therefore the curve will disappear after a few levels) or we add too many segments which might connect otherwise disconnected curves.
- Another problem with the binary representation of curves arises when different curves meet in one pyramidal cell, this might cause ambiguities in the representation due to the restricted storage capacity of the pyramidal cells.

Therefore it would be advantageous to have a non binary representation of curves in a pyramid. Since a curve introduces a relation between two sides of a pyramidal cell, a straight forward generalization is to consider fuzzy relations instead of binary ones. The strength of the curve is represented by the grade-membership of the fuzzy relation.

Definition 1 (Fuzzy Relation)
Let $X = \{x_1, \ldots, x_k\}$ be a $k-$element fuzzy (or crisp) set. The set $R \subseteq X \times X \times [0 \ldots 1]$ is a fuzzy relation. We denote a fuzzy relation between x_i, x_j with grade-membership $\mu(x_i, x_j)$ by $(x_i R x_j, \mu(x_i, x_j))$ $(x_i, x_j \in X, 0 \leq \mu(x_i, x_j) \leq 1)$.

We say for $\mu(x_i, x_j) = 0$ that there is no relation between x_i and x_j. The relation is symmetric if $\mu(x_i, x_j) = \mu(x_j, x_i)$ $\forall x_i, x_j \in X$. In what follows we will always assume symmetric relations.

Let us now describe the individual steps for constructing the $2 \times 2/2$ fuzzy curve pyramid. The steps are generalizations of the binary $2 \times 2/2$ curve pyramid to fuzzy relations. The basic idea is to replace the logical operations (and, or) by min and max operations.

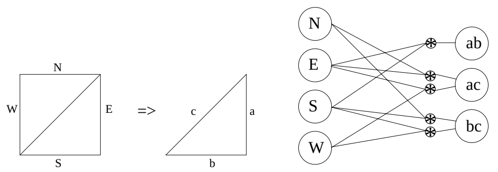
3.1 Fuzzy Curve Representation

In [3] an 8-bit code was suggested to store the curve relations. In order to simplify the discussion we use a simpler code (without end-codes), the extension to the equivalent of the 8-bit code is straight forward.

A cell in a pyramid has four sides, labeled N, E, S, W For each of these sides we use b bits (typically $b = 8$) to store the fuzzy relation. If a curve with strength μ intersects one of these sides we store a value of μ on the corresponding side. Therefore a single curve passing the cell of a pyramid will “activate” two sides of the pyramid with the same strength. It should be noted that the decisions are only made local for a particular cell, therefore a neighboring cell may have different strengths. In order to guarantee consistency we require that a curve relation does not end at the side of a cell (e.g. a NS relation in one cell and no relation at the neighboring cell). If more than one curve intersect a side of the pyramidal cell we store the maximum of all these curve strengths.

3.2 Fuzzy Split

The first operation to perform is to split the curve relations by the diagonal, as shown in Fig. 2a. The curve relations of the cell are transferred to the triangle with the sides a, b, c. Fig. 2b depicts a diagram



(a) Split curve relation by diagonal (b) Fuzzy split operation

Figure 2: Split operation

for computing these relations. In this and the following diagrams small circles with stars indicate min operations whereas the labeled nodes perform a max operation. For example the strength of ac is given by:

$$ac = \max(\min(N, E), \min(W, E))$$

3.3 Fuzzy Transitive Closure

The important operation in constructing a curve pyramid is the transitive closure operation. It constructs out of the curve segments of the four triangles at level n curve relations for the cell at level $n+1$. We make use of the transitivity of relations, i.e. $aRb, bRc \Rightarrow aRc$. For the case of fuzzy relations this generalizes to:

$$(aRb, \mu(a,b)), (bRc, \mu(b,c)) \Rightarrow \\ (aRc, \mu(a,c) = min(\mu(a,b), \mu(b,c))) \quad (1)$$

We compute the fuzzy transitive closure in two steps. We first merge the upper and the lower triangles in parallel (Fig. 3a) and then merge the resulting two triangles to the final square, as depicted in Fig. 3b.

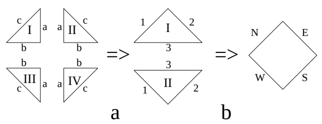
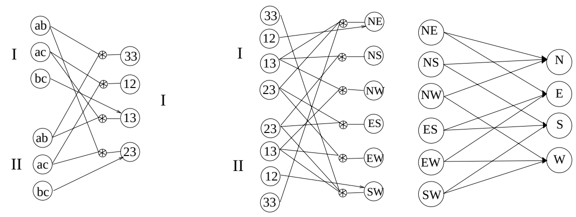


Figure 3: Transitive Closure operation

The diagram for the upper two triangles is shown in Fig. 4a left and for merging two triangles to a square in Fig. 4a right.

It is important to note that a single curve passing a cell will activate the corresponding unit according to the minimum strength of the curve segments.



(a) Transitive Closure (b) Merge

Figure 4: Fuzzy transitive closure and merge operation

3.4 Fuzzy Merge / Curve Selection

The third operation simply converts the relation code used for processing the transitive closure to the code used by the cell of the next level. This is depicted by the diagram of Fig. 4b.

Putting all these operations together results in a reduction step for a single cell. In order to handle U-turns and end-codes correctly some additional nodes are necessary.

4 Properties of Fuzzy Curve Pyramid

We have described the individual steps necessary to construct level $n+1$ from level n of the fuzzy curve pyramid. In this section we show several properties of the fuzzy curve pyramid.

Property 1 (Length Reduction) *The Fuzzy curve pyramid has the length reduction property (i.e. it reduces the number of curve code elements after every reduction step) similar to the $2 \times 2/2$ binary curve pyramid.*

Proof The proof is essentially the same as for the binary case (i.e. all curve relations with grade-membership $\mu > 0$ have to be considered). One can convert fuzzy curve relations to binary ones by setting an arbitrary threshold Θ at the grade-membership μ.

Property 2 (Minimum Strength) *If a single curve intersects the receptive field of a cell at level n, the grade-membership of the relation at level n will be the minimum of the grade-memberships of the curve segments of the base level.*

Proof Since we are dealing with a single curve we can proof this property recursively, therefore we need to consider only two successive levels of the pyramid. We can consider each individual step in the pyramid construction separately. Sinve in the case of a single curve the split and merge operations are simple copying mechanisms, we have only to check the transitive closure operation. From the definition of the fuzzy transitive closure in eq. (1) one can see that if two segments are connected the grade-membership of the resulting segment is assigned the minimum of the grade-membership of the individual segments. If we apply this recursively property 2 follows. qed.

For the case when two or more curves pass the receptive field of a cell we can proof a more general property: We define the strength of a curve as the minimum strengths of its segments (see above) then the strength of a fuzzy curve relation AB is the maximum strength of all curves connecting the side A with side B. This allows us to formulate following property:

Property 3 (Maximum Strength) *Let C be a curve in the base of the fuzzy curve pyramid with strength $\mu(C) = \omega_0$. Then this curve will remain connected by the fuzzy reduction process and the strength of all fuzzy curve relations derived from C will be greater or equal to ω_0, i.e. the strength of the curve is preserved.*

Proof 1. Assume that C has the greatest strength of all curves in the image. In this case taking the maximum at all operations where curve segments are merged preserves the strength as long as the curve does not disappear.
2. If the strength of C is not maximum it could happen that it gets merged with a curve of higher strength. The combined segment will receive the higher strength, e.g. $\mu(s_1 \cup s_2) = \max(\mu(s_1), \mu(s_2)) \geq \mu(s_1)$. qed.

The two last properties can be combined in the **Minmax Principle**, which is satisfied by the fuzzy curve pyramid.

Property 4 (Minimax Principle) *The strength of a curve is the minimum of the strength of its segments. And the strength of a side of a cell in the curve pyramid has the maximum strength of all curves intersecting this side.*

5 Enhancements

Until now we have not fully exploited the information provided by the grade-membership of the relations. This will be discussed in this section.

5.1 Ambiguity reduction

The basic idea is that due to the grade-membership of the relations we can discriminate between curves of different strength. This is important when more than one curve meets in a single cell. Consider as an example Fig. 5. We see that two curves with different strength meet in a common cell. In the case of the binary curve pyramid we cannot discriminate them, therefore we introduce all possible relations and the two curves are merged. Whereas in the case of the fuzzy curve pyramid we know due to the grade-membership that there are two curves (a strong one from North to West and a weak one from South to East).

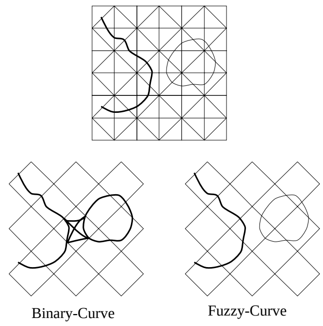


Figure 5: Example where curves get merged in the binary curve pyramid, and remain separate in the fuzzy curve pyramid

There are two possible ways how to modify the algorithm to introduce this ambiguity reduction:

1. Modification of the Split step: We can modify the split step in order to take the different grade-memberships into account. However this does not eliminate all ambiguities; e.g. curves with the same strength, two curves entering at the same side of a cell.

2. Eliminate Merge and Split Before the merge step we have all the necessary information about the curve relations. If we eliminate the merge and split step we do not introduce ambiguities. This can be done in the following manner: A single cell in a $2 \times 2/2$ pyramid has two fathers (see Fig. 6A). Therefore we can connect the result of the transitive closure operation directly to the two fathers of a cell, in the way it is depicted in Fig.6B, i.e. we eliminate the merge step and perform the split operation. Fig. 6C,D shows an example how the curve relations are transferred to the fathers. One can see that only those activations get activated where a curve is present. The disadvantage of this operation is that we need 10 links between the cells, whereas we need only 8 links for the original algorithm, however the number of links (processing steps) in a cell is reduced. It is important to note that as long as the curves are not parallel (entering and leaving at the same sides) they can be discriminated. In the case of parallel curves they will get merged. The resulting activation of the new curve segment is the maximum of the activations of the two parallel segments. It should be noted that the methods we have proposed in this section do not alter the properties we have discussed above.

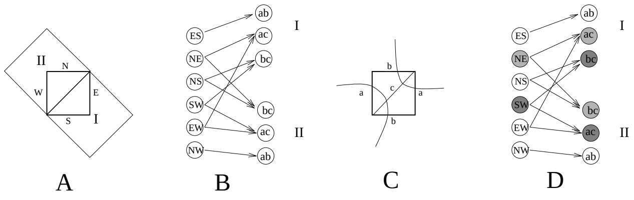


Figure 6: Direct Connection between the cells (without Merge step)

5.2 Curve selection

The binary as well as the fuzzy curve pyramid extract two important properties of curves, that is length and connectedness but the fuzzy curve pyramid additionally processes curve strength as. We can use this additional information to select certain curves in the pyramid construction or down projection phase, for example we build the pyramid up to certain level, then remove all curve relations with a grade-membership below a threshold. If we then down project the remaining relations we get a structural filter which takes also the curve strength into account.

6 Evaluation and Conclusion

In this paper we have proposed fuzzy relations to represent curves with strength. We have shown that the fuzzy relations can be processed in a similar manner as binary relations. Moreover the properties of the binary curve pyramid are preserved, additionally we have other information about the curve like the minimum grade-memberships of the curve segments. The method we have presented shares many similarities to [6]. As one of our next steps we would like to consider the inclusion of the maximum flow algorithm as in [6] to process fuzzy relations. The Minimax principle of curves follows a similar objective as snakes [1]: it combines the cross sectional strengths along the path of the curve. However, unlike snakes, the minimax principle uses min and max operators to combine “internal” and “external” forces instead of a linear (weight) combination. A lot of research has been done using hierarchical representation of edge information (i.e. edge pyramids) [4, 5] however these approaches store numerical information in the pyramidal cells, whereas the fuzzy curve pyramid combines symbolic with numeric information.

Compared to the binary curve pyramid the fuzzy curve pyramid needs more storage space. For the $2 \times 2/2$ curve pyramid we need only 8 bit per cell to represent the curve relations. For the fuzzy curve pyramid we need 8×8 bit per cell (if the end relations are also considered). The advantage gained is that we can process curves with different strength. Besides the storage requirements the computational complexity of the binary and the fuzzy curve pyramid are the same; i.e. we can build both pyramids in a logarithmic number of steps on a parallel hardware.

References

[1] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Computer Vision*, pages 259–268, London, England, June 1987.

[2] Walter G. Kropatsch. Curve representations in multiple resolutions. *Pattern Recognition Letters*, Vol. 6(No. 3):pp.179–184, August 1987.

[3] Walter G. Kropatsch. Elimination von "kleinen" Kurvenstücken in der $2 \times 2/2$ Kurvenpyramide. In E. Paulus, editor, *Mustererkennung 1987*, Informatik Fachberichte 149, pages 156–160. Springer Verlag, 1987.

[4] D.J. Park, R.H. Park, S.U. Lee, and J.S. Choi. Hierarchical edge detection using the bidirectional information in edge pyramids. *Pattern Recognition Letters*, 15:65–75, 1994.

[5] A. Schrift, Y.Y. Zeevi, and M. Porat. Pyramidal edge detection and image representation. In *SPIE Proc. Visual Communications and Image Processing '88*, volume 1001, pages 529–536, 1988.

[6] Zhenyu Wu and Richard Leahy. An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 15(11):1101–1113, November 1993.