An improved intuitionistic fuzzy estimation of the area of 2D-figures based on the Pick's formula

Krassimir Atanassov, Evgeniy Marinov,
Peter Vassilev and Alexander Alexandrov

12 April 2019

"Two polygons are equidecomposable iff they have the same area."

Wallace–Bolyai–Gerwien theorem

Introduction

An intuitionistic fuzzy estimation of the area of 2D-figures based on the Pick's formula is proposed. It is an improvement of the previous results of the first two authors, and it consists in an iterative procedure for estimation of the area surrounded by a simple closed curve in the real 2D space. Pick's Formula is employed for calculating the area surrounded by a special types of polygons. Also a formula for intuitionistic fuzzy estimation for the area surrounded the curve is given. The result is a numerical method permitting the implementation of the algorithm in any procedural language. The iterative process stops when a satisfactory small enough limit between the upper and lower estimation has been reached.

Quick overview of Pick's formula

Pick's theorem provides a simple formula for calculating the area S surrounded by a polygon in terms of the number I of grid-points in the interior of the polygon, i.e. not touching any of the sides, and the number B of grid-points on the boundary, i.e. placed on the polygon's perimeter. Assuming that we have a grid with grid-step equal to one, Pick's formula provides the number of unit squares through the following expression:

$$S = I + \frac{B}{2} - 1 \tag{1}$$

Therefore, assuming a grid-step equal to l_0, the formula provides

$$S(l_0) = l_0 \Big(I + \frac{B}{2} - 1 \Big) \tag{2}$$

for the area surrounded by the given polygon.

An example of simple curve

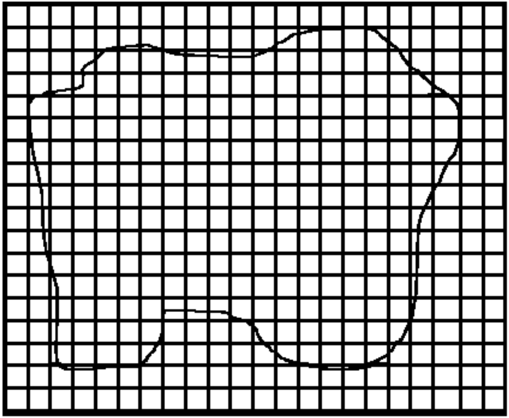

Figure: Simple curve and a grid with lines parallel to the two axes.

The procedure

Given a 2D Cartesian coordinate system Oxy and a simple curve parametrized by

$$\vec{r}(t) = (r_1(t), r_2(t)) : [0, 1] \to \mathbb{R} \times \mathbb{R} \tag{3}$$

we are going to split the underlying space to a grid with lines parallel to the two axes (as above)
This mesh has to be fine enough for a good estimation of the given curve. Finally, we will consider the smallest possible part of the grid having borders, consisting of squares from the grid, which enclose the curve and we shall denote it:

$$\mathcal{S}(\vec{r}, l_0), \tag{4}$$

where l_0 is the length of the grid cell.

Intuitionistic fuzzy measure

In order to define the intuitionistic fuzzy measure of the area enclosed by the initial curve, we are going to introduce an algorithm that provides a special type of polygon surrounding the initial curve $\vec{r}$. We suppose that the given parametrization $\vec{r}$ of the curve provides a positive orientation, i.e. anti-clockwise orientation. That is, following the path of $\vec{r}$ on the figure, the outer part will be on the right-hand side, while the inner one will lie on the left-hand side. Applying the same procedure for the curve $\overrightarrow{r}^{-1}(t) = \vec{r}(1-t)$, for which the inner part of $\overrightarrow{r}^{-1}$ coincides with the outer one of $\vec{r}$, while its outer part coincides with the inner part of $\vec{r}$. Therefore, in order to build an inner and outer polygon enclosing the given initial curve, it suffices to describe only the outer (surrounding) polygon of $\vec{r}$. These two polygons will be used to build a hull of the curve.

BUILDING THE OUTER POLYGON OF $\vec{r}$

Taking initially the grid with a step of length l_0, supposed to be sufficiently small, we pass along the curve with $\vec{r}(t)$ letting the parameter t run from 0 to 1. Along its path, $\vec{r}(t)$ intersects the lines of the grid in different points and may pass through various squares of the grid leaving/entering them through their vertices or edges.

Neighbor Square I

Let us consider all the nodes of the grid as being marked as white in the beginning. Subsequently, passing through some nodes while the procedure is running, they will be marked either as “black” or “not-allowed”, i.e. nodes already found in the $P - stack$ (program stack). In short, the procedure passes through some nodes constructing the outer polygon by edges and diagonals of squares of the grid putting them in the so called $P - stack$. It has to be mentioned that every node N of the grid has exactly eight neighbors because the node belongs to exactly four squares. These four squares have $4 \times 4 = 16$ nodes (vertices), four of them taken two times in the sum and the node in consideration is counted four times as well. Therefore, the neighbor nodes of the node in consideration are exactly $16 - 4 - 4 = 8$. They can be *diagonal* or *edge* nodes in respect of the current vertex, i.e. four diagonal

Neighbor Square II

neighbors and four edge neighbors as well – to be denoted *DiagN*(N) and *EdgeN*(N), respectively. Let us consider all the nodes of the grid as being marked as white in the beginning. Subsequently, passing through some nodes while the procedure is running, they will be marked either as "black" or "not-allowed", i.e. nodes already found in the $P - stack$ (program stack).
Let us denote $NeighborN(N) = DiagN(N) \cup EdgeN(N)$ and define the square consisting of all this neighbor nodes as the *neighbor square* of the current node and denote it by $NeighborS(N)$. The border of the neighbor square consists of eight sides, to be denoted by $\partial NeighborS(N)$. There are also four inner sides of length the grid-step l_0 and for diagonal sides (edges) of length $\sqrt{2}l_0$, denoted by $StraightE(N)$ and $DiagE(N)$, respectively. Let us denote by $InnerE(N) = StraightE(N) \cup DiagE(N)$ all the inner sides of $NeighborS(N)$. We can also consider $NeighborS(N)$ as the union

Neighbor Square III

of eight right isosceles triangles, i.e. triangles with a right angle ($\frac{\pi}{2}$), and also two equal angles (sides). The family of these eight triangles will be denoted by $NeighborT(N)$.

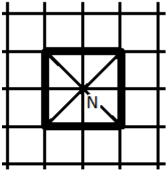

Figure: The vertex N with the neighbor square $NeighborS(N)$ and the border $\partial NeighborS(N)$ in bold.

Minimal polygon

The grid is in principle infinite but for the sake of simplicity we will concentrate our attention on the minimal part of the grid, which contains the curve as shown below. This constraint is important because on the basis of this picture we are going to introduce intuitionistic fuzzy estimation further.

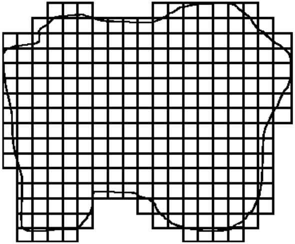

Figure: Minimal polygon with edges parallel to the axis part of the grid containing the curve. The area of this figure is $\mathcal{S}(\vec{r}, l_0) = 264 \times l_0$.

The procedure

Just after a node (vertex) has been colored "black" and put into the $P - stack$, consider only the "allowed" ones of its neighbors as candidates to be taken as "next node" when building the oriented polygon. Starting by the first vertex lying near the curve in the outer part of $\vec{r}$, we are going to describe a procedure which will add new nodes to the $P - stack$ in a sequence building correctly the outer polygon. If, according to the rules of the algorithm, we come to a step where the "next" node, say N'', which should be added to the stack, would be such one that the atomic particle $N'N''$ of the polygon crosses the curve, we stop the procedure and start again with a twice shorter grid-step.

The procedure producing the outer polygon I

We take as initial grid-step a sufficiently small number l_0 and start from a point $N \in \overline{OuterN}(\vec{r})$ as shown below

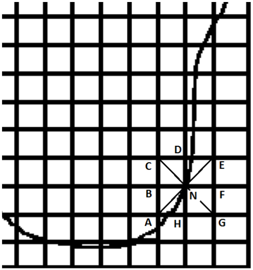

Figure: The vertex N with the neighbor square and all the neighbor nodes A, B, C, D, E, F, G and H. This is the right lower corner from Figure 1 .

The procedure producing the outer polygon II

Supposing that the procedure has already accomplished few steps and N is the last node in the $P - stack$, let us describe the next step. In what follows, by $R(t)$ we will denote the point from the curve $\vec{r}$ at $t \in [0,1]$, i.e. $\vec{r}(t) = OR(t)$ where O is the origin of the coordinate system. For the point N we want that for some $t_0 \in [0,1]$ the point $R(t_0)$ lies on some edge from $StraightE(N)$ and therefore we have that $d(R(t_0), N) \leq l_0$. Let for $t_1 \geq t_0$ the point $R(t)$ enter the inside of some of the four neighbor triangles $T \in NeighborT(N)$. Suppose that $R(t)$ then goes outside of the triangle T and enters T again, i.e. there are $t_3 > t_2 > t_1$ such that $R(t_1), R(t_3) \in T$ while $R(t_2) \notin T$. If this happens, then we break and start the procedure again with a grid-step $\frac{l_0}{2}$. Therefore, we can suppose that the grid-step l_0 is small enough so that the above

The procedure producing the outer polygon III

situation can not happen, i.e. for any right angle triangle T with vertices - three neighbor nodes on the grid, it is not possible that

$$(\exists t_1, t_2, t_3 \in [0,1])(R(t_1), R(t_3) \in T \,\&\, R(t_2) \notin T).$$

Going back to the current last point N in the $P - stack$ with $R(t_0)$ lying on one of the $StraightE(N)$, there are two possible cases

The procedure producing the outer polygon IV

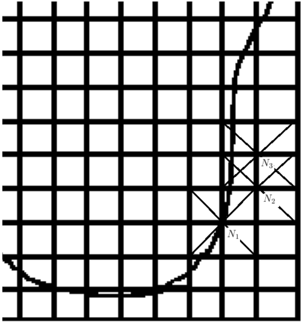

Figure: The vertices $N_1, N_2, N3$, which are the first three nodes from the $P - stack$. This is the right lower corner from Figure 1.

The procedure producing the outer polygon V

- IF $R(t_0, 1)$ lies in the neighbor square of N, i.e. $NeighborS(N)$. That means that the first point from the $P - stack$, N_1, belongs to $NeighborS(N)$, i.e. $N_1 \in NeighborN(N)$. If all the points of NN_0 lie in the outside of $\vec{r}$, then the **procedure finishes successfully** in letting N be the last node in the $P - stack$. Otherwise, if NN_0 has a point belonging to the inside of $\vec{r}$, then **break and restart the procedure with a grid-step $\frac{l_0}{2}$.**
- ELSE there is $t' \in (t_0, 1]$ for which $R(t') \in \partial NeighborS(N)$. If all the points of $NR(t')$ lie in the outside part of $\vec{r}$, then the **procedure continues** in letting $R(t')$ be the current last node in the $P - stack$. Otherwise, if $NR(t')$ has a point belonging to the inside of $\vec{r}$, then **break and restart the procedure with a grid-step $\frac{l_0}{2}$.**

Regarding the inner polygon

Thus, the whole procedure of finding an outer polygon has been described.

As we have already mentioned, we may apply the same procedure for the curve $\overrightarrow{r}^{-1}(t) = \vec{r}(1-t)$, for which the inner part of $\overrightarrow{r}^{-1}$ coincides with the outer one of $\vec{r}$, while its outer part coincides with the inner part of $\vec{r}$. Let us suppose that the procedure has stopped successfully with a grid-step $\frac{l_0}{2^{n_1}}$ in finding the outer polygon and $\frac{l_0}{2^{n_2}}$ for the inner polygon. We then take the minimum of the two grid-steps to be $\frac{l_0}{2^{j_0}}$ ($j_0 = \max(n_1, n_2)$) and start the procedure again to find a new outer polygon (if $n_1 > n_2$) or to find a new inner polygon (if $n_1 < n_2$). Therefore, we get the outer (in blue) and inner polygons (in green) in the same grid with grid-step $\frac{l_0}{2^{j_0}}$ as shown below

Outer and inner polygon

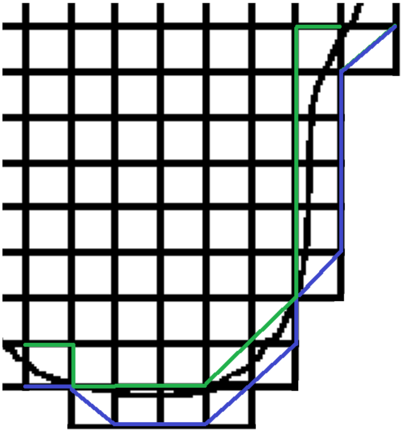

Figure: The inner and outer polygons with the end grid-step $\frac{l_0}{2^{j_0}}$. This is the right lower corner from Figure 3.

Outer and inner polygon

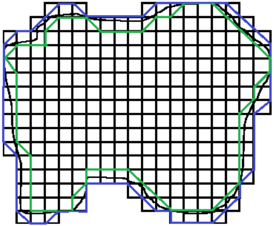

Figure: The whole picture of the inner and outer polygons with the end grid-step $l_1 = \frac{l_0}{2^{j_0}}$ of the first iteration. The inside area of the inner polygon is 205.5 while the area of the outer one is 252.5.

Intuitionistic fuzzy estimation of the inner area I

The notion of intuitionistic fuzzy set (or abbreviated as IFS) provides a very intuitive and natural tool for an adequate estimation of the area enclosed by a simple continuous curve. Following Zadeh, a fuzzy set (FS) in X is given by

$$A^{'} = \{\langle x, \mu_{A'}(x)\rangle | x \in X\} \quad (5)$$

where $\mu_{A'}(x) \in [0,1]$ is the *membership function* of the fuzzy set $A^{'}$. The fuzzy sets is extended to an intuitionistic fuzzy set (IFS) A, given by

$$A = \{\langle x, \mu_A(x), \nu_A(x)\rangle | x \in X\} \quad (6)$$

where: $\mu_A : X \to [0,1]$ and $\nu_A : X \to [0,1]$ such that

$$0 \leq \mu_A(x) + \nu_A(x) \leq 1 \quad (7)$$

Intuitionistic fuzzy estimation of the inner area II

and $\mu_A(x)$, $\nu_A(x) \in [0,1]$ denote a *degree of membership* and a *degree of non-membership* of x to A, respectively.
Further we give an intuitionistic fuzzy estimation of the initial curve based on the inner and outer polygons produced by the algorithm. As described above, the iterative procedure starts with inputs - the simple continuous curve $\vec{r}(t)$, an initial sufficiently small grid-step l_0. The algorithm ends up at an iteration, say $j(\vec{r}, l_0) = j_0$, where we may assume that j_0 is the maximum of the numbers of iterations producing the inner and outer polygons at first approximation of the procedure. And therefore, the end grid step becomes $l_1 = \frac{l_0}{2^{j_0}}$.
The area enclosed by the inner and outer polygons for the given grid-step will be denoted by $\mathcal{A}^i(\vec{r}, \frac{l_0}{2^{j_0}})$ and $\mathcal{A}^o(\vec{r}, \frac{l_0}{2^{j_0}})$, respectively.
The areas $\mathcal{A}^i$ and $\mathcal{A}^o$ can be calculated by Pick's formula.

Intuitionistic fuzzy estimation of the inner area III

In the above notations let us define

$$\mu_{(\vec{r},l_0)}(0) = \frac{\mathcal{A}^i(\vec{r}, \frac{l_0}{2^{j_0}})}{\mathcal{S}_{j_0}} \text{ and } \nu_{(\vec{r},l_0)}(0) = \frac{\mathcal{S}_{j_0} - \mathcal{A}^o(\vec{r}, \frac{l_0}{2^{j_0}})}{\mathcal{S}_{j_0}},$$

$$\mu_{(\vec{r},l_0)}(1) = \frac{\mathcal{A}^i(\vec{r}, \frac{l_0}{2^{j_0+1}})}{\mathcal{S}_{j_0}} \text{ and } \nu_{(\vec{r},l_0)}(1) = \frac{\mathcal{S}_{j_0} - \mathcal{A}^o(\vec{r}, \frac{l_0}{2^{j_0+1}})}{\mathcal{S}_{j_0}}.$$

More generally, let us inductively define for any positive integer k,

$$\mu_{(\vec{r},l_0)}(k) = \frac{\mathcal{A}^i(\vec{r}, \frac{l_0}{2^{j_0+k}})}{\mathcal{S}_{j_0}} \text{ and } \nu_{(\vec{r},l_0)}(k) = \frac{\mathcal{S}_{j_0} - \mathcal{A}^o(\vec{r}, \frac{l_0}{2^{j_0+k}})}{\mathcal{S}_{j_0}}.$$

Intuitionistic fuzzy estimation of the inner area IV

Taking into consideration the last definition we may write down the degree of uncertainty for the k–th step as $\pi_{(\vec{r},l_0)}(k) = 1 - \mu_{(\vec{r},l_0)}(k) - \nu_{(\vec{r},l_0)}(k)$. Therefore, we have that

$$\pi_{(\vec{r},l_0)}(k) = \frac{\mathcal{A}^o(\vec{r}, \frac{l_0}{2^{j_0+k}}) - \mathcal{A}^i(\vec{r}, \frac{l_0}{2^{j_0+k}})}{\mathcal{S}_{j_0}},$$

which is exactly the intuitionistic fuzzy estimation of the degree of uncertainty for the corresponding grid-step $\frac{l_0}{2^{j_0+k}}$. It can be easily proved that for $k_1 < k_2$ we have that $\pi_{(\vec{r},l_0)}(k_1) > \pi_{(\vec{r},l_0)}(k_2)$, which means exactly that $\pi_{(\vec{r},l_0)}$ is a decreasing function on the set of positive integer numbers $\mathbb{N}$.

Therefore, we may suppose that we are given a small enough positive real number ε_0 based on the curve $\vec{r}$, which interior is supposed to be estimated. Through the described algorithm, we

Intuitionistic fuzzy estimation of the inner area V

are computing then iteratively upper and lower estimations (through outer and inner polygons) until a positive integer k has been reached for which $0 \leq \pi_{(\vec{r},l_0)}(k) \leq \varepsilon_0$. The k-th iteration provides then a satisfactory intuitionistic fuzzy estimation of the curve. This also means that $\mathcal{A}^o(\vec{r}, \frac{l_0}{2^{j_0+k}})$ and $\mathcal{A}^i(\vec{r}, \frac{l_0}{2^{j_0+k}})$ provide a corresponding satisfactory upper and lower estimations of the area surrounded by the curve.

An example I

As an example of a curve $\vec{r}$ consider the one we saw previously

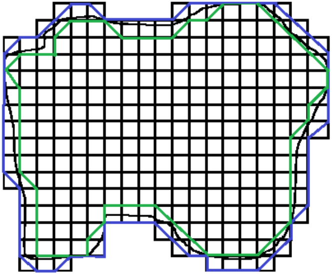

Figure: The whole picture of the inner and outer polygons with the end grid-step $l_1 = \frac{l_0}{2^{j_0}}$ of the first iteration. The inside area of the inner polygon is 205.5 while the area of the outer one is 252.5.

An example II

say after the end of the procedure started for first time with a grid-step l_0 and finishing with end grid-step $l_1 = \frac{l_0}{2^{j_0}}$), that

- $\mathcal{S}(\vec{r}, l_1) = \mathcal{S}_{j_0} = 264 \times l_1$,
- $\mathcal{A}^i(\vec{r}, l_1) = \left(178 + \frac{57}{2} - 1\right) \times l_1 = 205.5 \times l_1$,
- $\mathcal{A}^o(\vec{r}, l_1) = \left(223 + \frac{61}{2} - 1\right) \times l_1 = 252.5 \times l_1$.

Therefore, for the intuitionistic fuzzy estimations after the first application of the procedure, we have that

- $\mu_{(\vec{r}, g, l_0)}(1) = \frac{205.5}{264}$,
- $\nu_{(\vec{r}, g, l_0)}(1) = \frac{264 - 252.5}{264} = \frac{11.5}{264}$,
- $\pi_{(\vec{r}, g, l_0)}(1) = \frac{252.5 - 205.5}{264} = \frac{47}{264}$.

Conclusion

The described procedure and the intuitionistic fuzzy estimation provide an iterative numerical algorithm, which can be implemented in any procedural programming language.
The method described in this paper can be adequately applied to the estimation of the area of a forest fire spread. In future research, we will discuss a modification of the Pick's formula for 3D-figures.

Acknowledgments

This work has been partially supported by the Bulgarian Science Fund under Grant DN16/6/2017 "Integrated Approach for Modeling of the Forest Fires Spread".
https://forestfires.info/

Thank you for Your Attention!