Applied Mathematics and Computation 218 (2012) 9866–9874

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Finding the largest area rectangle of arbitrary orientation in a closed contour

Rubén Molano [a,*], Pablo G. Rodríguez [b], Andrés Caro [b], M. Luisa Durán [b]

[a] Department of Mathematics, Escuela Politécnica, University of Extremadura, Cáceres, Spain
[b] Department of Computer Science, Escuela Politécnica, University of Extremadura, Cáceres, Spain

ARTICLE INFO

Keywords:
Rectangle
Polygon
Region of Interest (ROI)

ABSTRACT

For many software applications, it is sometimes necessary to find the rectangle of largest area inscribed in a polygon, in any possible direction. Thus, given a closed contour *C*, we consider approximation algorithms for the problem of finding the largest area rectangle of arbitrary orientation that is fully contained in *C*. Furthermore, we compute the largest area rectangle of arbitrary orientation in a quasi-lattice polygon, which models the *C* contour. In this paper, we propose an approximation algorithm that solves this problem with an $O(n^3)$ computational cost, where n is the number of vertices of the polygon. There is no other algorithm having lower computational complexity regardless of any constraints. In addition, we have developed a web application that uses the proposed algorithm.

1. Introduction

Many Computer Vision algorithms focus on parts of an image, instead of processing the whole image. These Region of Interest (ROI) are usually sub-images with basic shapes [1], mainly rectangles. However, in many cases the regions are available as irregular polygons with many vertices and it is essential to compute the largest area rectangle contained in them. The algorithm developed and proposed in this paper derives from the need for specific practical applications developed in the field of food technology. The most representative muscle in a ripening meat piece is automatically segmented with active contours by magnetic resonance imaging (MRI), a non-destructive, non-invasive method [2,3]. The shape of this muscle is represented as a closed polygon. Then, computational textures features are obtained from these extracted muscles, whereas texture algorithms work over rectangular ROIs [4]. Therefore, the aim of this paper is to compute the largest of these ROIs, and thus have it be more representative for the study of product quality.

The problem of finding the largest area axis-aligned rectangle contained in a convex polygon was considered by Fischer and Höffgen [5]: given a convex polygon of n vertices (S), compute the rectangle $R \subset S$ with a maximum area whose sides are parallel to the x-axis and y-axis; their approach solved the problem in $O(\log^2 n)$ time. Later, in [6] this problem was solved in $O(\log n)$ time. The restriction of the problem for convex polygons was removed by Daniels et al. [7], computing the largest area axis-parallel rectangle in an n vertex general polygon in $O(n\log^2 n)$ time. Later, the authors Boland and Urrutia [8], showed how to solve the problem in a more efficient way, in $O(n \log n)$ time. Nonetheless none of the previous work solves the problem for plane figures that are not polygons. Furthermore, giving priority to the area of the rectangle, it could be larger if it is not an axis-aligned rectangle. So, recently, Knauer et al. [9] considered approximation algorithms for the problem of computing a rectangle with the largest area of arbitrary orientation in a convex polygon with n vertices and they proved that it can be computed in $O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon} \log n)$ time. In addition, for simple polygons with or without holes, they get the following

* Corresponding author.
E-mail addresses: rmolano@unex.es (R. Molano), pablogr@unex.es (P.G. Rodríguez), andresc@unex.es (A. Caro), mlduran@unex.es (M.L. Durán).

0096-3003/$ - see front matter
http://dx.doi.org/10.1016/j.amc.2012.03.063

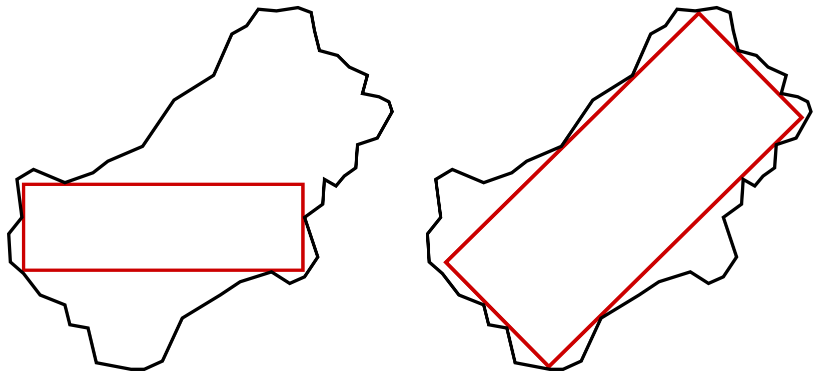


Fig. 1. Largest area rectangle with: (a) axis-aligned, and (b) arbitrary orientation.

running times: for simple polygons, $O(\frac{1}{\varepsilon}n^3 \log n)$ time and for polygons with holes, $O(\frac{1}{\varepsilon}n^3 \log^2 n)$ time. Our proposal has success for simple polygons without holes with a lower computational cost: $O(n^3)$. Fig. 1 shows two different rectangles inside the same polygon, one with axis-aligned and the other without this alignment. By removing this restriction, the problem becomes even more complex.

To our knowledge, there are no other approximation algorithms for finding the largest area rectangle of arbitrary orientation inscribed in a closed contour. Table 1 shows the computational costs and constrains of the main proposals. Our approach is successful without any constraints by achieving a better computational cost than Knauer et al. (see the two rows in bold at the bottom of the table). The following three sections will describe and detail the algorithm.

2. Largest area rectangle in a quasi-lattice polygon

Given the rectangle $[a,b] \times [c,d],\ a,b,c,d \in \mathbb{Q}$ a `regular partition` $P \times Q$ of order $r \times s$ are two ordered collections of $r+1$, $s+1$ equally spaced points which satisfy:

$$P = \{a = x_0 < x_1 < \ldots < x_r = b\},$$
$$Q = \{c = y_0 < y_1 < \ldots < y_s = d\}.$$

We denote $G_L = \{(x_i, y_j)\ :\ 0 \leqslant i \leqslant r,\ 0 \leqslant j \leqslant s\}$, the square grid composed of points of the partition $P \times Q$, where $L = |x_{i+1} - x_i| = |y_{j+1} - y_j|$ is the length of the side of each square formed by the square grid (also called `partition size`). We state that the partition $\dot{P} \times \dot{Q}$ is finer than the partition $P \times Q$, if it is verified that all points of $P \times Q$ belong to $\dot{P} \times \dot{Q}$. We denote $P \times Q \preceq \dot{P} \times \dot{Q}$.

Let S be a polygon whose vertices, in counter-clockwise order, belong to the square grid G_L for a regular partition $P \times Q$, on the rectangle circumscribed to S, and supposing that if $v_i = (x_u, y_k)$ and $v_j = (x_v, y_l)$ are two consecutive vertices of S, then:

$$(v - u, l - k) \in \{(z_1, z_2), (z_3, 0), (0, z_4) : z_i \in \mathbb{Z}, |z_1| = |z_2|\}.$$

That is, we assume that each pair of vertices (v_i, v_j) of the polygon are only connected in eight possible directions, $0^\circ, 45^\circ, 90^\circ, \ldots, 315^\circ$. A polygon S, defined in this way, is said to be a `quasi-lattice polygon` (Fig. 2).

We denote ∂S as the family consisting of boundary nodes of S, and its complementary in $S, \imath S$, the interior points, i.e. $S = \partial S \cup \imath S$. By Pick's theorem [10],

$$A(S) = \left(\#(\imath S) + \frac{\#(\partial S)}{2} - 1\right) \cdot L^2,$$

where $A(S)$ denotes the area of quasi-lattice polygon S and # represents the cardinality of the set. Similarly, we denote V as the family consisting of vertices of ∂S, and its complementary in ∂S, we denote $\imath\partial S$, i.e. $\partial S = V \cup \imath\partial S$. Then we decompose quasi-lattice polygon S as follows:

$$S = V \cup \imath\partial S \cup \imath S = \{p_1, p_2, \ldots, p_{n+m+o}\},$$

with $\#(V) = n,\ \#(\imath\partial S) = m,\ \#(\imath S) = o$ and $\#(S) = N = n + m + o \simeq kn,\ k \in \mathbb{N}$.

Thus, for Fig. 2: $V,\ \imath\partial S$ and $\imath S$ are represented by squares ($\square$), white circles ($\circ$) and black circles ($\bullet$), respectively. Therefore:

$$V = \{(x_2, y_4), (x_5, y_1), (x_4, y_0), (x_3, y_1), (x_2, y_0), (x_0, y_2)\},$$
$$\imath\partial S = \{(x_3, y_3), (x_4, y_2), (x_1, y_1), (x_1, y_3)\},$$
$$\imath S = \{(x_2, y_3), (x_1, y_2), (x_2, y_2), (x_3, y_2), (x_2, y_1), (x_4, y_1)\}.$$

2.1. The radial algorithm

It allows us to determine the position of a given point (p) with respect to a polygon (S). The radial algorithm [11] consists of traversing all edges of S in an orderly way, calculating the angle formed by its vertices, $V = \{v_1, \ldots, v_n\}$, and straight lines connecting them to the point, considering the sign of the orientation for the three points taken, $\{p, v_i, v_{i+1}\}$. Adding all the angles together with its sign, two solutions are possible, 0 and $\pm 2\pi$, which determine the position of the point in the polygon, 0 if the point is outside and $\pm 2\pi$ if inside. The function *radial (p: point; S: polygon)* returns `true` for the first case and `false` for the second. This can be solved in $O(1)$ time.

2.2. Vector matrix of S

We construct an upper triangular matrix U of dimension $N = n + m + o$ of vectors to determine which segments, when joining two points of S, remain within the polygon. These segments will form the sides of the largest rectangle, and therefore, we need certain conditions that the rectangle shall be contained within the polygon. Thus, we use the functions *radial (p: point; S: polygon)*, defined above, and *Intersection (A, B: point; S: polygon)*, computed in $O(1)$ time, which allow us to determine when the intersection between the segment $\overline{AB}$ and S is valid, i.e. the segment does not cross with any edge of the polygon. In this case, *Intersection (A, B: point; S: polygon)* returns `true`. To reduce the number of operations in the final algorithm (Section 2.3), we do not calculate the opposite vector to one given, since both have the same position relative to S. We also define the function of constant running time *inside (U: matrix; i, j: integer)* to check if vector $\overrightarrow{u_{ij}}$ is inside the polygon, where $\overrightarrow{u_{ij}} = \overrightarrow{p_i p_j} = p_j - p_i$ with $p_i, p_j \in S$.

The U matrix is as follows:

$$U = \begin{pmatrix} \vec{0} & \overrightarrow{u_{12}} & \overrightarrow{u_{13}} & \ldots & \overrightarrow{u_{1,N}} \\ \vec{0} & \vec{0} & \overrightarrow{u_{23}} & \ldots & \overrightarrow{u_{2,N}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vec{0} & \vec{0} & \vec{0} & \ldots & \overrightarrow{u_{N-1,N}} \\ \vec{0} & \vec{0} & \vec{0} & \ldots & \vec{0} \end{pmatrix}.$$

Algorithm 1 allows us to calculate matrix U in $O(n^2)$ time.

Algorithm 1: Function Compute_U (S: polygon) return U: matrix

Input: $S = V \cup \imath\partial S \cup \imath S$, $\#(S) = N = n + m + o \simeq kn,\ k \in \mathbb{N}$
Output: Matrix U of vectors $\overrightarrow{p_i p_j} = p_j - p_i$
for $i \leftarrow 1$ to N **do**
  for $j \leftarrow 1$ to N **do**
    //`Upper triangular: avoid calculations.`
    if $i \geqslant j$ **then**
      $U(i,j) \leftarrow 0$
    else
      if not *Intersection*(p_i, p_j, S) **then**
        $U(i,j) \leftarrow 0$
      else
        // $\overline{p_i p_j}$ `is inside or outside of` S. `We choose a point` p `to know the actual position.`
        $p \leftarrow (p_i + p_j)$ div 2;
        if *radial*(p, S) **then**
          // `Segment` $\overline{p_i p_j}$ `is not within` S.
          $U(i,j) \leftarrow 0$
        else
          // Update with the vector $\overrightarrow{p_i p_j} = p_j - p_i$
          $U(i,j) \leftarrow p_j - p_i$

2.3. Main algorithm

We use Algorithm 2 to calculate the coordinates of the largest area rectangle (or rectangles) contained within quasi-lattice polygon S, taking as vertices the points of the set $S = V \cup \imath\partial S \cup \imath S$, for a fixed partition size L.

Table 1
Computational cost.

Author	Year	Polygon	Orientation	Computational cost
Fischer and Höffgen	1994	Convex	Axis-aligned	$O(\log^2 n)$
Alt et al.	1995	Convex	Axis-aligned	$O(\log n)$
Daniels et al.	1997	Arbitrary	Axis-aligned	$O(n\log^2 n)$
Boland and Urrutia	2001	Arbitrary	Axis-aligned	$O(n \log n)$
Knauer et al.	2010	Convex	Arbitrary	$O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\log n)$
Knauer et al.	**2010**	**Arbitrary**	**Arbitrary**	$O(\frac{1}{\varepsilon}n^3 \log n)$
Molano et al.	**2010**	**Arbitrary**	**Arbitrary**	$O(n^3)$

To make the algorithm we must consider two conditions:

- For each pair of points (p_i, p_j), $U(i,j) \neq 0$, since the rectangle should have all sides in polygon S.
- As a rectangle is a parallelogram having four equal angles, we need to proceed to check if the vectors are parallel two-two and perpendicular. We build function of constant running time *perpendicular (u, v: vector)*, which returns `true` if the two vectors are perpendicular and `false` otherwise.

Algorithm 2 runs through all points of S, saving the coordinates of the rectangles found in case of obtaining a solution with an equal or better area and considering the two conditions above. The complexity order is $O(n^3)$ where n is the number of vertices the quasi-lattice polygon S.

Algorithm 2: Procedure Compute_largest_rectangles (in S: polygon; out Rectangles: set <rectangle>)

```
Input: S = V ∪ ı∂S ∪ ıS , #(S) = N = n + m + o ≃ kn,  k ∈ ℕ
Output: Rectangles: set < rectangle>, set of largest area rectangles
Max_area ← 0;
for i ← 1 to N − 3 do
  for j ← i + 1 to N − 2 do
    if inside(i,j) then
      // We seek the remaining points, p_k and p_s.
      for k ← j + 1 to N − 1 do
        if inside(i,k) and perpendicular(U(i,j), U(i,k)) then
          // to seek last point p_s such that p_i p_j (vector) = p_k p_s (vector).
          p_s ← p_j − p_i + p_k;
          if inside(k,s) and inside(j,s) then
            area ← |U(i,j)| · |U(i,k)|;
            // Update solution.
            if area > Max_area then
              Max_area ← area;
              Rectangles.clear ();
              R ← (p_i, p_j, p_k, p_s);
              Rectangles.insert (R);
            else if area = Max_area then
              R ← (p_i, p_j, p_k, p_s);
              Rectangles.insert (R);
```

3. Approximation for the largest area rectangle in a closed contour

Once we have shown how to calculate the largest area rectangle within a quasi-lattice polygon we find that a simple closed contour without holes can be achieved by means of the limit of the area of rectangles inscribed in it.

First, we applied the proposed algorithm by Freeman and Shapira [12] and we computed the rectangle of minimum area R_{min} that encloses a closed contour C. We also assume that the rectangle rotates clockwise with sides parallel to x and y axes (Fig. 3).

Definition 3.1. Let $P \times Q$ be a regular partition of the rectangle R_{min} with partition size L.

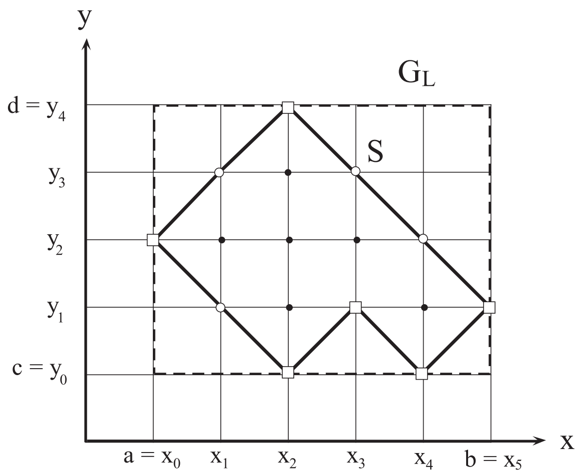


Fig. 2. Quasi-lattice polygon S on a regular partition of order 5×4 with partition size L.

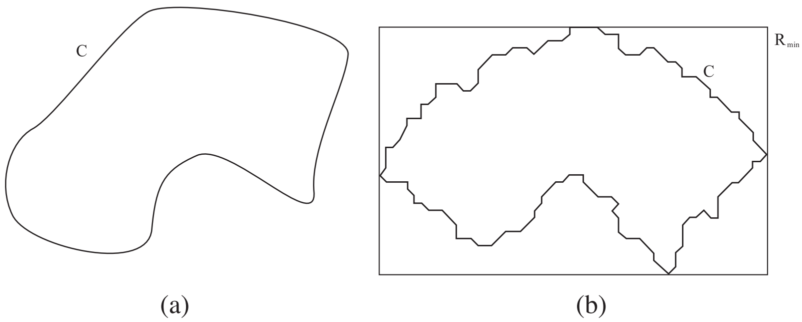


Fig. 3. A closed contour C (a), in chain code form and encased in the minimum-area rectangle R_{min} with sides parallel to x and y axes (b).

We define `Lower area`, $\underline{A}(C, P \times Q)$, as the largest area quasi-lattice polygon $\underline{S}$ contained in C and built by points of G_L (Fig. 4(a)). According to Pick's theorem [10]:

$$\underline{A}(C, P \times Q) = \left(\#(\imath\underline{S}) + \frac{\#(\partial\underline{S})}{2} - 1 \right) \cdot L^2,$$

where $\#(\imath\underline{S})$ and $\#(\partial\underline{S})$ are respectively the number of interior points and the boundary of $\underline{S}$ for the square grid. Furthermore, we denote $\underline{A_R}(C, P \times Q)$ as the largest area rectangle $\underline{R}$ contained in $\underline{S}$ and computed by the Algorithm 2.

Similarly, we define `Upper area`, $\overline{A}(C, P \times Q)$, as the smallest area quasi-lattice polygon $\overline{S}$ that contains C and is built by points of G_L (Fig. 4(b)) and $\overline{A_R}(C, P \times Q)$ as the largest area rectangle $\overline{R}$ contained in $\overline{S}$ and computed by the Algorithm 2.

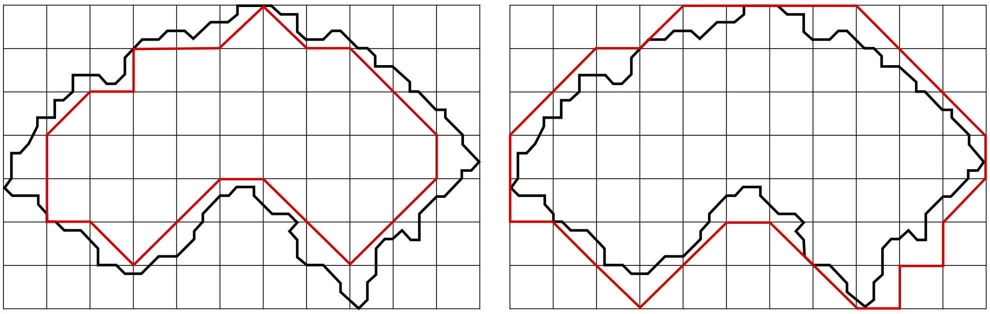
Lemma 3.2. *Let C be a simple closed contour without holes and $P \times Q$, $\dot{P} \times \dot{Q}$ regular partitions with $P \times Q \preceq \dot{P} \times \dot{Q}$. Then,*

$$\underline{A}(C, P \times Q) \leqslant \underline{A}(C, \dot{P} \times \dot{Q}) \leqslant \overline{A}(C, \dot{P} \times \dot{Q}) \leqslant \overline{A}(C, P \times Q).$$

Proof. As $\dot{P} \times \dot{Q}$ is finer than $P \times Q$, $G_L \subset G_{\dot{L}}$, then $\underline{S} \subset \underline{\dot{S}}$ and $\overline{S} \supset \overline{\dot{S}}$, where $\underline{S}$ and $\underline{\dot{S}}$ are the largest quasi-lattice polygon contained in C for the regular partitions $P \times Q$ and $\dot{P} \times \dot{Q}$, respectively, and $\overline{S}$ and $\overline{\dot{S}}$ the smallest quasi-lattice polygon that contain C for the same previous partitions. Therefore, $\underline{A}(C, P \times Q) \leqslant \underline{A}(C, \dot{P} \times \dot{Q})$ and $\overline{A}(C, \dot{P} \times \dot{Q}) \leqslant \overline{A}(C, P \times Q)$. The last inequality $\underline{A}(C, \dot{P} \times \dot{Q}) \leqslant \overline{A}(C, \dot{P} \times \dot{Q})$ is by definition, since $\underline{\dot{S}} \subset C$ and $C \subset \overline{\dot{S}}$. Thus, $\#(\imath\underline{\dot{S}}) \leqslant \#(\imath\overline{\dot{S}})$ and $\#(\partial\underline{\dot{S}}) \leqslant \#(\partial\overline{\dot{S}})$ and therefore $\#(\imath\underline{\dot{S}}) + \frac{\#(\partial\underline{\dot{S}})}{2} - 1 \leqslant \#(\imath\overline{\dot{S}}) + \frac{\#(\partial\overline{\dot{S}})}{2} - 1$. Then, $\underline{A}(C, \dot{P} \times \dot{Q}) \leqslant \overline{A}(C, \dot{P} \times \dot{Q})$. $\square$

Theorem 3.3. *Let C be a simple closed closed contour without holes and $P_1 \times Q_1$, $P_2 \times P_2$ regular partitions. Then,*

$$\underline{A}(C, P_1 \times Q_1) \leqslant \overline{A}(C, P_2 \times Q_2).$$



(a) $\underline{A}(C, P \times Q)$ (b) $\overline{A}(C, P \times Q)$

Fig. 4. Lower and upper area for the regular partition $P \times Q$ with partition size L.

Proof. By Lemma 3.2, there exists regular partition $P \times Q$ as finer than $P_1 \times Q_1$ and $P_2 \times Q_2$ at once. Then, $\underline{A}(C, P_1 \times Q_1) \leqslant \underline{A}(C, P \times Q) \leqslant \overline{A}(C, P \times Q) \leqslant \overline{A}(C, P_2 \times Q_2)$. $\square$

The previous theorem shows that, $\sup\{\underline{A}(C, P \times Q) : P \times Q \textit{ regular}\} \leqslant \overline{A}(C, \dot{P} \times \dot{Q})$ for all $\dot{P} \times \dot{Q}$ regular partition, and as $\inf\{\overline{A}(C, P \times Q) : P \times Q \textit{ regular}\} \leqslant \overline{A}(C, \dot{P} \times \dot{Q})$ for all $\dot{P} \times \dot{Q}$ regular partition, $\sup\{\underline{A}(C, P \times Q) : P \times Q \textit{ regular}\} \leqslant \inf\{\overline{A}(C, P \times Q) : P \times Q \textit{ regular}\}$. Therefore, it is possible that:

$$\sup\{\underline{A}(C, P \times Q) : P \times Q \quad \textit{regular}\} = \inf\{\overline{A}(C, P \times Q) : P \times Q \textit{ regular}\}. \tag{1}$$

The common number is called `Area of` C and is denoted by $A(D)$.

Note 3.4. *The closed contours no simple or with holes not verify the equality (1).*

The main result of this paper is the Theorem 3.5 which tells us that if we make successively finer partitions, (a) the area of a closed contour can be calculated by the inscribed limit area within the quasi-lattice polygon and (b) the largest area rectangle contained in a closed contour can be calculated by the inscribed limit area within the rectangles.

Theorem 3.5. *Let C be a closed contour that verifies:*

$$\sup\{\underline{A}(C, P \times Q) : P \times Q \textit{ regular}\} = \inf\{\overline{A}(C, P \times Q) : P \times Q \textit{ regular}\}.$$

Then, there exists a sequence of regular partitions $\{P_n \times Q_n\}_{n \in \mathbb{N}}$ with $P_i \times Q_i \preceq P_{i+1} \times Q_{i+1}$ for all i such that:

(a) $\lim_{n \to \infty}(\underline{A}(C, P_n \times Q_n)) = A(C)$,
(b) $\lim_{n \to \infty}(\underline{A_R}(C, P_n \times Q_n)) = A_R(C)$,

where $A(C)$ is the area of a closed contour C and $A_R(C)$ the largest area rectangle contained in it.

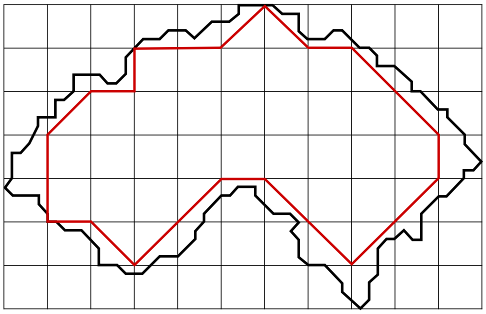
Proof. If $\sup\{\underline{A}(C, P \times Q) : P \times Q \text{ regular}\} = \inf\{\overline{A}(C, P \times Q) : P \times Q \text{ regular}\}$, we have that for all $\varepsilon > 0$, there exist regular partitions $\dot{P} \times \dot{Q}, \ddot{P} \times \ddot{Q}$ such that $|\overline{A}(C, \ddot{P} \times \ddot{Q}) - \underline{A}(C, \dot{P} \times \dot{Q})| < \varepsilon$. We consider a regular partition $P \times Q$ as finer than $\dot{P} \times \dot{Q}$ and $\ddot{P} \times \ddot{Q}$ at once. By Lemma 3.2, $\overline{A}(C, P \times Q) \leqslant \overline{A}(C, \ddot{P} \times \ddot{Q})$ y $\underline{A}(C, P \times Q) \geqslant \underline{A}(C, \dot{P} \times \dot{Q})$ and therefore $|\overline{A}(C, P \times Q) - \underline{A}(C, P \times Q)| \leqslant |\overline{A}(C, \ddot{P} \times \ddot{Q}) - \underline{A}(C, \dot{P} \times \dot{Q})| < \varepsilon$, i.e. for all $\varepsilon > 0$, there exists the regular partition $P \times Q$ such that $|\overline{A}(C, P \times Q) - \underline{A}(C, P \times Q)| < \varepsilon$. Then, there exists a sequence of regular partitions $\{P_n \times Q_n\}_{n \in \mathbb{N}}$ with $P_i \times Q_i \preceq P_{i+1} \times Q_{i+1}$ for all i such that $\lim_{n \to \infty}(\overline{A}(C, P_n \times Q_n) - \underline{A}(C, P_n \times Q_n)) = 0$ and so, $\lim_{n \to \infty}(\underline{A}(C, P_n \times Q_n)) = A(C)$. This proves (a).

We prove (b). Let a regular partition $P \times Q$ such that $|\overline{A}(C, P \times Q) - \underline{A}(C, P \times Q)| < \varepsilon$. As $\underline{A_R}(C, P \times Q) \leqslant \underline{A}(C, P \times Q)$ and $\overline{A_R}(C, P \times Q) \leqslant \overline{A}(C, P \times Q)$ for any regular partition, $|\overline{A_R}(C, P \times Q) - \underline{A_R}(C, P \times Q)| < \varepsilon$. Then, there exists a sequence of regular partitions $\{P_n \times Q_n\}_{n \in \mathbb{N}}$ con $P_i \times Q_i \preceq P_{i+1} \times Q_{i+1}$ for all i such that $\lim_{n \to \infty}(\overline{A_R}(C, P_n \times Q_n) - \underline{A_R}(C, P_n \times Q_n)) = 0$ and so, $\lim_{n \to \infty}(\underline{A_R}(C, P_n \times Q_n)) = A_R(C)$. $\square$

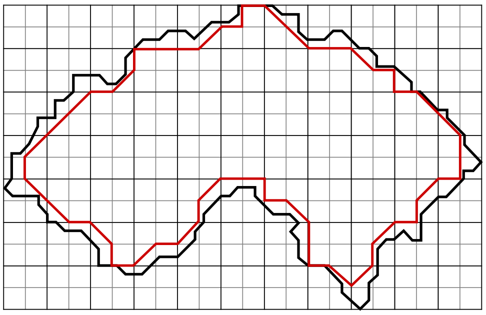
Example 3.6. In Fig. 5 we show geometrically that the area of a closed contour can be calculated by the inscribed limit area within the quasi-lattice polygon, constructing finer partitions with $L_{i+1} = L_i/2$ if $P_i \times Q_i \preceq P_{i+1} \times Q_{i+1}$ for all i.

4. Practical application

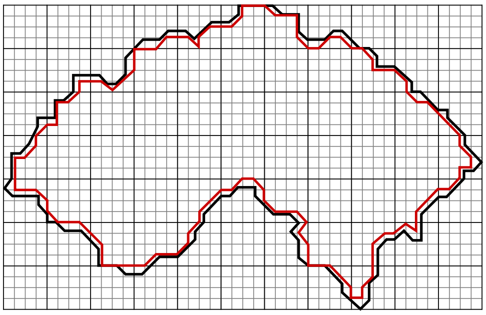
Mathematically, it has been proved by Theorem 3.5 that it is possible to compute an approximation algorithm for the problem of finding the largest area rectangle contained in a closed contour. However, for a practical application, it is not useful to successively take on a smaller partition size so that $L_{i+1} = L_i/2$ if $P_i \times Q_i \preceq P_{i+1} \times Q_{i+1}$ for all i, since there is a minimum



(a) $\underline{A}(C, P_1 \times Q_1)$ with partition size L_1



(b) $\underline{A}(C, P_2 \times Q_2)$ with $L_2 = L_1/2$



(c) $\underline{A}(C, P_3 \times Q_3)$ with $L_3 = L_2/2 = L_1/4$

Fig. 5. Lower area for different regular partitions, $P_1 \times Q_1 \preceq P_2 \times Q_2 \preceq P_3 \times Q_3$.

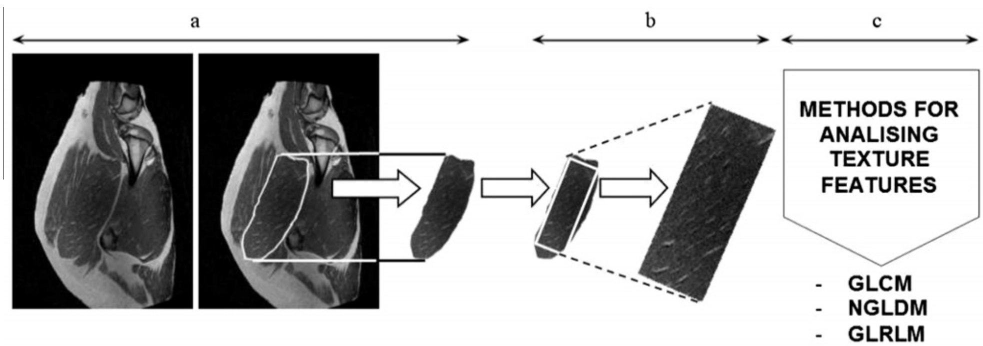


Fig. 6. Processing modules within the computer-aided MRI analysis: (a) muscle detection; (b) maximum ROI selection; (c) ROI analysis by using three different methods of texture features.

image unit, called pixel. Then, to compute the coordinates of the rectangles with the largest area inscribed in the closed contour C, it would be convenient to proceed with the following steps:

1. Apply the proposed algorithm by Freeman and Shapira [12] and to compute R_{min}.
2. Choose a partition size $L = 1$ pixel for a regular partition $P \times Q$ and to compute the quasi-lattice polygon $\underline{S}$.
3. Apply the Algorithm 2 to $\underline{S}$.

We have developed a real case application in http://gim.unex.es containing three modules (Fig. 6). The initial module aims to detect the muscle by using active contours according to the method described in [2]. The second module consists in the selection procedure for the ROI on each image; this selection draws up the maximum rectangular area in the muscle. Our paper focuses on the second step to follow in the feature extraction process. The sample needs to be sufficiently representative. In Fig. 7, the difference between an axis-parallel ROI (on the left) and the largest rectangle inside the contour of the

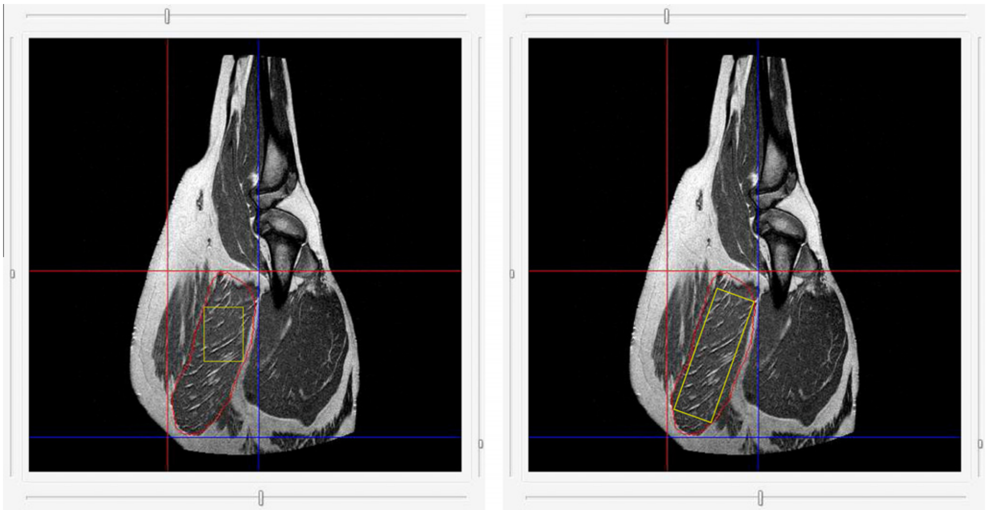


Fig. 7. The food technology application.

muscle (on the right) can be perceived. The method returns the largest area rectangle ROI covered by the muscle. Therefore the calculated ROIs are reasonably large dimensions in most cases, allowing for a greater surface cover. The third and last module includes the analysis of the ROIs by applying the three most common methods in computational texture analysis, which require the use of rectangular images. All three methods integrate matrices based on second order statistics [4] with which the most common features are obtained. The first one, Grey Level Coocurrence Matrix (GLCM) [13], is built with information of the complete ROI. Second, the so-called Neighbouring Grey Level Dependence Matrix (NGLDM) [14] gathers information from square neighborhoods inside the ROI, and third, the Grey Level Run Length Matrix (GLRLM) [15] only accounts for information about lineal segments of the ROI. This practical application has been successfully used in different previous research published in journals of varying impact factors [16,17].

5. Conclusions

In this paper, an approximation algorithm for the problem of finding the largest area rectangle of arbitrary orientation in a closed contour has been considered. Therefore, a solution in our research about texture analysis algorithms has been achieved, computing the largest ROI, since the most representative rectangle inside the muscle is needed and reached. In addition, we have developed a web application designed for that purpose.

Acknowledgments

The authors wish to acknowledge the funding received for this research from both the Junta de Extremadura (Regional Government Board – Research Project PDT08A021) and from the Spanish Government (National Research Plan) and the European Union (FEDER funds) by means of the grant reference TIN2008-03063.

References

[1] H. Pan, J. Li, W. Zhang, Incorporating domain knowledge into medical image clustering, Appl. Math. Comput. 185 (2) (2007) 844–856.
[2] T. Antequera, A. Caro, P.G. Rodríguez, T. Pérez-Palacios, Monitoring the ripening process of Iberian Ham by computer vision on magnetic resonance imaging, Meat Sci. 76 (3) (2007) 561–567.
[3] K.M. Iftekharuddin, J. Zheng, M.A. Islam, R.J. Ogg, Fractal-based brain tumor detection in multimodal MRI, Appl. Math. Comput. 207 (1) (2009) 23–41.
[4] E. Cernadas, P. Carrión, P.G. Rodríguez, E. Muriel, T. Antequera, Analyzing magnetic resonance images of Iberian pork loin to predict its sensorial characteristics, Comput. Vis. Image Und. 98 (2) (2005) 340–360.
[5] P. Fischer, K.U. Höffgen, Computing a maximum axis-aligned rectangle in a convex polygon, Inform. Process. Lett. 51 (4) (1994) 189–193.
[6] H. Alt, D. Hsu, J. Snoeyink, Computing the largest inscribed isothetic rectangle, Proceedings of 1995 Canadian Conference on Computational Geometry (CCCG) (1995) 67–72.
[7] K. Daniels, V. Milenkovic, D. Roth, Finding the largest area axis-parallel rectangle in a polygon, Comput. Geo. Theor. Appl. 7 (1–2) (1997) 125–148.
[8] R.P. Boland, J. Urrutia, Finding the largest axis-aligned rectangle in a polygon in O(nlog n) time, Proc. 13th Canad. Conf. Comput. Geom, 2001, 41–44.
[9] C. Knauer, L. Schlipf, J.M. Schmidt, H.R. Tiwary, Largest inscribed rectangles in convex polygons, Proceedings of the 26th European Workshop on Computational Geometry (EuroCG'10), Dortmund, Germany, 2010.
[10] M. Bruckheimer, A. Arcavi, Farey series and Pick's area theorem, Math. Intell. 17 (4) (1995) 64–67.
[11] M. Martín-Arriarán, El algoritmo "radial": otra solución para el problema del "Punto en Polígono", Mapping 62 (2000) 20–22.
[12] H. Freeman, R. Shapira, Determining the minimum-area encasing rectangle for an arbitrary closed curve, Commun. ACM 18 (7) (1975) 409–413.
[13] R.M. Haralick, L.G. Shapiro, Computer and Robot Vision, Addison-Wesley, Reading, 1993.
[14] C. Sun, G. Wee, Neighboring gray level dependence matrix, Comput. Vis. Graph. Image Process 23 (1982) 341–352.

[15] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis and Machine Vision, second ed., PWS Publishing, Pacific Grove, 1999.
[16] T. Pérez-Palacios, T. Antequera, M.L. Durán, A. Caro, P.G. Rodríguez, R. Palacios, MRI-based analysis of feeding background effect on fresh Iberian Ham, Food Chem. 126 (2011) 1366–1372.
[17] T. Pérez-Palacios, T. Antequera, R. Molano, P.G. Rodríguez, R. Palacios, Sensory traits prediction in dry-cured hams from fresh product via MRI and lipid composition, J. Food Eng. 101 (2) (2010) 152–157.