A Qualitative Shape Description Scheme for Generating New Manufactured Shapes

Lledó Museros[1], Zoe Falomir[1], Luis Gonzalez-Abril[2], and Francisco Velasco[2]
[1]Universitat Jaume I and [2]Universidad de Sevilla

Abstract

A qualitative shape description scheme is presented in this paper. This scheme has been defined in order to have a formal theory that allows the construction of new shapes from a set of given shapes using a juxtaposition operation. The juxtaposed shapes may be regular or non-regular polygons, and the edges considered during the juxtaposition may be of similar or different length. Furthermore, the scheme presented is a pragmatic one since it has been developed considering its application to robotics and industrial systems.

1 Introduction

It is difficult to obtain a precise description of shape. In the *Oxford English Dictionary*, the term *shape* is described as follows: "external form o contour; that quality of a material object (or geometrical figure) which depends on constant relations of position and proportionate distance among all the points composing its outline or its external surface". This meaning of shape emphasizes the fact that human beings are aware of *shapes* through outlines and surface objects, both of which can be visually perceived. As Ghosh and Deguchi [2008] stated, in more technical terms, the *shape* of an object is "information about the geometrical aspects of the surface of the object". Therefore, *shape description* involves specifying the information through a scheme or a system.

It is well known that shape description is now a major field of study in the disciplines of computer vision, robotics, pattern analysis and recognition, computer graphics, image processing and computer-aided design and manufacture.

Although there are studies for describing the shape of an object, shape description is still challenging. This is due to the fact that it is particularly difficult to describe a shape computationally, because when a three-dimensional object is projected from the real world onto a two-dimensional image plane, one dimension of the object information is lost. Therefore, the shape extracted from a digital image of the object only represents the projected object partially. Furthermore, digital images are often corrupted with noise, defects, arbitrary distortion and occlusion.

Because of the numerical properties of digital images, most of the image processing in computer vision has been carried out by applying mathematical models and other quantitative techniques to describe and identify the shape of the objects contained in an image [Mehtre, Kankanhalli and Lee, 1997; Rucklidge, 1997; Peura and Iivarinen, 1997; Belongie, Malik, and Puzicha, 2001; Zhang and Lu, 2002; Lu and Sajjanhar, 1999; Morse, 1994].

Most of these quantitative approaches have succeeded in the task for which they were designed. However, the problem of qualitative shape description and identification is studied in this paper since human beings describe and use visual knowledge about shapes qualitatively.

The definition and use of a theory for qualitative description of shape is important in computer vision, which up to now uses quantitative methods with a high computational cost. The use of a qualitative theory for shape description and recognition will increase the efficiency in vision recognition because the recognition of a shape or an environment will be carried out by looking for only the salient characteristics and not analysing each pixel of the image.

Museros and Escrig [2004a] presented a qualitative theory able to describe different types of shapes. It can describe regular and non-regular closed shapes, of which the boundaries can be completely straight, or curved, or a mixture of both. These shapes may also contain holes whose shape and location can also be described by our approach. To be precise, it is based on qualitative representations of: angles, types of curvature, length of the edges, convexities, and concavities. This theory has been applied to different domains, such as the industrial domain [Museros et al., 2011] or the mobile robot navigation domain [Museros and Escrig 2004b], and it is also being extended by Falomir et al. [2008, 2011] in the qualitative image description domain. These applications provide evidence for the effectiveness of using qualitative information to describe and identify shapes.

However, there are some new applications where it is needed to create new shapes from two other given shapes, and as a consequence there exists the necessity of extending the theory developed by Museros and Escrig [2004a] in order to be able to define a new shape by providing several shapes and using operations among shapes (such as addition, juxtaposition, intersection and difference of shapes).

For instance, in order to describe the geometry of the environment where a robot navigates sometimes it is needed to describe a new shape by juxtaposing the shapes of the environment already described. Another example arises in the industrial domain, as in the case of *trencadís* ceramic mosaics (mosaics hand-made of broken tiles), where it is necessary to create a design by juxtaposing broken tiles with a priori unknown shapes. Therefore, it is needed to create a *shape description scheme*, that is, a notational system for expressing the shape of objects, a way of writing the shape information symbolically, just as the notation used to express music or electronic circuitry.

Although in the literature we can find shape descriptions schemes which have been used with much success to study and represent past and contemporary architectural and other designs [Ahmad and Chase, 2004; Andaroodi et al., 2006; Chase, 1989; Knight, 1994; Lebigre, 2001; Stiny, 1980a], and they have been used also into the education and practice field [Halatsch et al., 2008; Stiny, 1980b; Wang and Duarte, 1998], in this paper a *qualitative shape description scheme* for the theory described by Museros [2004a] is presented. The scheme presented is able to construct new shapes from a set of given shapes using the juxtaposition operation between shapes.

The rest of this paper is organized as follows. The next section presents an overview of part of the qualitative shape description theory presented by Museros [2004a], specifically the part of the theory which is able to describe regular and non-regular polygonal objects. Then, the qualitative shape description scheme able to juxtapose two shapes of this theory is described. Finally, our conclusions and future work are drawn.

2 Qualitative Shape Description Theory (QSDT) for describing polygons

The shape description theory presented uses some reference points, which are understood as the points that completely specify the boundary. For polygonal boundaries, the vertices have been chosen.

The qualitative description of a reference point, named *j*, is determined using the previous reference point, named *i*, and following reference point, named *k*. The order of the reference points is given by the natural cyclic order of the vertices of closed objects. It is only necessary to determine the sense in which each reference point is visited or described, which should be the same for the description of all the objects. The sense chosen is the counterclockwise. The description of each reference point is given by a triple, which is:

<Aj,Cj,Lj>, where *Aj* and *Cj* are the angle and the type of convexity for the reference point *j*, respectively, and *Lj* means the relative length of the edges associated to reference point *j* (edge formed by vertices *i* and *j* versus edge formed by vertices *j* and *k*), where:

Aj ∈ *{right, acute, obtuse}*;
Cj ∈ *{convex, concave} and*
Lj ∈ *{smaller, equal, bigger}*.

Color is also stored in the description for later comparisons when matching objects by using the RGB coordinates.

Therefore, the complete description of a shape of a 2D object is defined by the following tuple:

[[Color, [A1, C1, L1] ... [An, Cn, Ln]],

where *n* is the number of reference points of the container and *j* is the number of reference points of the hole. Each vector, *[Ai, Ci, Li],* represents a qualitative description node.

Figure 1 shows an example of a shape and its qualitative shape description, formally named *QualShape(S),* being *S* the reference to the object described.

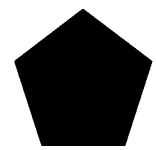


Figure 1. Example of a black shape.

QualShape(S)= [[0,0,0], [obtuse, convex, equal], [obtuse, convex, equal], [obtuse, convex, equal], [obtuse, convex, equal], [obtuse, convex, equal]].

3 Juxtaposition between shapes

There are two kinds of shape description schemes: a *pure* scheme or a *pragmatic* one. A pure scheme deals primarily with the question of "what" rather than "how", and it is very rigorous, and therefore its application to practical problems is incidental and of lesser interest. In contrast, a pragmatic scheme is a utilitarian scheme, devised as a response to the needs of a particular problem, its application in the real world being of primary concern. In a pure scheme the study of a subject is regarded as a self-sufficient exercise in pure thought, while in a pragmatic one the method is devised as a response to the needs of a particular problem. An attitude of the truly pragmatic approach is that "a mathematical object does not exist unless it can be constructed". Therefore, in a pragmatic approach it is not sufficient to specify "there exist", it is also necessary to "find/construct".

In this paper, the qualitative shape description scheme developed is a pragmatic one, and therefore it has to be defined by a 4-tuple as follows:

*(P, *, C, A)* where:

1. *P* is a set of *shapes*,
2. * is a set of functions/operations, called *shape operators,*
3. *C* is a set of *production rules*, which specifies how the shape operators are to be used to construct new shapes from the already existing shapes,
4. And *A* is a set of *explicit axioms*, which specify conditions that each constructed shape must satisfy. In a sense, *A* is a set of *constraints* or *restrictions*. In a shape

description scheme, the set *A* may or may not be present.

To the best of our knowledge, this is the first pragmatic qualitative shape description scheme able to juxtapose two shapes (Figure 2).

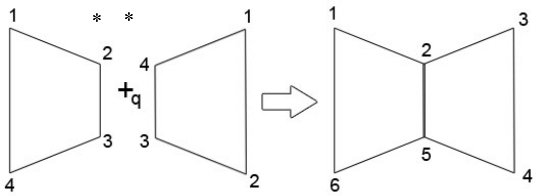


Figure 2. Graphic example of the juxtaposition operation between two shapes. The symbol "*" in figures specifies the edges that should be considered during the operation.

Next, the 4-tuple is defined for the qualitative shape description scheme presented in previous section.

3.1 The set *P*

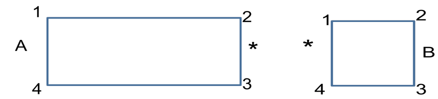
The set *P* is the set of the regular and non-regular polygonal closed shapes described by the QSDT presented in section 2, plus a special shape, called *empty shape (*represented by *∅s)*, which represents that there is no shape to juxtapose. The shapes to be juxtaposed may contain convex and concave segments. Moreover, the polygonal shapes to be juxtaposed are shapes of manufactured objects or potentially manufactured objects (well-designed objects, in the sense that they are the designs of a man-made object intended for production), and they are not shapes of all possible natural objects (with sharper angles and very curvy sides, which means objects that cannot be manufactured). This classification is necessary because the techniques that are well suited for description of the shapes of manufactured objects turn out to be inadequate for the shapes of natural objects.

3.2 The shape operator *

The set * is a set composed of a unique operator, which is the qualitative juxtaposition operator, named $+_q$. In order to juxtapose two shapes it is necessary to indicate the related edges in $+_q$. For instance, in order to juxtapose the shape in Figure 3a) with the shape in Figure 3b) it is necessary to indicate that the juxtaposition has to be done considering the edge going from vertex 2 to 3 in Figure 3b) and the edge going from vertex 4 to 1 in Figure 3c). Therefore the following notation for the qualitative juxtaposing operation has been defined as:

$$A(v_i) +_q B(v_j) \quad = C \qquad (1)$$

Where *A, B* are shapes to be juxtaposed, with *n* and *m* vertices respectively, and v_i and v_j indicate the first vertex of the edge (in a cyclical clockwise) to be considered in the juxtaposition operations (with $0<=i<=n$, and $0<=j<=m$). For instance the result of $A(2) +_q B(4)$ is graphically shown in Figure 4.



a) b)

Figure 3.a) First operand in a juxtaposing operation, b) second operand.

Figure 4. Shape C, resulting of $A(2) +_q B(4)$, given shapes A and B in Figure 3.

The set *P* is *closed* under the operator $+_q$, that is, the result of juxtaposing two shapes in *P* will result a new shape valid in *P*.

The operator $+_q$ also has the *commutative property*. In fact $A(2) +_q B(4) = B(4) +_q A(2)$.

For $+_q$ the *identity element* is the *empty shape* (*∅s*), which, as previously mentioned, means that there is no shape to juxtapose.

However, the operator $+_q$ has not the *associative property*, because the vertices in the second operand cannot be determined a priori.

Therefore, the set *P* with the $+_q$ operation is a *groupoid*.

3.3 The set of production rules *C*

The set *C* describes how to apply the juxtaposition operator. This set is defined by the following Extended Backus-Naur Form (EBNF) [Scowen, 1993] production rules:

$Region_i$::= Vertex+ (minimum 3 vertices)
Vertex::= <A, C, L>
A ::= *obtuse* | *right* | *acute*
C ::= *concave* | *convex*
L ::= *smaller* | *equal* | *bigger*

Hence, the $+_q$ operation between two vertices is a new vertex:

$<A_1,C_1,L_1> +_q <A_2,C_2,L_2> = <A_3, C_3, L_3>$ where:
$A_3 \in \{obtuse, straight, acute, \varnothing\}$
$C_3 \in \{convex, concave, \varnothing\}$
$L_3 \in \{smaller, equal, bigger, \varnothing\}$

The symbol ∅ means that no one of these properties is applied because the vertex disappears.

To compute A_3, C_3 and L_3, six tables are used (Tables 1 to 6). When juxtaposing two shapes, named A and B, Table 1 is applied when two vertices (one of shape A, and the other of shape B) are juxtaposed in order to obtain the new vertex in the new shape, named C. It sets the new angle, A_3, and the new convexity type, C_3, knowing A_1, A_2 and C_1 and C_2. It considers angle and convexity because convexity information depends on the angle information too. In fact, qualitative angles are defined using always the smaller angle and the convexity related to the angle indicates if the angle is measured from the interior or the exterior of the object. For instance, using Table 1, if A_1 is *acute*, and C_1 is *convex*, and

A_2 is *right* and C_2 is *concave* then in the new shape C, A_3 is *acute* and C_3 is *concave* (Figures 2-4).
Table 2 computes the new relative length L_3, knowing L_1 and L_2, and knowing that the edges used in the juxtaposition have a similar absolute length[1] (Figures 2-4).
Tables 3 and 4 are used in the case that, when juxtaposing A and B, the two vertices involved in the juxtaposition (one of shape A, and the other of shape B), instead of generating a new vertex in shape C, cause the vertex to disappear. This means that in the new shape C the juxtaposition of the two vertices generates a continuous line without a vertex, as it happens in the juxtaposition of the shapes in the example in Figures 3 and 4. In this case, the vertex disappears, but the relative lengths of the previous and following vertices have to be modified. Table 3 computes the new relative length of the previous vertex of the one disappeared, and Table 4 the new relative length of the following vertex.
Finally, if the edges involved in the juxtaposition are not of a similar length, then one vertex of one shape is juxtaposed to another vertex of the other shape, but the other vertex of the one of the shapes is juxtaposed to an edge of the other shape (see Figure 5 as an example). In order to juxtaposing the two vertices Table 1 is used to compute the angle and the convexity (A_3 and C_3), but to compute the relative length (L_3) Table 5 has to be used. Furthermore, Table 6 is used to compute the angle and the convexity (A_3 and C_3) of the new vertex that will appear when juxtaposing the vertex and the edge. In this case, the relative length of the new vertex has to be computed using the absolute length of the edges involved in the juxtaposition. It is not possible to compute the relative length of the new vertex, because it is only known the relative length of the edges inside a shape, and it is available any information about the relative length between the edges of two different shapes. Therefore, if we use the relative length information that we have in the QSD then as we do not have enough information about both relative lengths (one from one shape and the other from another) the result will be all the possible relative lengths between two edges, therefore we add a lot uncertainty. Thus, the absolute length is needed.
In all these tables there are situations that cannot happen when juxtaposing two shapes, therefore the symbol "*np*" appears in the corresponding cell of the table, meaning "not possible. If the symbol ∅ appears, it means that the vertex disappears, and then Tables 3 and 4 are needed. In each cell of these tables different possibilities may appear, for instance, when juxtaposing a convex and obtuse vertex, with a convex and acute vertex, the new vertex can be convex or concave and obtuse, or even dissapear. Moreover, in these tables the following notation is used:

- *cx* and *cv* for convex and concave respectively,
- *a, r,* and *o* for acute, right and obtuse respectively,
- and *s, e,* and *b* for small, equal and bigger.
-

[1] The absolute length is computed when creating the qualitative description of the shape, and it is stored for future uses.

	cx / a	cx / r	cx / o	cv /a	cv / r	cv / o
cx / a	cx + [a,r,o]	cx + o	cx + o, ∅, cv + o	∅	cv + a	cv + [o, r]
cx / r	cx + o	∅	cv + o	np	∅	cv + a
cx / o	[cx,cv]+o, ∅	cv + o	cv + o	np	np	∅
cv / a	∅	np	np	np	np	np
cv / r	cv + a	∅	np	np	np	np
cv / o	cv + [o, r]	cv + a	∅	np	np	np

Table 1. $+_q$ angle-convexity table.

	s	e	b
s	s	s	[s,e,b]
e	s	e	b
b	[s,e,b]	b	b

Table 2. $+_q$ relative length table.

	s	e	b
s	s	s	s
e	s	s	s
b	[s,e,b]	[s,e,b]	[s,e,b]

Table 3. $+_q$ table to compute the new relative length of the previous vertex to the one involved in the juxtaposition when this last one disappears.

	s	e	b
s	[s,e,b]	[s,e,b]	[s,e,b]
e	b	b	b
b	b	b	b

Table 4. $+_q$ table to compute the new relative length of the following vertex to the one involved in the juxtaposition when this last one disappears.

	s	e	b
s	[s,e,b]	[s,e,b]	[s,e,b]
e	[s,e,b]	b	b
b	b	[s,e,b]	[s,e,b]

Table 5. $+_q$ table to compute the new relative length of the two vertices juxtaposed when the edges to be juxtaposed have not similar absolute length.

	New Vertex
cx / a	cv / o
cx / r	cv / r
cx / o	cv / a
cv / a	np
cv / r	np
cv / o	np

Table 6. $+_q$ table to compute the convexity and angle of the new vertex appeared in the new shape when juxtaposing a vertex and an edge.

Note that, from Table 6, it can be concluded that it is not possible to juxtapose a concave vertex with an edge. Moreover, the convexity and angle of the new vertex is the inverse operation to the convexity and angle of the first vertex involved in the operation (see Figure 9 for an example).

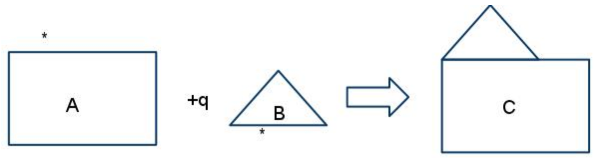


Figure 5. Example of the juxtaposition of two shapes where the absolute length of the edges involved in the juxtaposition is not similar.

Finally, a shape C is the $+_q$ of shapes A(v_i) and B(v_j) if:

- It has at most |A| + |B| -1 vertices, being |A| the number of vertices in A and |B| the number of vertices in B;
- Each vertex v_i in C must be either:
  - equal to one of the original vertices of A and B,
  - or the result of applying the $+_q$ operation to one vertex of A and another of B,
  - or equal to one of the original vertices of A and B, but with a new relative length,
  - or a new vertex the resulting from of juxtaposing a vertex and an edge.

3.4 The set of explicit axioms *A*

The set *A* specifies the restrictions or constraints that must be satisfied in order to apply the $+_q$ operator to obtain a correct new shape. The restrictions defined, some of them mentioned previously, are:

- It is not possible to overlap shapes when computing $+_q$. If the edges involved in the operation will return an overlapping shape, one of the shapes should be rotated before applying the $+_q$ operation (see Figure 8 for an example). After rotating the shapes in order to try the juxtaposition, to ensure that they do not overlap we have defined two restrictions. If any of them does not hold, it is because the juxtaposition is not possible because both shapes will overlap. The restrictions are:
  - The area of the final shape constructed by the juxtaposition has to be the addition of the areas of the two basic areas considered in the operation.
  - The number of vertices of the final shape constructed by the juxtaposition is at most $n+m-1$, where n and m are the number of vertices of the juxtaposed shapes.
- The shapes considered as operands are only simply connected and closed 2D regions.
- In $+_q$ it is necessary to specify the edges involved in the operation.
- Before applying the $+_q$ operation it is necessary to know the absolute length of the edges involved in the juxtaposition, in order to know if

$$Length(edge_1) \approx Length(edge_2)$$

- If the **absolute length of both edges is similar**, then having shape A of n vertices and shape B of m vertices, computing the operation $A(v_i) +_q B(v_j)$ means to compute the juxtaposition of two vertices of each shape, the vertex v_i of A is juxtaposed with vertex $(v_{j+1} \bmod m)$, and the vertex $(v_{i+1} \bmod n)$ is juxtaposed with the vertex v_j (see Figures 2 and 3 for examples). The process (**process I**) defined, **when juxtaposing edges with similar length**, to create the description of the resulting shape, named C, is as follows:

Step 1I. Starting by vertex 1 in A, it is copied in the new shape C and this step is repeated with the next vertex in a counterclockwise, up to the vertex which is one of the vertices in the $+_q$ operation.

Step 2I. This vertex is replaced by the result of applying $+_q$ between this vertex and the corresponding one to compute the juxtaposition in shape B using Tables 1 and 2. It may occur that the result obtained from these tables is that the vertex disappears, then:

Step 2.1I. If the vertex disappears because the angles to juxtapose are supplementary, then it is necessary to change the relative length of the previous vertex in shape C using Table 3. Moreover, it is also known that the relative length of the next vertex of figure C (which is the next one to the disappeared one of shape B) has to be modified also using Table 4 (see Figures 3-4 for an example).

Step 2.2I. If the vertex disappears because a convex vertex and a concave vertex are being juxtaposed, then the previous and following vertices of the new shape C may be completely different, depending on their position. Then, once more the absolute length of the edges is needed. If the next vertex does not *touch* the boundary of the other shape being juxtaposed (it does not intersect with any point of the boundary of the other shape), then its description does not change, and the next vertex is copied to the next shape. But, if it touches the edge of the other shape, then its new convexity is the inverse of the one it has and the relative length has to be computed using the absolute length information, also previous and next vertices relative length have to be recomputed. And, finally, if it is in contact with other vertex of the other shape, then its description is computed using Tables 1 and 2, and in the case that it disappears again, all this step is repeated up to the moment a vertex that does not disappears is reached. See Figure 6 for a complete example.

Step 3I. We continue copying the vertices of shape B (in a counterclockwise) until the next vertex in shape B related by the $+_q$ operation is reached. This vertex is replaced by the $+_q$ of this vertex and its corresponding one in shape A (repeating the step before described).

Step 4I. If there are still vertices in shape A, which have not been considered during the juxtaposition, they are copied to the new shape C too.

- But, if the **absolute length of both edges is not similar**, then it is established that the juxtaposition will be done by juxtaposing the first vertex of each edge of both shapes (Figure 5 shows an example), and always the shape with a smaller edge will be the first operand (shape A). Therefore, in order to juxtapose the first vertex of both shapes the process will be similar to the previous step, but the other vertex of shape A is juxtaposed with the involved edge in shape B. Therefore, the process (**process II**) to compute the juxtaposition **when juxtaposing a vertex and an edge** is different, as follows:

Step 1II. Starting by vertex 1 in A, it is copied in the new shape C and this step is repeated with the next vertex in a counterclockwise, up to the vertex that is the vertex involved in the $+_q$ operation with an edge or with the other vertex of shape B.

Step 2II. The corresponding $+_q$ operator is applied depending on:

Step 2.1II. If the juxtaposition is between a vertex and an edge, then this vertex is replaced by the result of applying $+_q$ using Table 6, and the new relative length is computed using the absolute lengths of the two edges involved. As a new vertex is appearing in the resulting shape, then the relative length of the next vertex has to be computed. This will be done in the following step.

Step 2.2II. If the juxtaposition is between two vertices (one of each shape), then, *Step 2I* of the previous algorithm is applied, but in order to calculate the new relative length of the new vertex (if it does not disappear), Table 5 is used instead of Table 2.

Step 3II. First of all, if a new vertex has appeared as consequence of the juxtaposition between a vertex and an edge, then the relative length of the first vertex found now (a vertex of B) has to be computed, once more using the absolute length information. Then, the rest of vertices of shape B are copied (in a counterclockwise) until the next vertex in shape B related with the $+_q$ operation. In this case, *step2.1II* or *step2.2II* has to be applied depending on the type of operation (juxtaposing a vertex and an edge, or juxtaposing two vertices).

Step 4II. As in the *Step3II*, if the result of the previous +q operator is also a new vertex (+q between an edge and a vertex), the relative length of first vertex considered now (of shape A) has to be recomputed. Then, if there are still vertices in shape A, which have not been considered during the juxtaposition, they are copied to the new shape C too.

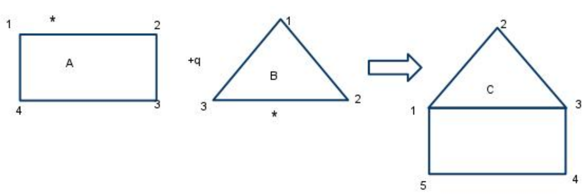


Figure 6. Example of the juxtaposition of two shapes which does not generate indetermination in the final description.

In Figure 6 a simpler example is shown, where *A(1)* $+_q$ *B(2)* is computed, which means that vertex number 1 of A is juxtaposed with vertex number 3 of B, and vertex number 2 of A is juxtaposed with vertex number 2 of B. The qualitative descriptions of shapes A and B, which start always by the upper-leftmost vertex (vertex number 1) are:

QualShape(A)= [[right, convex, smaller], [right, convex, bigger], [right, convex, smaller], [right, convex, bigger]].

QualShape(B)= [[acute, convex, equal], [acute, convex, smaller], [acute, convex, bigger]].

And, after applying the *process I*, the qualitative description of shape C is:

QualShape(C)= [[obtuse, convex, smaller], [acute, convex, equal], [obtuse, convex, bigger], [right, convex, smaller], [right, convex, bigger]].

Note that in this case, the qualitative description of shape C has no indeterminations. However, Figure 7 shows a more complicated juxtaposition example, where indeterminations are found in the relative length feature. In Figure 7, *A(3)* $+_q$ *B(1)* is computed, resulting the shape C. The qualitative description of the boundaries of shapes A and B are:

QualShape(A)= [[right, convex, bigger], [right, convex, bigger], [right, convex, smaller], [right, concave, smaller], [right, convex, bigger], [right, convex, smaller]].

QualShape(B)= [[right, convex, bigger], [right, convex, smaller], [right, convex, bigger], [right, convex, smaller]]].

In this case we are juxtaposing vertex number 3 of A with vertex number 2 of B, and vertex 4 of A with vertex 1 of B. And the complete description of the new shape C, after applying *process I*, is:

QualShape(C) = [[[right, convex, bigger], [right, convex, [smaller, egual, bigger]], [right, convex, [smaller, equal, bigger]], [right, convex, [smaller, equal, bigger]].

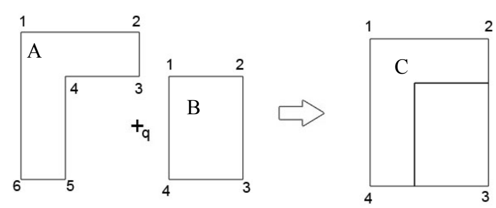


Figure 7. Example of the juxtaposition of a concave vertex in shape A and a convex vertex in shape B.

In order to show the result of applying *process II*, Figure 5 is used as an example. This example is similar to the one shown in Figure 6, but now the absolute lengths of the edges involved in the juxtaposition are different. In this case, *A(1)*

$+_q$ *B(2)* is also computed, but now it means that vertex 1 of A is juxtaposed with vertex 3 of B, and vertex 2 of A is juxtaposed with the edge going from vertex 2 to vertex 3 in shape B. The qualitative descriptions of shapes A and B are the same as in the case of shapes A and B of Figure 6. And, after applying *process II*, the result is the qualitative description of shape C as follows:

QualShape(C)= [[obtuse, convex, [smaller, equal, bigger]], [acute, convex, equal], [obtuse, concave, bigger], [right, convex, smaller], [right, convex, smaller], [right, convex, bigger]].

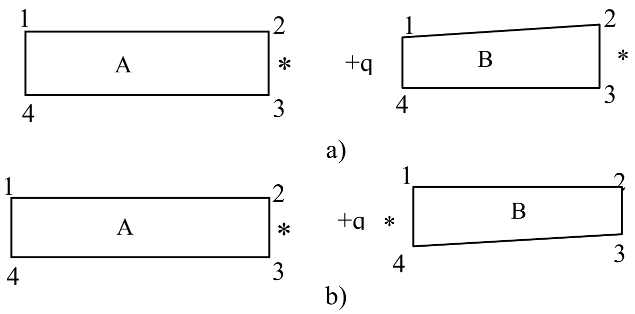


Figure 8. If we want to juxtapose figures A(2) +q B(2) as they are represented in a) the figures will overlap, therefore before calculating this operation figure B has to be rotated as it is shown in b), and we will calculate the operation A(2) +q B(4).

Finally, a more complex juxtaposition example is shown in Figure 9, in which after doing the juxtaposition all the vertices of shape *B* finish sharing the boundary of shape *A*, and the resulting QSD of the juxtaposed shape is given below.

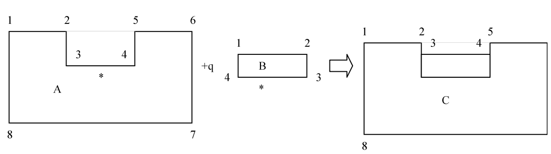


Figure 9. Juxtaposition example of two shapes.

In Figure 9, *A(3)* $+_q$ *B(4)* is computed. As the length of the edges to be juxtaposed is similar we follow the *process I*. First we start with vertex number 1 of shape *A*, and as it is not involved in the juxtaposition operation it is copied to the qualitative description of the new shape. The same happens with vertex 2 of *A*. Therefore, up to this moment *QualShape(C)* is:

QualShape(C) = [[right, convex, bigger], [right, convex, smaller]…]

Now, vertex 3 of *A* is the one that has to be juxtaposed with the vertex 1 of *B*. Vertex 3 of *A* is *[right, concave, smaller],* and vertex 1 of *B* is *[right, convex, bigger],* therefore looking at tables 1 and 2 we get that the new point in C disappears because we are juxtaposing a concave with a convex angle. Therefore, following the *step 2.2.I*, we know that the previous and following vertices of *C* may be completely different, depending on their position. As we have done a juxtaposition the next vertex to consider is the vertex 1 of shape *B*, and as it is *touching* the boundary of shape *A*, then we get that the new vertex in *C* is: *[right, concave, smaller].* Note that in this case the relative edge length feature has to be determined using the absolute edge information. Moreover, as is has appeared a new vertex, the relative length of the previous vertex has to be checked again in order to see if it is still correct. Therefore, the previous vertex of C is recomputed and now it is *[right, convex, bigger].*
The next vertex to consider is the vertex number 2 of the shape *B*, and as it is not involved in the juxtaposition it is copied to the description of shape *C*. Therefore, up to now *QualShape(C)* is:

QualShape(C) = [[right, convex, bigger], [right, convex, bigger], [right, concave, smaller], [right, convex, bigger]…].

The next vertex of *B* (vertex number 3) is again one that has to be juxtaposed, in this case with the vertex 4 of A. They are *[right, convex, smaller*] and *[right, concave, bigger]* respectively. Looking at tables 1 and 2 we get that once more the new vertex disappears in *C*, and again it is due to the juxtaposition of a concave vertex and a convex vertex. Therefore, following the *step 2.2I* we know that the following vertex or the previous one has to be recomputed. In this case, the previous vertex was number 2 in *B*, that has been copied in *C.* As before, this vertex has to be modified and it must be *[right, concave, bigger]*. And we now that the relative length feature of the next vertex has to be recalculated. Therefore now *QualShape(C)* is:

QualShape(C) = [[right, convex, bigger], [right, convex, bigger], [right, concave, smaller], [right, concave, bigger], right, convex, bigger]…].

Finally, now the last step is to copy the rest if vertices of shape *A* to the *QualShape(C)* up to the moment that we reach vertex 1 of *A* again. Thus, the final qualitative description of *C* is:

QualShape(C) = [[right, convex, bigger], [right, convex, bigger], [right, concave, smaller], [right, concave, bigger], [right, convex, bigger],[right, convex, smaller],[right, convex, smaller], [right, convex, bigger].

4 Conclusions and future work

A qualitative shape description scheme for the juxtaposition of two polygons, which cannot overlap, has been defined. The edges to be juxtaposed can be of similar length or of different length, which represents a step forward with respect to the work presented by Museros et al. (2010), which was able to qualitative juxtapose polygons only by edges of similar lengths. It represents an important improvement with respect to the work in Museros et al. (2010), because we have realized that when we juxtapose several shapes, there is a point in which it is not possible to juxtapose two shapes

with similar edge lengths, because after several juxtapositions the edge lengths have increased. Therefore, with this new method we will be able to juxtapose shapes even after several juxtaposition steps.
Moreover, now, given as input an image, its qualitative description is created, and some quantitative information is also stored, as its area and the absolute length of its edges. The juxtaposition operation has been designed using as much as possible the qualitative information of the shapes, and quantitative information, as the absolute length of the edges is used only in two cases: first in order to know which process has to be applied (*process I* or *process II*), and then when to much indetermination is get as result without using this information. But we are working on creating an absolute qualitative length system in order to use it in the QSD of a shape, and then study how it can be used to compute the QSD of a new shape resulting from the juxtaposition of two shapes completely in a qualitative way, without the use of quantitative information. In order to do this, the absolute qualitative length system has the be defined in function of the application itself, which means in function of the edges lengths of all the shapes that we can find in an specific application.
The shape description scheme is being implemented in an application where, given two images, first the qualitative shape description of the shape in each image is calculated, and then, the qualitative shape description of the shape obtained by the juxtaposition of both shapes.
We will use this application to show experimental results that will exemplify the validity of the defined qualitative shape description scheme composed only by the juxtaposition operation, although in the near future we want to extend this scheme by defining other operations, such as the difference between two shapes.
In this case, as the set P is represented by closed 2D polygons, the shapes are always *without holes* and *without curves*. But, in order to be able to apply this scheme to other domains, we want to extent the presented juxtaposition theory in order to consider the rest of features of the original qualitative description theory used for this scheme, which is able to describe curves and holes. Also we want to extent it considering the *Color* feature. Currently, we are working in its extension considering shapes with holes, where, as the juxtaposed shapes will only share boundaries, the qualitative description of each hole will be the same because the containers cannot overlap and the holes boundaries will not change. However, the new orientation of each hole in the resulting shape has to be calculated considering the original positions of the holes and the relative orientation of the edges considered for juxtaposition of the two original shapes.
With respect to the *Color* feature, as shapes cannot overlap, the new shape has the two colors of the juxtaposed shape, and up to this moment this is considered as a vector with the RGB coordinates of both shapes, but currently we are also working in relating each color to each part of the new shape by using qualitative orientation information.

Acknowledgments

This work has been supported by Universitat Jaume I – Fundació Bancaixa (P1·1A2008-14), by the Spanish Ministry of Science and Innovation R&D project ARTEMISA (TIN2009-14378-C02-01), by Pla Estratègic 2011/12 of the Departament d'Enginyeria i Ciència dels Computadors at Universitat Jaume I, and by IJCAI'2011 Student Travel Grant Program.

References

S. Ahmad, S. Chase, Design Generation of the Central Asian Caravanserai, *First ASCAAD International Conference, e-Design in Architecture KFUPM*, Dhahran, Saudi Arabia, pp. 43–58, 2004

E. Andaroodi, F. Andres, A. Einifar, P. Lebigre, N. Kandoa, Ontology-based shape-grammar schema for classification of caravanserais: a specific corpus of Iranian Safavid and Ghajar open, on-route samples, *Journal of Cultural Heritage 7* 312–328, 2006.

S. Belongie, J. Malik, J. Puzicha, Matching shapes. In: *8th IEEE International Conference on Computer Vision (ICCV2001)*, Vol. I, pp. 454–461, Vancouver, Canada, July, 2001.

S.C. Chase, Shapes and shape grammars: from mathematical model to computer implementation, *Environment and Planning B: Planning and Design, 16:215-242*, 1989.

Z. Falomir, J. Almazán, L. Museros, M. T. Escrig, Describing 2D Objects by using Qualitative Models of Color and Shape at a Fine Level of Granularity, *Proceedings of the Spatial and Temporal Reasoning Workshop at the 23rd AAAI Conference on Artificial Intelligence*, ISBN: 978-1-57735-379-9, Chicago, Illinois, USA, July 2008.

Z. Falomir, E. Jiménez-Ruiz, M.T. Escrig, L. Museros, Describing Images using Qualitative Models and Description Logics, *Journal of Spatial Cognition and Computation (JSCC)*, *Volume 11 Issue 1,* January 2011. doi:10.1080/13875868.2010.545611.

P. J. Ghosh, and K. Deguchi, *Mathematics of Shape Description: A Morphological Approach to Image Processing and Computer Graphics*. Wiley ed. ISBN: 978-0-470-82307-1, May 2008.

J. Halatsch, K. Antje, and G. Schmitt, Using Shape Grammars for Master Planning. In *Design Computing and Cognition '08, Proceedings of the Third International Conference on Design Computing and Cognition* (John S. Gero and Ashok K. Goel eds), ISBN978-1-4020-8727-1, Springer Netherlands, pp. 655-673, 2008.

T. W. Knight, Shape grammars and color grammars in design, *Environment and Planning B: Planning and Design 21*(6) 705 – 735, 1994

P. Lebigre, Towards an inventory on internet and the creation of a digital network of silk roads countries. In: K. Ono (Ed.), *Proceedings of the 2001 Tokyo Symposium for Digital Silk Roads, National Institute of Informatics*, Tokyo, ISBN: 4-86049-007-X, 2001.

G.J. Lu, A. Sajjanhar, Region-based shape representation and similarity measure suitable for content-based image retrieval. *Multimedia Syst*. No. 7 (2) 165–174, 1999.

B.M. Mehtre, M.S. Kankanhalli, W.F. Lee, Shape Measures for Content Based Image Retrieval: A comparison. *Information Processing and Management* 33 (3), pp. 319-337, 1997.

B.S. Morse, Computation of object cores from grey-level images. In: Ph.D. Thesis, University of North Carolina at Chapel Hill, 1994.

L. Museros, M. T. Escrig, A Qualitative Theory for Shape Representation and Matching for Design. *In: Proceedings ECAI 2004*, IOS Press ISSN 0922-6389, pgs. 858-862, 2004.

L. Museros L., M. T. Escrig, A Qualitative Theory for Shape Matching applied to Autonomous Robot Navigation. In: *Frontiers in Artificial Intelligence and Applications. Recent Advances in Artificial Intelligence Research and Development.* ISBN 1 58603 466 9, 2004.

L. Museros, Qualitative Theories on Shape Representation and Movement. Application to Industrial Manufacturing and Robotics. In: Ph.D. Thesis, University Jaume I, Castellón, Spain, 2006.

L. Museros L., L. Gonzalez-Abril, F. Velasco F., Z. Falomir, A Pragmatic Qualitative Approach for Juxtaposing Shapes, *Journal of Universal Computer Science 16(11),* 1410-1424, 2010.

L. Museros, Z. Falomir, F. Velasco, L. Gonzalez-Abril, and I. Martí, 2D qualitative shape matching applied to ceramic mosaic assembly, *Journal of Intelligent Manufacturing*, DOI 10.1007/s10845-011-0524-6, 2011.

M. Peura, J. Iivarinen, Efficiency of simple shape descriptors. In: *3rd International Workshop on Visual Form*, pp. 443–451, Capri, Italy, May, 1997.

W.J. Rucklidge, Efficient locating objects using Hausdorff distance. *Int. J. Comput. Vision* 24 (3) 251–270, 1997.

Roger S. Scowen, Extended BNF – A generic base standard. *Software Engineering Standards Symposium,* 1993.

G. Stiny, Introduction to Shape and Shape Grammars. *Environment and Planning B: Planning and Design,* 7: 343-351, 1980.

G. Stiny, Kindergarten grammars: designing with Froebel's building gifts, *Environment and Planning B 3*: 459-460, 1980.

Y. Wang, and J. P. Duarte, Synthesizing 3D forms: shape grammars and rapid prototyping. In: *Workshop on Generative Design, Artificial Intelligence in Design '98 Conference,* Lisbon, Portugal, pp. 7-18, 1998.

D.S. Zhang, G. Lu, A comparative study of Fourier descriptors for shape representation and retrieval. In: *5th Asian Conference on Computer Vision (ACCV02),* pp. 646–651, Melbourne, Australia, January 22–25, 2002.