Description Logics for Image Recognition: a preliminary proposal

Eugenio Di Sciascio **Francesco M. Donini**
Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari
Via G. Re David, 200 I-70125 Bari
{disciascio,donini}@poliba.it

1 Motivations

Recent years have seen an extreme increase of the size of digital image collections. In order to use these images, tools and techniques are needed to efficiently index and retrieve them. Traditional approaches to image databases have been based on the association to images of a textual description and then use text-based Database management systems to perform image retrieval ([2, 3]). There are two major drawbacks in this approach. First, the amount of human effort needed to annotate images when collections are large. Second, the difficulties in representing by textual descriptions the rich content of images, and the difference in human perception of the same image.

Lately there has been a growing interest towards content-based image retrieval. Techniques and systems have been proposed to index and retrieve images by their own visual properties, e.g. color, shape, texture, etc. ([6, 9]). All these techniques require image analysis and processing prior to indexing. While color and texture based retrieval have been widely experimented with good results, retrieval by shape is still a challenging problem. As a matter of fact, retrieval by shape requires image segmentation. Good results have been obtained also in this field, but a blind automatic segmentation provides a partition of images in regions, which is quite different from actual objects. Most systems overcome this difficulty either indexing regions [10] or reducing the domain of interest to limited classes of objects ([4, 13]) or also reverting to human assistance [5]. Reasons for these difficulty are clear: there is a gap to bridge between the low-level features obtainable by a vision algorithm and the complex structures of objects. Here we propose to bridge the gap with a structured approach based on description logic.

The use of structural descriptions of objects for the recognition of their images can be dated back to Minsky's frames.The idea is to associate parts of an object (and generally of a scene) to the regions an image can be segmented into. Structured descriptions of three-dimensional images are already present in languages for virtual reality like VRML [8]. However, the semantics of such languages is operational, and no effort is made to automatically classify objects with respect to their structural properties (e.g. a Jeep is a particular car) or semantical properties (cars and trucks are two classes of vehicles).

We believe that the use of a language for describing parts of images can provide several benefits. First of all, it semantically clarifies the process of image retrieval as an instance checking (classification, or, using a LOOM term which is here appropriate, *recognition*). Secondly, it allows to express complex queries to an image database, such as "retrieve all images with vehicles with a green background". Third, it allows a compilation of recognition (i.e., images are classified along known categories as soon as they are entered in the database) which can considerably speed up query answering. Finally, if the parts of an object are given a name — as in our proposal — this name can be used as part of the explanation of the recognition: i.e., the system can reasonably argue that an image contains a car, pointing to the chassis, the wheels, and the front screen. Obviously, the system can be wrong, as humans do; but at least, it can give explanations for the wrong recognition, allowing the image database administrator to correct or refine the description of the objects.

We present here a description logic which deals only with *exact* recognition of two-dimensional objects in images, i.e., recognition of objects by matching exactly their contour, up to linear transformations (rotations, scaling and translations). Obviously, image recognition should allow for a certain degree of vagueness in matching shapes and identifying colors. However, we believe that before giving a logic of approximate recognition, the logic of exact recognition must be clearly stated as such, and (at least some of) its complexity studied. We claim that the logic we present can be subsequently modified along the same lines of [15, 14] to accomodate for rough recognition, preserving the original framework.

2 Syntax

We assume that each image is divided into *regions*, where a region (denoted by letter r) is a connected part of an image with homogeneous color inside. Uniform color will be replaced by "almost uniform" (fuzzy) color in later versions of the logic. An image I is identified by the set of all its regions. Each region in an image comes with a color $c(r)$ and a contour (or, edge set) $e(r)$. Also a set of regions $\{r_1, \ldots, r_n\}$ has a global contour, which is the composition of the contours $e(r_1), \ldots, e(r_n)$, dropping all parts of contours which are internal to $r_1 \cup \ldots \cup r_n$.

Our framework does not require a particular method to represent edges and contours; e.g., B-splines, discrete Fourier transforms, etc. could be used. Also the representation of the color c is independent of our framework; e.g., it can be given as a three-component vector RGB, or HSV, etc.

Given a linear transformation τ (a rotation, a scaling, and a translation) we denote as $\tau(e(r))$ the (pointwise) linear transformation of $e(r)$. We recall that linear transformations are a mathematical group.

Any shape which is likely to come with a uniform color in an image — such as rectangles, circles, squares, but also the chassis of a car — is a *basic shape*. Basic shapes are given a concept name B, and are defined by a *prototype* contour $e(B)$. Intuitively, a region r is recognized as a shape B iff there exists a linear transformation τ mapping $e(r)$ in $e(B)$, i.e., such that $\tau(e(r)) = e(B)$. In this way, we can abstract from the method used to recognize basic shapes, as long as the method is robust wrt rotations, scaling, and translations. The equality $\tau(e(r)) = e(B)$ will be relaxed to approximate equality (distance within a given range) in a later (fuzzy) version of the logic.

Definition 1 *A* basic shape *is a pair* $(B, e(B))$ *where* B *is a concept name, and* $e(B)$ *(the* prototypical *contour of* B*) is the contour of an image region corresponding to the shape.*

Following hierarchical representations, the contour $e(B)$ can be assumed to be given with respect to the centroid of the region — i.e., coordinates for $e(B)$ are given considering the centroid of the region as the origin.

As an example, one can define the two basic shapes `rectangle` and `halfCircle` by giving theircontour contours $e(\texttt{rectangle}) = E_r$ and $e(\texttt{halfCircle}) = E_h$.

Definition 2 *A* basic role *is a triple* (R, τ, c)*, where* R *is a role name,* τ *is a linear trasformation (i.e., a rotation angle, a scaling factor, and a translation), and* c *is a color.*

Definition 3 *A* basic component *is denoted as* $\exists(R, \tau, c).B$*, where* B *is a basic shape.*

Intuitively, a component describes a region whose contour is $\tau(e(B))$, which is filled with color c. The role name R gives the name of the component as part of a complex shape. For example, the right pier (of a white complex shape) could be denoted as $\exists(\texttt{rightPier}, \tau_2, \mathit{white}).\texttt{rectangle}$. Anticipating semantics, we point out that the role name only labels the part in the complex shape, defined as follows.

Definition 4 *A* complex shape *is a conjunction of components, denoted as* $C_1 \sqcap \cdots \sqcap C_n$*, where* C_i *is either a basic component, or a complex component (see Definition 6). A* complex shape definition *is denoted as* $C \doteq D$*, where* C *is a concept name and* D *is a complex component.*

For example, the shape of a white arch can be defined as a vault (depicted as a white `halfCircle`) over two piers (depicted as white `rectangle`s), with suitable positioning of the three parts with the three linear tranformations τ_1, τ_2, τ_3:

$$\begin{aligned} \texttt{arch} &\doteq \exists(\texttt{vault}, \tau_1, \mathit{white}).\texttt{halfCircle} \\ &\sqcap \exists(\texttt{rightPier}, \tau_2, \mathit{white}).\texttt{rectangle} \\ &\sqcap \exists(\texttt{leftPier}, \tau_3, \mathit{white}).\texttt{rectangle} \end{aligned}$$

The contour of the whole arch $e(\texttt{arch})$ is the familiar arch contour, composed of the edges of the three basic shapes joined together, deleting edges which are inside the union of the regions.

Definition 5 *A* general role *is a pair* (R, τ)*, where* R *is a role name, and* τ *is a linear transformation.*

A general role has the same properties of basic roles, but for the color. The color of a complex shape results from the colors of the basic components, which cannot be overridden. General roles apply to complex shapes, to build nested structures.

Definition 6 *A* complex component *is denoted as* $\exists(R, \tau).C$*, where* C *is a complex shape.*

For complex components, the transformation τ is rigidly applied to all subcomponents. E.g., the complex component

$$\exists(R_1, \tau_1).(\exists(R_2, \tau_2, c_2).B2 \sqcap \exists(R_3, \tau_3, c_3).B3)$$

describes all shapes obtained by applying the transformation $(\tau_1 \circ \tau_2)$ to $e(B2)$, and filling the region $\tau_1(\tau_2(e(B2)))$ with color c_2, and applying $\tau_1 \circ \tau_3$ to $e(B3)$, and filling $\tau_1(\tau_3(e(B3)))$ with c_3. Observe that $(\tau_1 \circ \tau_2)$ and $(\tau_1 \circ \tau_3)$ are still linear transformations.

Continuing the example, a (kind of) ancient roman city gate `threePortalsGate` may be defined as two small arches on the two sides of a bigger one, using transformations τ_4, τ_5, τ_6:

$$\texttt{threePortalsGate} \doteq \exists(\texttt{leftPortal}, \tau_4).\texttt{arch}$$

$$\sqcap \quad \exists(\mathtt{rightPortal}, \tau_5).\mathtt{arch}$$
$$\sqcap \quad \exists(\mathtt{mainPortal}, \tau_6).\mathtt{arch}$$

Intuitively, such description is satisfied by all images containing a shape which can be matched with the above one, modulo a linear transformation — i.e., any image containing a rotated, translated and/or scaled gate satisfies the above description.

Observe that other kinds of gates can be defined. They would all be classified as shapes containing special kinds of arches.

A usual description logic can be added on top of the constructs above, allowing to state, e.g., that both `arch`es and `doors` (with different shapes and parts) are subclasses of `entrance`s.

3 Semantics

The domain Δ of an interpretation $\mathcal{I}$ is a set of images $I_1, \ldots, I_n$, where each image is considered as the set of its regions. To simplify notation, we let Δ be the (flat) set of all regions in all images $\Delta = I_1 \cup \ldots \cup I_n$, and we assume that sets of regions are formed only among regions of the same image.

An interpretation maps a shape to a set of regions. An interpretation comes in two ways: either a linear transformation is specified (written $\mathcal{I}(\tau)$), or not (we write simply $\mathcal{I}$). In the second case, we assume that *there exists* a linear transformation with particular properties.

An interpretation $\mathcal{I}(\tau)$ maps every basic shape B with prototypical contour $e(B)$ to the set of all regions in the images of Δ whose contour is transformed by τ into $e(B)$. In formulae,

$$B^{\mathcal{I}(\tau)} = \{r \in \Delta | e(r) = \tau(e(B))\}$$

The region is said to be "interpreted as a B". Then, dropping the transformation, the interpretation of a basic shape is:

$$\begin{aligned} B^{\mathcal{I}} &= \{r \in \Delta | \exists \tau : r \in B^{\mathcal{I}(\tau)}\} \\ &= \{r \in \Delta | \exists \tau : e(r) = \tau(e(B))\} \end{aligned}$$

E.g., $\mathtt{rectangle}^{\mathcal{I}}$ is the set of all rectangular regions (with a fixed ratio between sides) in Δ. Then a particular region can be at first interpreted (recognized) as a `rectangle` by a suitable segmentation algorithm.

A similar interpretation is assumed for basic components, taking color into account:

$$(\exists(R, \tau_1, c).B)^{\mathcal{I}(\tau)} = \{r \in \Delta | e(r) = \tau(\tau_1(e(B))) \wedge c(r) = c\}$$

Again, dropping the transformation one obtains the interpretation of the basic component as regions that can be matched by some linear transformation:

$$\begin{aligned} (\exists(R, \tau_1, c).B)^{\mathcal{I}} &= \{r \in \Delta | \exists \tau : r \in (\exists(R, \tau_1, c).B)^{\mathcal{I}(\tau)}\} \\ &= \{r \in \Delta | \exists \tau : e(r) = \tau(\tau_1(e(B))) \wedge c(r) = c\} \end{aligned}$$

A region in the interpretation of a component is then said as the R part of a complex shape. E.g., the region previously recognized as a `rectangle` can be recognized as the `rightPier` of an arch, if also the other parts are recognized, in the proper positions.

Complex components are interpreted in the same way, excluding the color.

Complex shapes are interpreted as sets of regions in one image, such that there exists one linear transformation, mapping rigidly all parts of the complex shape definition to (sets of) regions in the image. When the complex shape is composed by basic shapes only, it is sufficient to find as many different regions as the components of the complex shape — each region mapped into a component. However, when the complex shape is composed by nested complex shapes, the interpretation requires to have different *sets of* regions $s_1 \ldots, s_n$, where each set can be mapped into a complex component.

$$\begin{aligned} (C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}(\tau)} = \; & \{\{r_1, \ldots, r_m\} \in \Delta | \\ & \exists s_1 \cdots \exists s_n \subseteq \{r_1, \ldots, r_m\} : \\ & (i \neq j \Rightarrow s_i \neq s_j) \\ & \wedge \forall i \in 1..n : s_i \in (C_i)^{\mathcal{I}(\tau)}\} \end{aligned}$$

We recall that the regions in a set $\{r_1, \ldots, r_m\}$ are assumed to belong to the same image. Dropping τ, one obtains the interpretation of a complex shape.

$$\begin{aligned} (C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}} = \; & \{\{r_1, \ldots, r_m\} \in \Delta | \\ & \exists \tau \exists s_1 \cdots \exists s_n \subseteq \{r_1, \ldots, r_m\} : \\ & (i \neq j \Rightarrow s_i \neq s_j) \\ & \wedge \forall i \in 1..n : s_i \in (C_i)^{\mathcal{I}(\tau)}\} \end{aligned}$$

Note that the same interpretation $\mathcal{I}(\tau)$ maps all complex components $C_1, \ldots, C_n$ (otherwise the overall shape would not be preserved). This deals with rigid transformations of all components of a complex shape.

This semantics deals with the (simple) case in which regions of basic components are always properly segmented. This is not true whenever two touching regions of basic components in a complex shape have the same color; in that case, a single region formed by the union of regions with the same color would be segmented. To deal with this case, the contour of the complex shape should be taken into account — e.g., $\mathtt{arch}^{\mathcal{I}}$ should contain also a single region with the shape of a (completely white) `arch`. However, we observe that dealing with this problem would entail also a correct treatment of mimesis, which is outside the scope of this proposal. Hence, we keep here the simpler semantics of differently colored touching regions.

Definition 7 *An image $I = \{r_1, \ldots, r_n\}$ in the domain Δ of an interpretation $\mathcal{I}$* satisfies *a shape description D if there exists a subset of $\{r_1, \ldots, r_n\}$ which belongs to $D^{\mathcal{I}}$. An interpretation $\mathcal{I}$* satisfies *a description D if Δ contains an image satisfying D. A shape description is* satisfiable *if there exists an interpretation satisfying it.*

Observe that descriptions could be unsatisfiable: if two components have overlapping regions, no image can be segmented in a way that satisfies both components. This constraint would be dropped if we moved to describing images of three-dimensional scenes.

Definition 8 *A description D_1* subsumes *a description D_2 if every interpretation satisfying D_2 satisfies also D_1.*

For example, the description `arch` subsumes `threePortalsGate`. Intuitively, this happens because to satisfy `threePortalsGate` the domain of an interpretation $\mathcal{I}$ should contain an image with three arches, hence $\mathcal{I}$ must satisfy also `arch`.

We note that even if we did not introduce *backgrounds*, they could be added to our framework as special basic components $\exists(\texttt{background}, c).\top$ with the property that a region b satisfies the background simply if colors match. Also, more than one background could be added; in that case background regions should not overlap, and the matching of background regions should be considered after the regions of all the basic shapes recognized are subtracted to the background regions.

4 Reasoning services

An image database *is* an interpretation. Technically, one might think about an image database as an ABox; however, observe that usually a *Domain Closure Assumption* (DCA) is made for images — that is, there are no regions but the ones which can be seen in the images themselves. This transforms the problem of instance checking to simpler model checking. Obviously, DCA should be dropped for a description logic dealing with three-dimensional shapes (and scenes).

The possible reasoning services in a description logic for image recognition are:

1. **image recognition**: given a new image and a collection of descriptions, the new image is classified by finding the most specific descriptions it satisfies;
2. **description satisfiability**: verify if a given description may contain at least one image — e.g., there are no overlapping regions in the description;
3. **image retrieval**: given a database of images and the description of a scene, retrieve all images that belong to the description;
4. **description subsumption** (and classification): given a (new) description D and a set of descriptions $D_1, \ldots, D_n$, decide if D subsumes/is subsumed by each D_i, $i \in 1..n$.

Observe that the expensive process of image retrieval can be simplified by image recognition, in the usual way a classification wrt a TBox can help instance checking: Given a query description D, if there exists a collection of descriptions $D_1, \ldots, D_n$ and all images in the database were already classified wrt $D_1, \ldots, D_n$, then it may suffice to classify D wrt $D_1, \ldots, D_n$ to find (most of) the images satisfying D. E.g., to answer the query asking for images containing an `arch`, a system may classify `arch` and find that it subsumes `threePortalsGate`; then the system can include in the answer all images recognized as containing ancient roman gates, without recomputing whether these images contain an arch or not. The whole process will be convenient if classifying the query description on-line is computationally less demanding than computing the answer from scratch. The problem of computing subsumption between descriptions is dealt with in the next section.

5 Computing subsumption and satisfiability

Deductive services in this simple description logic rely on the prototypical contours of basic shapes.

Definition 9 *The prototypical image $I(D)$ of a shape description D is an image containing a set of regions defined recursively as follows:*

1. *if D is a basic shape B, then $I(B) = \{r\}$ where r is the region with contour $e(B)$;*
2. *if D is a basic component $\exists(R, \tau, c).B$, then $I(D) = \{r\}$, where r is the region with contour $\tau(e(B))$, filled with color c;*
3. *if D is a complex shape $C_1 \sqcap \ldots \sqcap C_n$ then $I(D) = I(C_1) \cup \cdots \cup I(C_n)$, where $I(C_i)$ is the prototypical image of the i-th component C_i;*
4. *if D is a complex component $\exists(R, \tau).C$, then $I(D) = \tau(I(C))$, where $\tau(\{r_1, \ldots, r_n\})$ denotes the set of regions obtained from $\{r_1, \ldots, r_n\}$ by transforming each contour $e(r_i)$ to $\tau(e(r_i))$.*

In practice, from a shape description one builds its prototypical image just applying the stated transformations and color fillings to its components.

Satisfiability of a shape description D can now be easily decided by trying to build $I(D)$, and see if there are not overlapping regions.

Proposition 1 *A shape description D is satisfiable iff its prototypical image $I(D)$ contains no overlapping regions.*

It is also easy to decide subsumption, thanks to the following general property relating subsumption to prototypical images.

Proposition 2 *A shape description D_1 subsumes a shape description D_2 if and only if the prototypical image $I(D_2)$ satisfies D_1.*

Observe that since there is always the inverse of a linear transformation τ, if D_1 and D_2 are basic shapes, either they are equivalent (each one subsumes the other) or neither of the two subsumes the other.

6 Comparison with related work

Some formalisms were proposed, integrating Description Logics and image retrieval [11], and spatial reasoning [7]. Both proposals build on the clean integration of Description Logics and *concrete domains* of [1]. However, in our opinion neither of the formalisms can be used to build complex shapes by nesting (more simple) shapes, with the *relative positions* of the shape components wrt the overall shape, and — most of all — to *reason* about shape descriptions with the usual concepts of subsumption, recognition, and satisfiability.

Moreover, we believe that the extensional Tarski-style semantics given to our formalism, if suitably extended with a (fuzzy) measure of the predicate $\in$, contributes to a smooth integration of Description Logics and existing image retrieval systems, which interpret a user query as a set of images matching the image query — up to a given threshold.

7 Future work

This proposal is still in progress. We need to extend the formalism to *approximate* matching of shapes and colors. In particular, observe that the similarity degree that is presently computed by all image retrieval systems should be extended to a similarity degree of complex shapes, taking into account both the (usual) similarity degree between simple shapes, and a (new) similarity degree between the *relative positions* of shape components in an image and in a shape description. To this aim, we envisage the application of frameworks for "fuzzy" description logics such as [15, 14]. We are designing an implementation of our formalism adapting the description logic system CLASSIC [12].

Acknowledgements

We thank Marco Schaerf for useful discussions about VRML. This research has been supported by research project CNR-DeMAnD. Francesco Donini acknowledges the Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", and in particular the research project CNR-LAICO, for supporting this research.

References

[1] Franz Baader and Philipp Hanschke. A schema for integrating concrete domains into concept languages. In *Proc. of IJCAI'91*, pages 452–457, Sydney, 1991.

[2] N.S. Chang and K.S. Fu. Query by pictorial example. *IEEE Trans. Software Engineering*, 6(6), 1980.

[3] S.K. Chang and et al. Intelligent image database system. *IEEE Trans. Software Engineering*, 14(5), 1988.

[4] E. Di Sciascio and M. Mongiello. Query by sketch and relevance feedback for content-based image retrieval over the web. *Journal of Visual Languages and Computing*, 1999. To appear.

[5] M. Flickner and et al. The QBIC system. *IEEE Computer*, 28(9), 1995.

[6] V.N. Gudivada and J.V.Raghavan. Special issue on content-based image retrieval. *IEEE Computer*, 28(9), 1995.

[7] V. Haarslev, C. Lutz, and R. Moeller. Foundations of spatioterminological reasoning with descrition logics. In *Proc. of KR'98*, pages 112–123, 1998.

[8] Jed Hartman and Josie Wernecke. *The VRML 2.0 Handbook*. Addison-Wesley, 1996.

[9] R. Jain. Special issue on visual information systems. *Communications of the ACM*, 40(12), Dec. 1997.

[10] W.Y. Ma and B.S. Manjunath. NETRA: A toolbox for navigating large image database. In *Proc. of IEEE ICIP*, 1997.

[11] C. Meghini, F. Sebastiani, and U. Straccia. The terminological image retrieval model. In *Proc. of ICIAP-97*, number 1311 in LNCS, pages 156–163. Springer-Verlag, 1997.

[12] Peter F. Patel-Schneider, Deborah L. McGuiness, Ronald J. Brachman, Lori Alperin Resnick, and Alexander Borgida. The CLASSIC knowledge representation system: Guiding principles and implementation rational. *SIGART Bulletin*, 2(3):108–113, 1991.

[13] Y. Rui, A.C. She, and T.S. Huang. Automated shape segmentation using attraction-based grouping in spatial-color-texture space. In *Proc. of IEEE ICIP*, 1996.

[14] U. Straccia. A fuzzy description logic. In *Proc. of AAAI'98*, pages 594–599, 1998.

[15] John Yen. Generalizing term subsumption languages to fuzzy logic. In *Proc. of IJCAI'91*, pages 472–477, 1991.