

# Visual interfaces for a semantic content-based image retrieval system

Hagit Hel-Or<sup>a</sup> and Dov Dori<sup>b</sup>

<sup>a</sup>Dept of Computer Science, University of Haifa, Haifa, Israel

<sup>b</sup> Faculty of Industrial Engineering, Technion, Haifa, Israel

## ABSTRACT

In an earlier study a Semantic Content Based Image Retrieval system was developed. The system requires a *Visual Object Process Diagram - VOPD* to be created for each image in the database and for the query. This is a major drawback since it requires the user and database manager to be acquainted with the rules and structures of the VOPD. This is not trivial and in fact troublesome to the naive user. To overcome this drawback two approaches are presented in this work, to provide an interface to the Image Retrieval system and to bypass the need of manually creating VOPD representations.

**Keywords:** semantic Content based Image Retrieval, Image search, Visual Object Process Diagrams, Image Database, Image retrieval interface

## 1. INTRODUCTION

A semantic content based image retrieval system has been developed which is based on the object-process methodology. The suggested approach bridges the gap between the classical text-based keyword oriented approach and the visual feature based approach. In our approach image content, based on the visual scene, forms the basis for both indexing and retrieval by constructing Visual Object-Process Diagrams (VOPDs) to represent images. These descriptions involve the objects in the scene and their inter- and intra-relationships. VOPDs provide abstract, high-level representation of the layout of the scene, as well as a distinction between the dominant core of the scene and its background. The VOPD representation for images is hierarchical in nature and encompasses both the high level layout of the visual scene and the low-level visual features. Given the hierarchical nature of the VOPD representation, the matching process is defined at several levels allowing for keyword, structure and attribute matching. The principles and guidelines for the construction of VOPDs and the outline of the matching process between two VOPDs has been presented in earlier work.

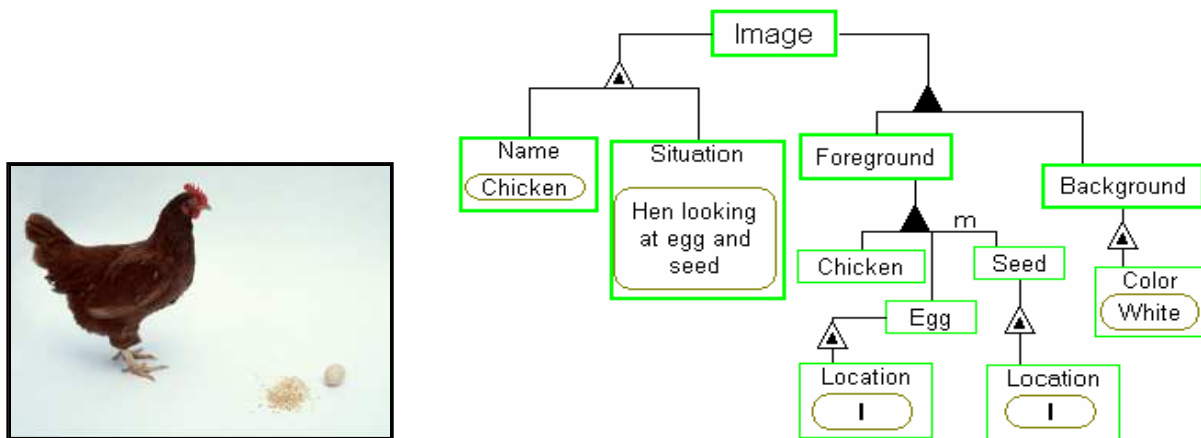
The VOPD based Semantic Content Based Image Retrieval system requires a VOPD to be created for each image in the database and for the query. This is a major drawback since it requires the user and database manager to be acquainted with the rules and structures of the VOPD. This is not trivial and in fact troublesome to the naive user. To overcome this drawback two approaches are presented in this work, to provide an interface to the Image Retrieval system and to bypass the need of manually creating VOPD representations.

## 2. THE VOPD REPRESENTATION OF IMAGES

In this section we briefly review the *Semantic Content Based Image Retrieval System* that has been previously developed.<sup>1-3</sup> In this system, Cognitive Image Content forms the basis for image retrieval and for the user interface. The approach is superior to both the non-intuitive approach of low level features as search keys and the key word search which does not capture the structure and dynamics of the image scene. The system has hierarchical characteristics and incorporates both low-level image features and textual key words as descriptors of the image. In this retrieval system, images are represented using diagrams called *Visual Object-Process Diagrams (VOPD)*. These are graph-like descriptions<sup>1-3</sup> involving the objects in the scene and their inter- and intra-relationships. This allows for abstract, high-level representation of the layout of the scene, as well as a distinction between the dominant core of the scene and its background. This representation inherently includes textual key-words as object description and low level features as object attributes. Querying is performed by representing the sought image with a VOPD and finding the images in the database whose VOPD best match the query. An example image and its VOPD representation are shown in Figure 1.

---

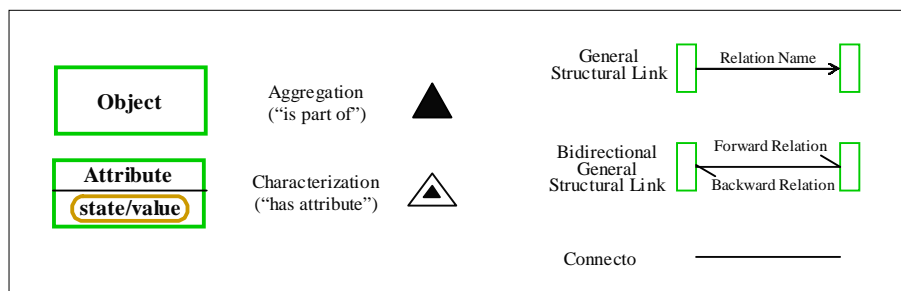
Send correspondence to [hagit@cs.haifa.ac.il](mailto:hagit@cs.haifa.ac.il)



**Figure 1.** An image (left) and its Visual Object-Process Diagrams (VOPD).

The VOPD representation of images is well defined, as is the matching algorithm for comparison between two images represented as VOPDs. Both are directed by a collection of structure definitions, rules and constraints.<sup>1,2,4</sup> The VOPD is a tree structure (with possible inter-node connections) that emerges naturally and is based on, the hierarchical nature of the image content. It is designed to enable efficient comparison between two such structures. The structural elements of the VOPD are shown in Figure 2. These notations include objects, attributes, the fundamental structural relations of aggregation and characterization, as well as general unidirectional and bi-directional structural links. The same VOPD symbols set is used for both indexing and query generation. All VOPDs are created from a generic VOPD shown in Figure 3. The root of the generic VOPD tree is an object node denoted Image. It has two descendants, one representing the image attributes and the other - the content of the image. Image attributes include the name of the image, and the *Situation*. The latter represents observations or conclusion that can be inferred from the scene, though not necessarily visible, such as effects, mood, atmosphere and lighting.

Basic to the generic VOPD is the distinction between foreground and background within the root sub-tree associated with image content. This distinction is primarily a content-based, or semantic observation. Viewers naturally distinguish between foreground and background image contents. Foreground image content is typically of greater importance and significance within the image. The inherent distinction between foreground and background within the VOPD is exploited during matching and image retrieval. The generic VOPD is expanded with branches using the structural relations: aggregation and characterization (Figure 2). The foreground and background nodes of the VOPD may consist of any number of descendent objects organized hierarchically in any number of levels. The Background and Foreground sub-trees of the VOPD contain the image cognitive and



**Figure 2.** The structural elements used for building VOPDs.

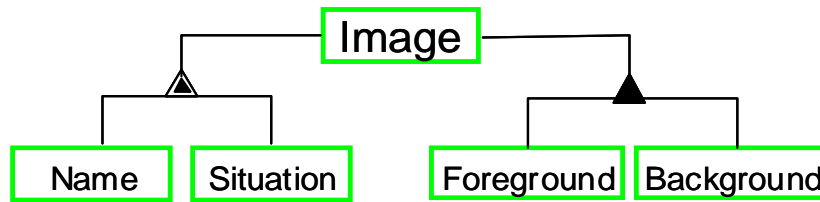


Figure 3. The generic VOPD.

formative content associated with the background and foreground of the scene. If there is no clear separation between the foreground and the background in the image, then the entire image is represented under the foreground sub-tree of the VOPD. Inter-branch connections may be incorporated by using the General (unidirectional or bi-directional) Structural links. These are used to describe structure or dynamic features in the image as expressed between two objects (e.g. Figure 4). Each of these relations is associated with an explanatory label. The aggregation connection may be associated with a multiplicity expression (a positive integer or the letter "m" representing "many") denoting the multiplicity of the objects in the scene (e.g. the object "seed" in Figure 1).

Each object in the VOPD representation is associated with the following optional attributes:

- **Name** - The private name of the object, e.g. "Clinton", "Paris", "Macintosh".
- **Color** - The color of the object is specified by this attribute. Color value is represented either by a color name selected from a predefined list of colors, or by numeric RGB values. Currently only color values may be associated with each object. Future versions will support multi-color names and/or colormaps.
- **Location** - The Location attribute describes the approximate location of the object within the image frame. The location is specified as one or more of the enumerated rectangles obtained by dividing the image plane into equal rectangles. Figure 5a shows the division of an images into 9 rectangles enumerated "A"..."I".
- **Size** - The size of the object is given as a percentage of the image size.
- **Completeness** - This attributes describes the visibility of the object in the image. The attribute may have one of the following possible values: "Whole" - when the object is fully visible, "Truncated" - when the object exceeds the boundaries of the image, and "Occluded" - when the object is partially occluded by another object in the image (see Figure 5b).
- **Importance** - This attribute is used only in Query VOPDs. It refers to the importance assigned to the object by the user during the Query VOPD generation process. The attribute may have one of the following possible (numeric) values: "Must be" (3), "Should be" (2 - the default value), "Not important" (1), and "Don't care" (0).

In addition to their value, each attribute in a query VOPD may be assigned an importance factor in the range (0...3) with 3 denoting "must be" and 0 "don't care". The above defined object attributes are all optional. An object may have no value associated with its attributes. Additional attributes may be added in the future.

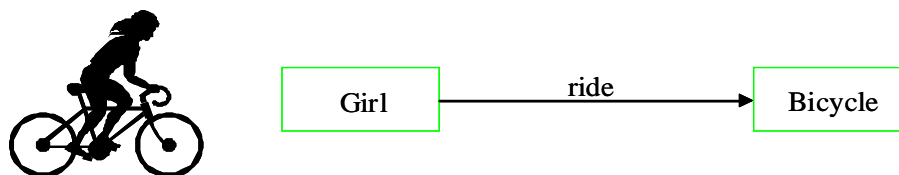
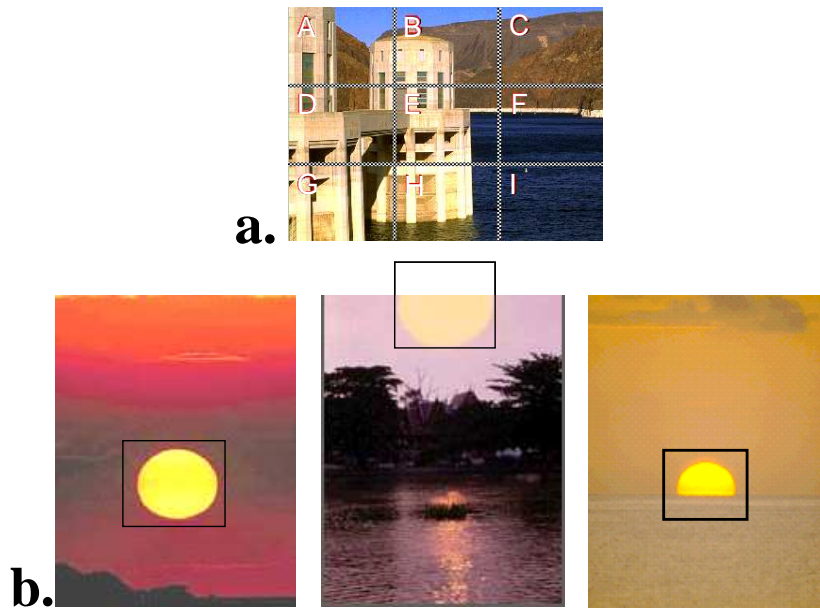


Figure 4. Example using a unidirectional structural link.



**Figure 5.** a. An image divided into 9 regions marked "A"..."I" which are used for the "Location" attribute of objects. b. The "Completeness" value for object "sun" are (left to right) "Whole", "Truncated" and "Occluded".

### 3. VOPD INTERFACE

A major drawback of the VOPD based Semantic Content Based Image Retrieval system is the need to associate every image with a VOPD representation. Manual insertion of the VOPDs is time consuming and requires expertise and knowledge of the rules and structures of the VOPD. This is not trivial and in fact troublesome to the naive user. To overcome this drawback two approaches are suggested that provide an interface into the Image Retrieval system and that bypass the need of manually creating VOPD representations. The optimal solution would be to provide an automatic VOPD generator which provides a VOPD for any given input image. This, however, would require very sophisticated Computer Vision and Image Understanding techniques, which are currently unavailable. In this paper, a practical approach is proposed. Rather than attempting to fully automate the VOPD creation for all images, two VOPD generators are suggested which relax the constraints and limitations of an automatic VOPD generator and enable a working interface.

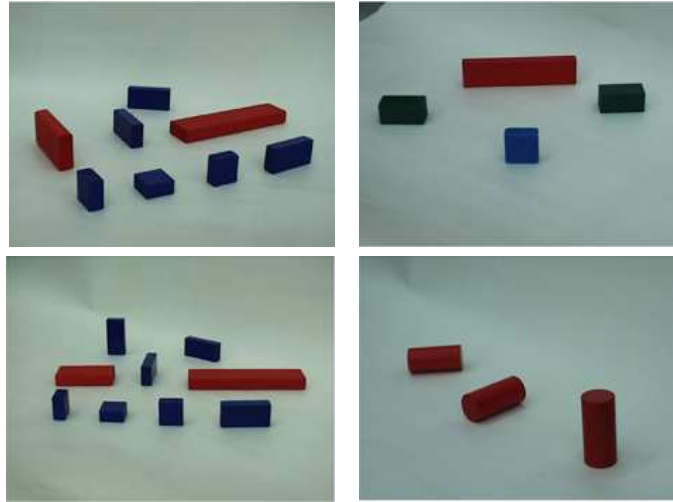
In the first approach, an automatic VOPD generator is still assumed however the input to the generator is restricted to images from within a very specific and predefined class. Computer Vision techniques can be exploited and are known to perform very well for highly constrained inputs. Details of this approach are given in Section 3.1.

In the second approach, the fully automatic characteristic of the system is relaxed. In this approach, an interface tool is developed to guide the naive user into supplying the necessary details required to create the image VOPD. The interface tool is very intuitive and is based on the simple process of creating an image using visual and graphic tools. The naive user requires almost no training in using this tool to create VOPDs from images or to create VOPD queries. Details of this approach are given in Section 3.2.

#### 3.1. Automatic VOPD Generator

In this approach an automatic VOPD generator is developed for a specific class of images. As a demonstration of the approach, the class of "Block World" images is chosen. This class includes images of polyhedral building blocks of different colors. Examples of images in this class are shown in Figure 6.

The automatic VOPD generator consists of two main parts:



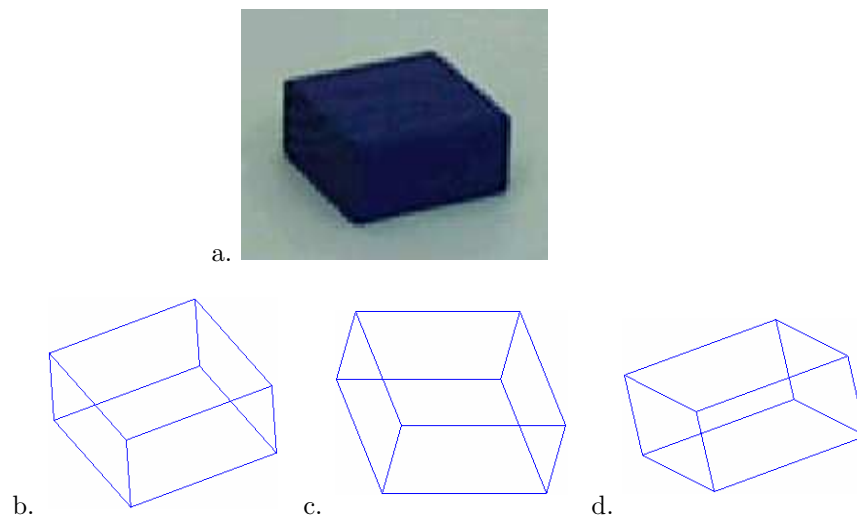
**Figure 6.** Examples of images in the "Block World" class.

- Preprocessing - Creating the database of models that may appear in Block World images. This stage is performed off-line, and is allowed extensive run time, memory and human intervention.
- Online VOPD generation - Given an input image of the class, automatically create its representative VOPD based on the model database. This stage should be efficient in terms of run time.

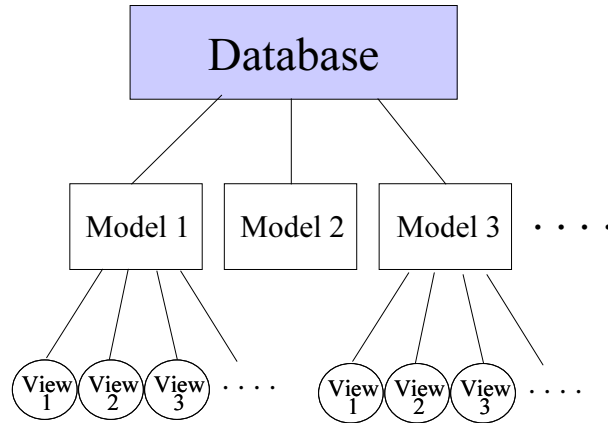
### 3.1.1. Creating the database of models

The objects in the Block World images are colored building blocks. For each object, a model is built and a set of representations of the model under all possible views is created. Since run time and memory is not an issue in this stage, all possible views of all possible models are created and maintained in memory rather than created on the fly during online VOPD generation which should be time efficient. The database of models is created as follows:

1. Images of individual blocks are acquired (see Figure 7a).



**Figure 7.** The blocks are modeled as wireframe models.  
a) Block image. b) Wireframe model. c-d) Views generated from the model.



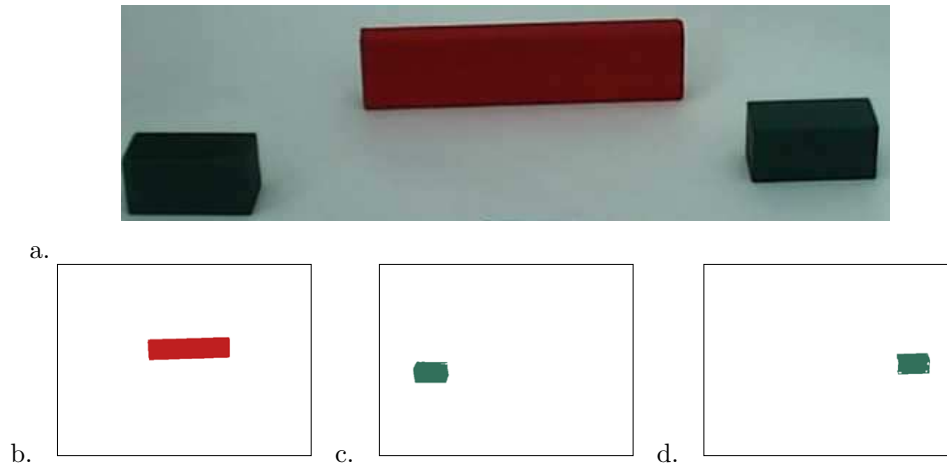
**Figure 8.** The model database is organized hierarchically.

2. For each block, a 3D model is created based on manual measurements of the block image and taking into consideration the polyhedral and symmetric characteristics of the objects. The block models are wireframes given as a list of 3D vertices and a list of the connecting edges (see Figure 7b).
3. For each object, a database of images of the object under various views is created. The wireframe model of the block is used to automatically generate different views of the block (Figure 7c-d). All block images in the database are normalized to a standard size.
4. The model database is organized hierarchically for efficient access to the different models and to the different views within each model (see Figure 8).

### 3.1.2. Generating the VOPD

Once the model database has been constructed the automatic VOPD generator can be applied. Given an input image from the Block World class, its VOPD representation is automatically created. The following steps are involved in this process:

1. Segmentation - automatic segmentation of the objects in the image is performed. Color based segmentation was chosen as the initial step followed by connected component detection and noise cleaning (small components are rejected). Although edge based segmentation was initially incorporated as well, it was found that color segmentation provides adequate segmentation for the VOPD generation steps. The output of this step is a set of sub images, each containing a single connected component (see Figure 9). Each such component is called a *measured* object.
2. Matching - this is the dominant and most complex step in the VOPD generation process. It must be performed efficiently with a high success rate. The suggested procedure considers the database of models, and efficiently prunes irrelevant models. The process then searches for the views of the relevant models that best match the measured object. Matching is performed sequentially for each measured object. Prior to matching, each object is normalized similar to the procedure performed for the database models. Matching progresses according to the following steps:
  - Prune all irrelevant models - at this stage pruning is performed based on color content. Only models with color similar to the measured objects are maintained.
  - Calculate three measurements over the relevant models. Several similarity measurements were considered. It was found that the following three measures supply adequate values for matching:
    - Correlation - pattern matching based on correlation between the measured image and the model image provides a measure of overlap which serves as a measure of similarity.



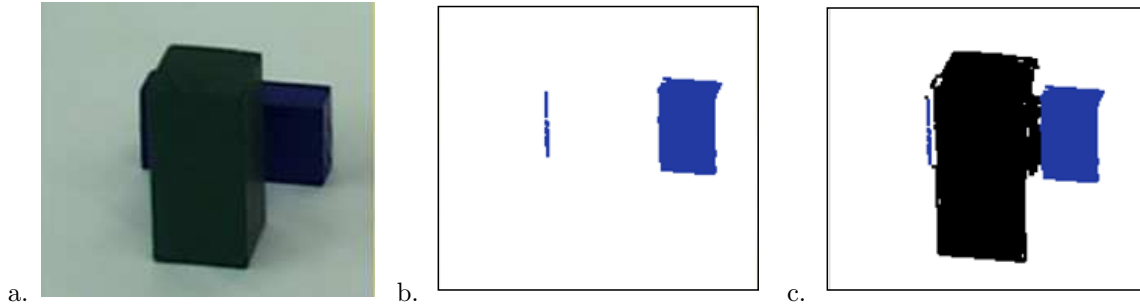
**Figure 9.** Input image is segmented.

a) Input image b-d) Segmented sub-images containing a single *measured* object.

- Number of pixels - comparison between the number of pixels in the measured object and the number of pixels in the model object serves as an additional measure of similarity. Although this measure often correlates with the first measure, there are instances where the number of pixels provides additional ranking of similarity and quality of match.
  - Relative size - since all objects (both model and measured) are normalized, the relative size is not taken into account in the first two measures. Comparison of the x-scaling and y-scaling applied to the objects during normalizing serves as an additional similarity measure.
  - Calculate the overall fitness. The three measures are combined into a single overall value of similarity between the measured object and a model object. The value is defined as the weighted sum of the three similarity measures.
  - For each measured object, the highest ranking models in terms of similarity are given as output. For example, considering the measured object shown in Figure 9b, the most similar model (and model view) was found to be that shown in Figure 10.
3. Object properties and Object relationships - Each measured object has been associated with a database model and a specific view. To create a VOPD of the image each object must be associated with a set of attributes. Additionally, inter-relationships between objects must be incorporated into the VOPD. In this project the inter-relationships are limited to positional relationships (above, below, behind, right of and left of). In this stage the properties of the objects and their inter-relationships are automatically determined:
- The characteristics: color, size, position, name, are easily and automatically extracted from the measured images. For the segmented object of Figure 9b, the following attribute list is obtained automatically.  $V(\text{"LargeRed"}) = \{\text{"red"}, \text{"B"}, 0.2, \text{"complete"}\}$
  - Determining the object attributes of truncation and occlusion is a more complex task. Till now the measured objects were segmented from the input image and each object was assumed to be a single



**Figure 10.** Most similar model and model view found for the measured object shown in Figure 9b.



**Figure 11.** Occluded Objects

a) Input image with an occluded object. b) Segmentation is into two components associated with a single object. c) The occluding object is 'erased' and pixels become "don't care".

connected component. However, for occluded objects this is not necessarily true (see for example Figure 11a). This stage of the project is currently being developed. A multi-level approach will be used in which objects which were rated with a high similarity measure to a database model, are assumed to be un-occluded. These will be virtually 'erased' (see Figure 11c). Objects in the measured image having a low similarity rating will be re-matched as follows: Objects with similar characteristics (e.g. color) will be grouped and the matching algorithm (Step 2 above) will be re-applied against the group of objects (see Figure 11b). The 'erased' regions of the image will be considered as 'don't care' regions during the matching process so that no penalty is accumulated due to pixels in these regions. This process will continue iteratively until most objects obtain a high similarity measure. The process of matching occluded objects to the database model will also consider the relative sizes of the occluding and occluded objects to assist in the matching process.

- Given the objects in image, the inter-object relationships are deduced. Using the attributes of "location", "size" and "completeness", an Object Relationship matrix is built which describes the positional relationship between every 2 objects in the scene. An example matrix for the image of Figure 11a is given in Table 1.

	LargeRed	SmallGreen1	SmallGreen2
LargeRed	–	right-behind	left-behind
SmallGreen1	left-front	–	left
SmallGreen2	right-front	right	–

**Table 1.** Object Relationship matrix for the image of Figure 11a

Following these steps a VOPD can be generated. Figure 12 shows the VOPD for the image in Figure 9a. The tools and techniques used in this Section can be extended to other classes of images. The complexity of images in the class is limited only by the capabilities of Computer Vision.

### 3.2. PictureMaker - A VOPD Interface

The limitation of the previous approach to a specific class of images, requires the VOPD creation system to be developed separately and uniquely for every class. Although producing good performance, this is very restrictive and time consuming. Rather than limiting the database to specific image classes, the approach suggested in this section can deal with *any* image. The goal of creating VOPDs is attained by relaxing the fully automatic characteristic of the system.

In the suggested approach an interface tool called *PictureMaker* is developed to guide the naive user into supplying the necessary details required to create the image VOPD. The interface tool is very intuitive and is



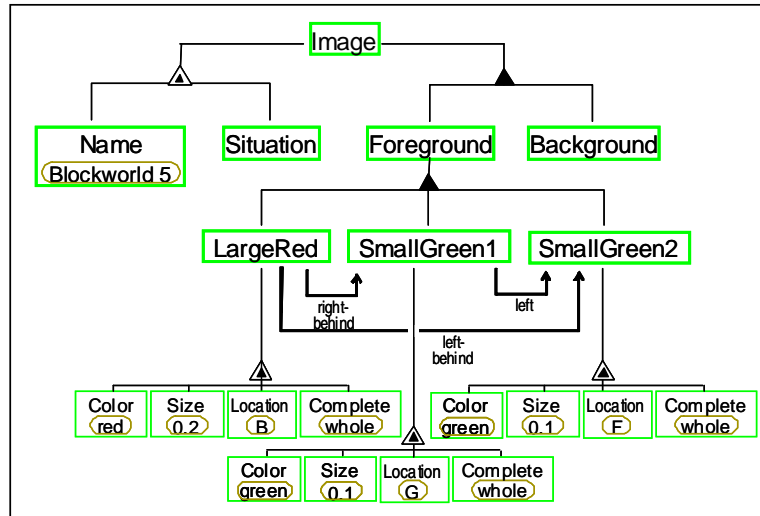


Figure 12. the VOPD for the image in Figure 9a.

based on the simple process of creating an image using visual and graphic tools. The PictureMaker is based on objects being drawn in layers to create a schematic "drawing" of the original or query image.

To create a VOPD of an image, objects in the image, their properties and their relationships must be represented. Thus, the basic entities in the PictureMaker are objects. The VOPD is an hierarchical tree structure, accordingly the PictureMaker imposes an ordering and a hierarchy of the objects being invoked. The underlying principle of the PictureMaker is the notion of objects being positioned in *layers* (see Figure 13). The further back the layer, the closer the associated object is to the background. The notion of layers allows easy distinction between foreground and background (which is the basic structure of the VOPD). It also allows simple computation of positional relationships and occlusion relationships between objects in the image.

Figure 14 shows an example image for which a VOPD must be created. The objects in the scene can be described as an ordered sequence of object layers. These are represented as an ordered list of object names with a special delimiter distinguishing between the objects in the background and those in the foreground of the scene.

This ordering of objects, and the distinction between foreground and background is expressed naturally in a VOPD. The PictureMaker exploits this characteristic to create the VOPD. Allowing the object layers to be grouped and allowing layers to be "a part of" other layers, provides the hierarchical nature of the image description which is an inherent property of VOPDs. Thus for the example of Figure 14, objects "window" and

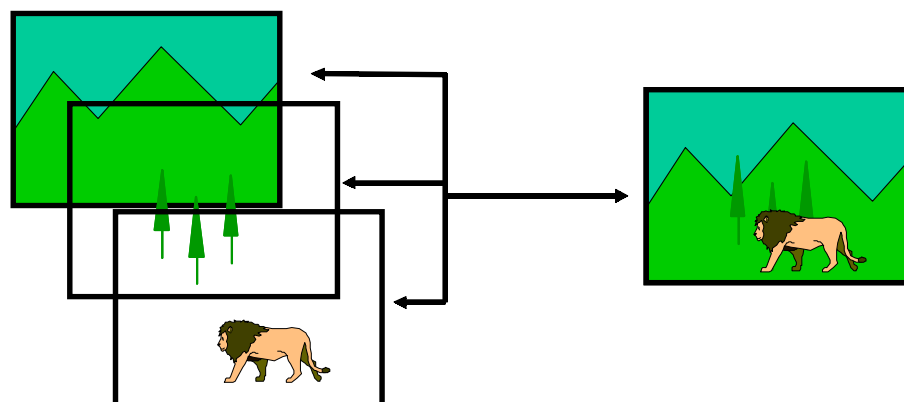
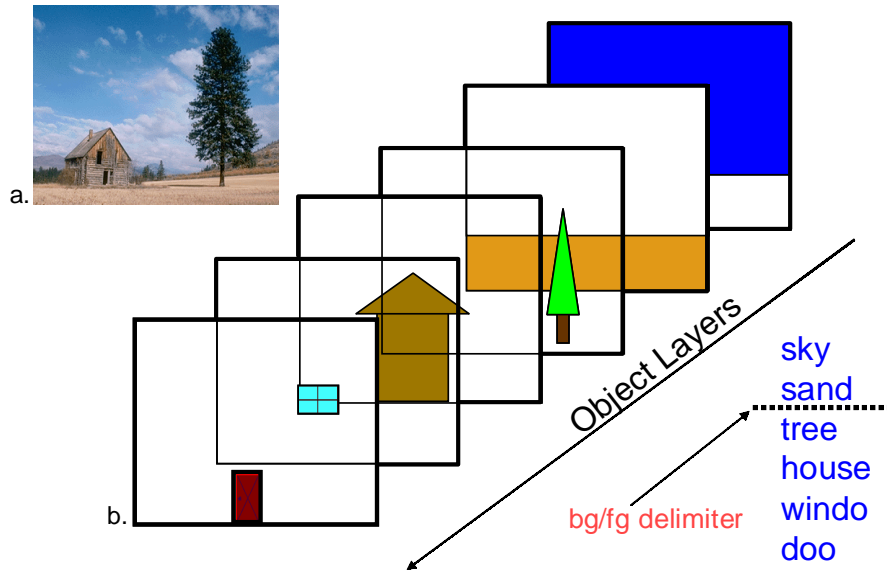


Figure 13. The main concept of PictureMaker is the layering of objects in the scene.



**Figure 14.** An image for which a VOPD is created (top-left) is described as an ordered sequence of object layers (middle). The layers can be given as a list of objects with a special delimiter distinguishing between the background and the foreground (bottom-right).

”door” should be defined as a ”part of” object ”house” and as such will appear as descendants of ”house” in the VOPD (see Figure 15).

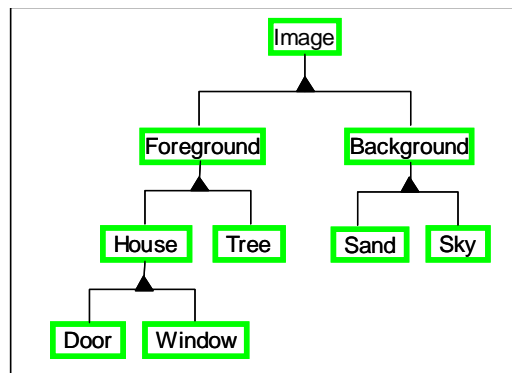
### 3.2.1. PictureMaker - the Application

The PictureMaker interface to creating VOPDs was developed in Java so that it may run and operate on any platform, and specifically on web compatible platforms. Description, technical details and working instructions can be found in.<sup>5</sup>

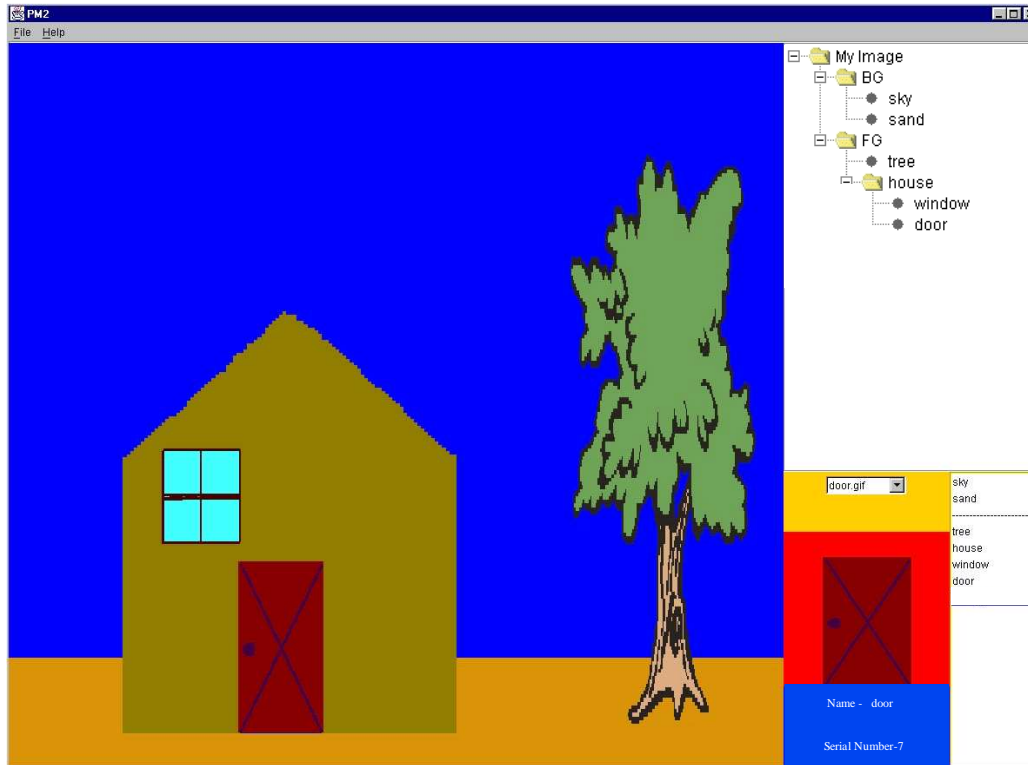
Figure 16 shows a screen shot of the PictureMaker application. As can be seen the user has created an iconized version of an image (in this case, the image of Figure 14).

Figure 17 shows the layout of the PictureMaker application. The application is divided into several frames:

- *Main Screen Frame* - This is the frame in which the user ”draws” the scene using icons. This frame allows the user to add, delete, move and resize object icons.



**Figure 15.** The VOPD for the image of Figure 14.



**Figure 16.** A screen shot of the PictureMaker application.

- *VOPD Tree Frame* - This frame displays a tree diagram representing the VOPD that is automatically created from the user input.
- *Layers frame* - This frame displays the object layers and their order. The object layers are depicted as a list of the object names. The dashed line represents the background-foreground delimiter. All layers in the background are listed above the delimiter and all foreground layers are listed below.
- *Image Viewer Frame* - This frame shows the current selected icon.

The application is object-based. An object that is created during application run time appears simultaneously in the Main Screen Frame, the Layers Frame and the VOPD Tree Frame. Thus the PictureMaker creates the layers and the VOPD tree interactively, as the user adds objects to the scene. An object may be selected in any one of the 4 frames however the manipulations allowed on the object are determined by the frame. In the Image Viewer Frame, the user selects an object from the menu containing a predefined list of objects. Following selection from the menu, the object icon appears in the frame. The object icon appears in the Main Screen Frame when it is dragged from the Image Viewer Frame. Within the Main Screen Frame, the user positions and resizes the icon. An icon may also be deleted. In the Layers frame an object layer may be moved up or down the list. This includes the foreground-background delimiter which may be moved along the list as well. In the VOPD Tree Frame an object may be selected and associated with other objects in the tree. Associations include grouping, becoming "a part of", or creating a relation between two objects.

Hidden from view, the application also automatically calculates object attributes including "size", "location" and "completeness". These are determined from the icon layout in the Main Screen Frame. The attribute "color" and "importance" are set to default values and may be interactively changed by the user. Object attributes may be viewed at user's request.

The application allows the user to save the data and reload it at a later time. The output of the application provides data representing an image VOPD or a query VOPD.

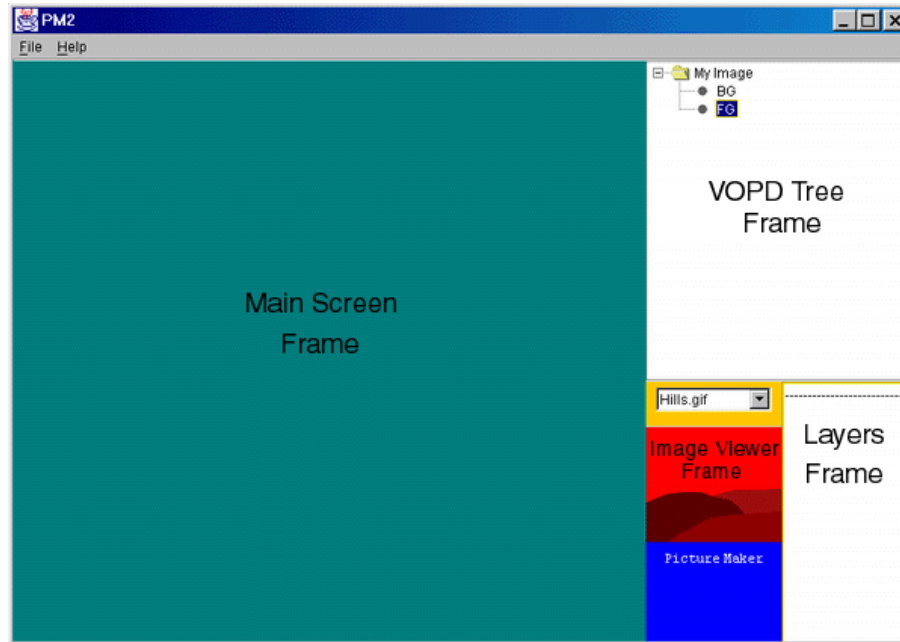


Figure 17. General layout of the PictureMaker application.

Currently, the PictureMaker is a stand-alone application. In the future it will be part of the full Semantic Content Based Image Retrieval system.

#### 4. SUMMARY AND DISCUSSION

A major drawback of the previously defined semantic content based image retrieval system is the need to create a VOPD representation for each database image. This implies that the user must be acquainted with the rules and principles of the VOPD. To overcome the demands on the user of learning the rules and structure of the VOPD, we presented two approaches to provide an interface into the Image Retrieval system and to bypass the need of manually creating VOPD representations. This idea can and should be generalized to other image retrieval applications in which there is a wide gap between human performance and automatic computer capabilities.

#### ACKNOWLEDGMENTS

The authors wish to thank Ohad Matitya, Uri Ben-Zeev, Ami Ben Shushan, Idan Plonsky for their contribution in implementing several stages of the project. This work was funded by the Israeli Ministry of Science.

#### REFERENCES

1. D. Dori and H. Hel-Or, "Semantic content based image retrieval using object-process diagrams," in *7th International workshop on Structural and Syntactic Pattern Recognition SSPR98*, pp. 15–30, 1998.
2. D. Dori and H. Hel-Or, "Semantic content-based image retrieval using object-process diagrams," *Advances in Pattern Recognition, Lecture Notes in Computer Science* **1451**, pp. 230–241, 1998.
3. D. Dori, "Representing pattern recognition-embedded systems through object-process diagrams: the case of the machine drawing understanding system," *Pattern Recognition Letters* **16**(4), pp. 377–384, 1995.
4. D. Dori, "Cognitive image retrieval," in *15th International Conference on Pattern Recognition, 1 - Computer Vision and Image Analysis*, pp. 42–45, 2000.
5. I. Plonsky and A. Ben-Shushan, "Picturemaker." May be downloaded from <http://cs.haifa.ac.il/hagit/papers/PictureMaker.doc>.