A Grammar for the Specification of Forensic Image Mining Searches

Ross Brown
Faculty of IT, Queensland University of Technology,
GPO Box 2434 Brisbane
AUSTRALIA
r.brown@qut.edu.au

Binh Pham
Faculty of IT, Queensland University of Technology,
GPO Box 2434 Brisbane
AUSTRALIA
b.pham@qut.edu.au

Olivier De Vel
Information Networks Division,
Defense Science and Technology Organisation,
PO Box 1500 Edinburgh
AUSTRALIA
Olivier.DeVel@dsto.defence.gov.au

Abstract
There has been much research into the use of similarity measures to facilitate content-based image retrieval. However, there are other application areas where the user will wish to retrieve images that contain objects in specified arrangements. This has particular application in the area of Image Forensics, where legal investigations require the ability to perform search queries on images containing suspicious objects in relevant spatial organisations. In this paper we present a grammar which augments a body detection system by allowing arbitrarily arranged detectors.

Keywords
Intelligent Image Mining, Pattern Recognition, Image processing, Support Vector Machines

INTRODUCTION

Computer forensics is the application of computer analysis techniques to determine potential legal evidence of computer crimes or misuse that are caused by unauthorised users or by unauthorised activities generated by authorised users. It covers a wide range of applications such as law enforcement, fraud investigation, theft or destruction of intellectual property. Techniques used for such investigations are varied and may include data mining and analysis, timeline correlation, information hiding analysis, etc. Data for evidence are of various types and come from different sources, e.g. storage devices, networks, etc. Since multimedia format is widely used and readily available via the Internet, there are increasing criminal activities in the last few years, which involve the transmission and usage of inappropriate material in this format. Hence, much forensic evidence comes in the form of images or videos which contain objects and/or scenes that may be related to criminal behaviours. A typical investigation in computer forensics can generate large image and video data sets. For example, a disk can easily store several thousands of images and videos in normal files, browser cache files and unallocated space (i.e., non-file system areas on the disk which may contain fragments of files). This can make the task of searching for, and retrieving, images/videos very time consuming. Digital image forensics efficiently seeks for evidence by using appropriate techniques based on image analysis, retrieval and mining. The use of such techniques for investigative purposes have only recently emerged, although they have been intensively researched over the last three decades for many other important applications: medical diagnosis, mineral exploration, environmental monitoring and planning, aerial surveillance, etc.

Image mining is a specific area of image analysis in which images of certain characteristics are detected and retrieved from a large set of images. The goal of image mining is to combine image retrieval techniques with the ability to learn and model specific objects of interest found in an image. Earlier approaches to image retrieval rely on the retrieval of associated text strings which provide some descriptions of the images such as name, place, date and annotation. Content-based approaches which came much later allow searches based on some general low-level visual features such as colour, shape, texture e.g. [1]. Search-by-example is a common practice whereby an image is supplied and the system would return images which have features similar to those of the supplied image. The similarity of images is determined by the values of similarity measures which are specifically defined for each feature according to their physical meaning. For example, a similarity measure for colour can be defined as the sum of square of the difference in red, green and blue components. Retrieved images can be ranked according to these similarity measures. Users are allowed to select specific features and their weights to add subjective bias based on preferences or previous experience. Since the quality of the retrieval results relies on the choice of features and their similarity measures, much research has been focused on identifying features with strong discriminatory power and similarity measures which are meaningful and useful. In addition, we would ideally want a more "intelligent" system which can include high-level knowledge, deal with incomplete and/or uncertain information, and learn from previous experience. Such systems could include, for example:

- methods that develop a model of each object to be recognised (called *model-based* methods). These objects are classified using their constituent components that in turn are characterised in terms of their primitives,

- methods that use statistical techniques to assign semantic classes to different regions/objects of an image (called *statistical modelling* methods), and
- methods that require user feedback to drive and refine the retrieval process (called *user relevance feedback* methods). The system is thus able to derive improved rules from the feedback and consequently generate better semantic classes of images [2].

Image mining in computer forensics would ideally use a combination or hybridization of these methods. For example, the inherently interactive and continually changing nature of a forensic investigation would favour a user relevance feedback approach together with either a model-based or statistical approach that would capture knowledge about the objects in the image and better retrieve images with specific types of objects (rather than just similar images).

For general image forensics, investigators often identify certain objects or scenes in an image which might suggest a criminal activity by their co-occurrence or relationships with each other. For example, the presence of firearms and maps of an important building might suggest a potential armed hold-up. Thus, our aim is to look for a generic method of image mining which is capable of detecting objects and/or scenes that are made up of components, where components can be nominated by investigators. These components can also be constrained by spatial or non-spatial relationships and might be deformed by various standard transformations. We also wish to provide a system that can be trained by examples, and furthermore can be iteratively improved by using relevance feedback supplied by investigators on retrieval results.

We now elucidate a prototype mining system [3], in order to provide a basis for the later grammar which we use to describe scenes in more complex search queries. The prototype system is for detecting partially clad humans in swimwear.

PROTOTYPE SYSTEM OVERVIEW

The system is composed of two main parts: training and querying. The training process involves the setting up of object models which includes the specification of the parameters for image features used by the component classifier, and the constraints on the position and orientation of the components to make up an object. The model trainer who performs this task should have some understanding of the principles underlying the classifiers as well as insight into the types of objects required for forensic evidence. The system performs segmentation on a sample set of images (called the training set) to obtain: image patches which contain those components of interest; the feature parameters of these components (e.g. arm, leg); and their position and orientation constraints (e.g. above, behind). These will be used as input to train the classifier. The training process gradually refines the models through the modification of parameters and the addition of false positives and false negatives resulting from test runs. Once the models have been specified and the classifier has been trained, the query operator can set up a query based on the components of interest and the constraints to be placed on them to form an object of interest. The system will then perform the classification and return those images that contain the object of interest.

Image Component Detectors

We achieve improvements on the performance of image component detectors in a number of aspects. The detectors are sensitive over the problem of scaling caused by relative size of the image, object and object components. We also find that the detection accuracy is improved when HSV colour space is used instead of RGB. In particular, YCbCr colour space gives better discrimination for human skin tone. Mohan et al. [4] dealt with the translation of objects and their components by traversing the wavelet decomposition space to crop certain regions of coefficients used for classification. We deal with in-plane rotation by rotating the wavelet space coefficients to search for a match. We find that the out-of-plane rotation does not cause a problem if fairly coarse scale coefficients are used. We use low coefficients in addition to high wavelet coefficients in order to add more discrimination for areas of low contrast.

Hierarchical Classifier

The component detectors are integrated into a hierarchical system in the following way. The patches that contain object components are constrained to locations based on knowledge obtained from the training sets. The results of the component detectors at constraint locations are then fed into a trained SVM for the whole object. While a quadratic SVM is used for component detectors, a linear SVM is used for the object detector because of its robustness for partial detection.

Feature Vectors

Each vector is made up of two main groups of information edge coefficients defining the outline of body parts and regions defining areas of continuous tones. The number of entries for each vector depends on the number of coefficients making up the two components of the component detector. Therefore, the feature vector varies for each component detector generated. Each vector thus contains two levels of the wavelet pyramid at the 16x16 and 32x32 pixel coefficients – except in the case of faces which are 8x8 and 16x16 due to the smaller patch size. That is, the region and edge information is repeated for each level within the final feature vector.

Edge Coefficients

Figure 1 shows the high threshold coefficients for the ensemble images (white squares are above an arbitrary threshold set to obtain a nice outline). These represent consistently high Haar wavelet coefficient values throughout the training set of pelvis images. The image at the bot-

tom is the vertical wavelet coefficients, while the image at the top is the horizontal wavelet coefficient image.

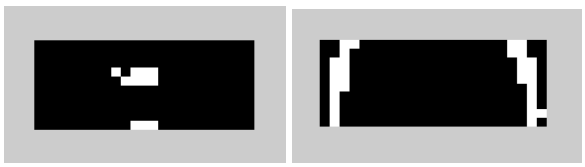

Figure 1 Horizontal and vertical wavelet coefficients for a pelvis.

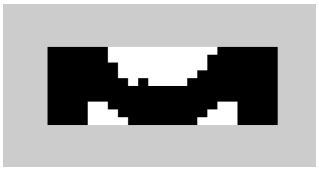

Figure 2 Threshold image of a pelvis.

Each of the white coefficients forms a variable in the feature vector. The maximum of the YCbCr components is written out from the actual image being processed. Therefore, in short, the white squares are the most important coefficients in the process of recognising the *outline* of a pelvis with a bikini. So the first component of each feature vector is a list of the most important coefficients from the vertical and horizontal wavelet coefficients.

Region Information

Figure 2 illustrates the threshold images from the same ensemble image set. Here, the white regions indicate coefficients below an arbitrary threshold. Therefore, these are regions of consistent colour across all the images in the training set. For the pelvis example, this indicates the presence of regions of skin that identify the shape of a pelvis.

Figure 3 illustrates an example image found by the detector. The white rectangles indicate the discovery of a body component (pelvis, torso, face). The results of the component detectors are then assessed by a whole body detector.

SEARCH DESCRIPTION GRAMMAR

From the basic system described, a grammar can be constructed to enable arbitrary scene descriptions to be devised for search queries. This grammar allows the search to be specified as a hierarchy of detectors, working at different structural resolutions. The grammar is made up of component detectors and object detectors and their related spatial relationships and position data.

Object detectors and component detectors are an abstraction of the detection mechanisms used to find components. The object detectors themselves are hierarchies of other component detectors and/or other objects. An entire scene description can be used in another object detection scene, and so on. The hierarchies can be at varying levels of precision. The high layers may implement rough detectors to find regions of interest for the lower levels to apply finer level detectors in a multi-resolution manner. Thus, the grammar allows the encapsulating system to store the resolution of the search at either a broad structural level (e.g. skin detection) down to fine grain informational detection (e.g. face, pelvis, torso).

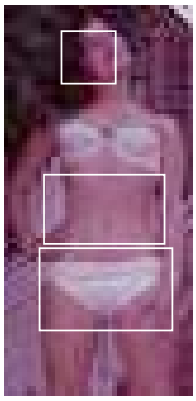

Figure 3 Example illustrating the locations of detectors on a possible positive image.

Relationships encapsulate relative spatial arrangements between the object detectors within the query description. This is specified in a multi-precision manner: from rough linguistic terms like up, down, above, below; to more precise terms like north, south, east, west; and then to orientation and absolute position specifications. Each of the absolute position measurements is specified in normalised image-space coordinates [0..1]. The orientation angle, specified in degrees relative to the horizontal axis, is also stored within each node. Each of these values have intervals attached which allow for deviation from the original positions. These terms, values and deviations facilitate the complete 2D specification of the arrangements of the detectors at levels of resolution relevant to the search task.

We describe the grammar in two sections. One for static scenes, and one for dynamic scenes. The static scene grammar will be implemented in full, to provide support for image query specifications across many application areas.

Static Scenes

The following diagram and text outlines the grammar for static search scenes. Figure 4 illustrates the n-ary tree data structure [5] for an example detector finding the co occurrence of human bodies and tanks in images. The human detector has been implemented, while the tank detector is hypothetical, but based upon the work developed for the partially-clad human body detector.

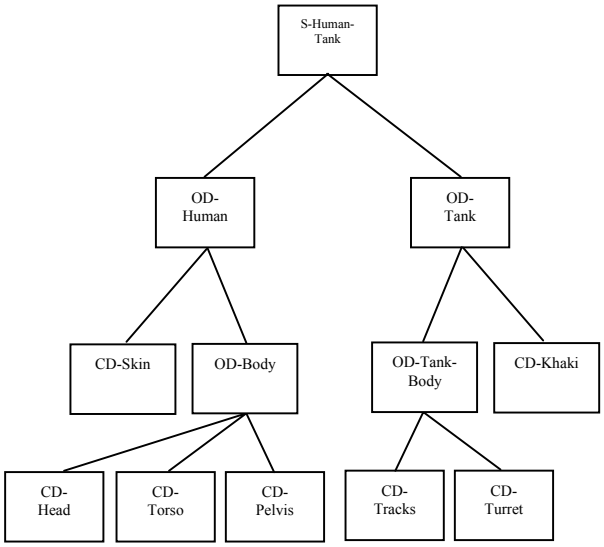

Figure 4 Diagram illustrating the hierarchical nature of the search grammar.

As can be seen in the diagram, the image mining query is described as a scene (marked S)to be searched for using a number of object detectors (marked OD) and component detectors (marked CD). Each node contains the relevant information to fully specify the spatial attributes of the object or component detector. Note that an object detector can be a collection of components and/or other object detectors, giving complete freedom to use predefined scenes as object detectors in other scenes. The object and component detector attributes are listed in Table 1 and Table 2.

Table 1 Table listing of Component Detector attributes

Component Detector Attributes	*Description*	*Constraints*
Comp-Detector-ID	Identification string for component detector.	Unstructured string of characters.
Comp-Detector-Loc	Directory location for detector.	Unstructured string of characters.
Displacement	2D motion displacement vector for detector from previous frame.	2 floats [0.0..1.0]
Orientation	Angle in degrees that the detector is rotated relative to vertical axis.	Float [0.0..360.0)
Relation-List	List of relationships to other objects/components in the scene.	Array

Table 2. Table listing of Object Detector attributes

Object Detector Attributes	*Description*	*Constraints*
Object-Detector-ID	Identification string for component detector.	Unstructured string of characters.
Object-Detector-Loc	Directory location for detector.	Unstructured string of characters.
Displacement	2D motion displacement vector for detector from previous frame.	2 floats [0.0..1.0]
Orientation	Angle in degrees that the detector is rotated relative to vertical axis.	Float [0.0..360.0)
Relation-List	List of relationships to other objects/components in the scene.	Array
Detector-List	List of object/component detectors making up the object detector.	Array must contain at least one component detector.

File Grammar

This hierarchy has been encapsulated into a file grammar to support the storage and manipulation of the data structure for future use by a user. This has been described using the following grammar, semi compatible with lex and yacc [6]. Comments are inserted to illustrate more obscure constructs. Some of the more atomic definitions have been elided in order to save space, e.g. the definitions for *free-form strings*.

```
Forensic-Scene:
    Scene
        Scene-Detector-ID
        Comp-Detector-ID
    End-Scene
    Scene
        Scene-Detector-ID
        Object-Detector-ID
    End-Scene

Comp-Detector:
    Component
        Comp-Detector-ID
        Comp-Detector-Loc
        Displacement_opt
        Orientation_opt
        Relation-List_opt
    End-Component
```

```
Object-Detector:
    Object
        Object-Detector-ID
        Object-Detector-Loc
        Displacementopt
        Orientationopt
        Relation-Listopt
        Detector-List
    End-Object

Detector-List:
    Detector-List, Gen-Detector-ID

Gen-Detector-ID: one of
    Object-Detector-ID, Comp-Detector-ID, Scene-Detector-ID

Scene-ID:
    S Freeform-String

Comp-Detector-ID:
    CD Freeform-String

Object-Detector-ID:
    OD Freeform-String

Comment:
# Freeform-String

Relation-List:
    Relation-List, Relation

Displacement:
    (Integer Integer)
    Compass-Dir
    Ling-Term

Relation:
    Compass-Dir Object-Detector-ID
    Abs-Pos
    Ling-Term Object-Detector-ID

# These compass terms cover a number of
# direction specifications, e.g. North and North West.

Compass-Dir:
    Compass-Term
    Compass-Term Compass-Term

Ling-Term: one of
    Above Below Left Right Up Down
```

The following lists the file format generated for the example scene detector in Figure 4. Note how the detectors are defined once and then referred to with an ID number.

```
Scene S-Human-Tank
    OD-Human
    OD-Tank
End-Scene

Object OD-Human
    CD-Skin
    OD-Body
End-Object

Object OD-Body
    /usr/CFIT/objects/body.svm
    CD-Head
    CD-Torso
    CD-Pelvis
End-Object

# The directory locations are just arbitrary
# examples, and are not representative of
# actual detectors.

Component
    CD-Skin
    /usr/CFIT/objects/skin.exe
End-Component

Component
    CD-Head-Detector
    /usr/CFIT/objects/head.svm
    90
    Above CD-Torso-Detector
End-Component

Component
    CD-Torso-Detector
    /usr/CFIT/objects/torso.svm
    90
    Above CD-Pelvis-Detector
End-Component

Component
    CD-Pelvis-Detector
    /usr/CFIT/objects/pelvis.svm
    90
End-Component

Object OD-Tank
    CD-Khaki
    OD-Tank-Body End-Object  Object OD-Tank-Body
    CD-Tracks
    CD-Turret
End-Object
# Tank components are defined similarly to the body detector.
```

Dynamic Scene Grammar

The previous static image search query grammar requires augmentation to model changes in the structure and relationships of the detectors when searching motion videos. The grammar needs to model the following dynamic effects:

- appearance/disappearance of objects – implemented via insertion and deletion operations on the n-ary tree;
- changes in the attributes of objects – implemented via searching for the appropriate tree node and then modifying contents;
- changes in spatial relationships between objects – implemented via the attribute modification operation;
- changes in the structure of the tree hierarchy – implemented as a deletion and insertion of sub-trees within the n-ary tree.

The appearance of a new object is handled by insertion of the new object detector into the hierarchy. This will facilitate the tracking of objects in motion scenes. Similarly, the disappearance of an object is handled by the deletion of a node from the n-ary tree.

Changes in the attributes of the objects include transformations such as: scale, shear, translation, rotation, and related modifications to linguistic spatial relationships to other objects and occlusion effects. This is accommodated by searching the n-ary tree for the object detectors in question, and then modifying the tree node contents. An additional displacement field stores the frame to frame 2D displacement of the detector in question.

Changes in the n-ary tree organisation, which may or may not indicate changes in the actual scene being searched for, are handled by standard sub-tree deletion and reinsertion operations. This may occur due to user modification of the search specification upon receipt of search results in a *user in the loop* feedback query scenario.

Each node can also have an additional co occurrence field to record the number of occurrences of the object within a scene. This allows the setting up of queries which seek to find more than one occurrence of the object in the scene without specifying the actual number of objects.

In the case of objects which become occluded in the scene, a visibility flag is used to indicate occlusion of an object, and not disappearance. Occlusion occurs when the detectors tracking objects in the scene overlap other detectors and one of the objects disappears.

The adaptive features of the data structure can be used to provide data mining facilities for the system. In particular, the number of occurrences of objects within an image database can be used to search for patterns that show likely relationships. The search engine may not specify a structure, but may contain a number of detectors which find occurrences of the objects and simply report back their number and spatial locations for use by rule induction techniques.

CONCLUSION

We have described a grammar for the specification of forensic image mining queries that facilitates the specification of arbitrary scenes using different detection systems at differing structural resolutions. Examples have been shown for the grammar that illustrates its power in describing a search query. This grammar will be implemented for the static scenes and in the future will be modified to accommodate motion scenes in videos. The adaptive nature of the search query grammar allows for other data mining applications, which may infer rules from searches of the image database.

We expect the applications of this grammar to be in the law enforcement, homeland security and web search facility areas. In general, the grammar can be used where a search specification requires a particular spatial arrangement of objects.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the assistance of the Australian Defence Science and Technology Organisation in the development of this project.

REFERENCES

1. Niblack, W., et al. *Updates to the QBIC System*. in *Storage and Retrieval for Image and Video Databases*. 1997.
2. Muller, H., et al. *Strategies for positive and negative relevance feedback in image retrieval*. in *Proc. International Conference on Pattern Recognition ICPR2000.* 2000.
3. Brown, R., B. Pham, and O. de Vel, *Image Mining for Computer Forensics*. Submitted to IEEE Security and Privacy, 2003.
4. Mohan, A., C. Papageorgiou, and T. Poggio, *Example-Based Object Detection in Images by Components.* IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2001. **23**(4): p. 349-361.
5. Gonzalez, R. and M. Thomason, *Syntactic pattern recognition : an introduction*. 1978, Reading, USA: Addison-Wesley.
6. Kernighan, B. and D. Ritchie, *The C Programming Language*. 2nd ed. 1988, Murray Hill: Prentice Hall.