Satellite Image Processing with Biologically-Inspired Computational Methods and Visual Attention

Md Ibne Sina

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of
the requirements for the degree of

Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering and Computer Science
University of Ottawa

May 2012

Abstract

The human vision system is generally recognized as being superior to all known artificial vision systems. Visual attention, among many processes that are related to human vision, is responsible for identifying relevant regions in a scene for further processing. In most cases, analyzing an entire scene is unnecessary and inevitably time consuming. Hence considering visual attention might be advantageous. A subfield of computer vision where this particular functionality is computationally emulated has been shown to retain high potential in solving real world vision problems effectively. In this monograph, elements of visual attention are explored and algorithms are proposed that exploit such elements in order to enhance image understanding capabilities. Satellite images are given special attention due to their practical relevance, inherent complexity in terms of image contents, and their resolution. Processing such large-size images using visual attention can be very helpful since one can first identify relevant regions and deploy further detailed analysis in those regions only.

Bottom-up features, which are directly derived from the scene contents, are at the core of visual attention and help identify salient image regions. In the literature, the use of intensity, orientation and color as dominant features to compute bottom-up attention is ubiquitous. The effects of incorporating an entropy feature on top of the above mentioned ones are also studied. This investigation demonstrates that such integration makes visual attention more sensitive to fine details and hence retains the potential to be exploited in a suitable context. One interesting application of bottom-up attention, which is also examined in this work, is that of image segmentation. Since low salient regions generally correspond to homogenously textured regions in the input image; a model can therefore be learned from a homogenous region and used to group similar textures existing in other image regions. Experimentation demonstrates that the proposed method produces realistic segmentation on satellite images.

Top-down attention, on the other hand, is influenced by the observer's current states such as knowledge, goal, and expectation. It can be exploited to locate target objects depending on various features, and increases search or recognition efficiency by concentrating on the relevant image regions only. This technique is very helpful in processing large images such as satellite images. A novel algorithm for computing top-down attention is proposed which is able to learn and quantify important bottom-up

features from a set of training images and enhances such features in a test image in order to localize objects having similar features. An object recognition technique is then deployed that extracts potential target objects from the computed top-down attention map and attempts to recognize them. An object descriptor is formed based on physical appearance and uses both texture and shape information. This combination is shown to be especially useful in the object recognition phase. The proposed texture descriptor is based on Legendre moments computed on local binary patterns, while shape is described using Hu moment invariants.

Several tools and techniques such as different types of moments of functions, and combinations of different measures have been applied for the purpose of experimentations. The developed algorithms are generalized, efficient and effective, and have the potential to be deployed for real world problems. A dedicated software testing platform has been designed to facilitate the manipulation of satellite images and support a modular and flexible implementation of computational methods, including various components of visual attention models.

Acknowledgements

First of all, I would like to deeply thank my advisor, Prof. Pierre Payeur, whose constant guidance, support, inspiration, and encouragement have made this thesis possible. I am truly indebted to him for enabling me to carry on this work. I have not only learned the subject matter from him but also have acquired other invaluable skills such as scientific thinking, critical judgment and conscious writing.

I am sincerely thankful to Prof. Jochen Lang and Prof. Eric Dubois for all the knowledge that I acquired from attending their sessions. I would also like to express my gratitude to Dr. Ana-Maria Cretu for her sincere cooperation. She was always available when I needed to discuss something. My special thanks to Dr. Alberto Chávez-Aragón for all those interesting discussions on diverse types of topics.

I owe my deepest gratitude to my lovely and loving wife, Shaela Khan. Her constant presence and love inspired me in every way. I specially acknowledge my parents to whom I owe my entire life. I also acknowledge all my friends and lab mates for keeping me cheered up.

List of Figures

List of Tables

List of Acronyms

BGR	Blue-Green-Red
FIT	Feature Integration Theory
FOA	Focus Of Attention
FSD	Filter Subtract Decimate
HMM	Hidden Markov Model
HSV	Hue-Saturation-Value
LBP	Local Binary Patterns
LPF	Low Pass Filter
LSE	Least Squared Error
LSR	Least Salient Region
MSR	Most Salient Region
NVT	Neuromorphic Vision Toolkit
PDF	Probability Density Function
RGB	Red-Green-Blue
ROI	Region Of Interest
SVD	Singular Value Decomposition

1 Introduction

1.1 Context

Several problems in the domain of digital image understanding have long been eluding researchers. Numerous classical algorithms that attempt to solve those problems depend on various assumptions either explicitly or implicitly. Moreover, many algorithms oversimplify the reality and often ignore the possibilities of the presence of other objects in input images. These assumptions and over simplifications lead the algorithms to poor performance on real life complex images that may encompass several types of objects having different poses at various scales, arbitrary shapes, shadows, etc. On the other hand, human beings are extremely efficient at accurately processing huge amounts of visual data. Our visual system accomplishes the task of image understanding at every moment with great reliability. We can identify an object almost instantly, irrespective of its pose, scale, the presence of other objects, or partial occlusion. While a good understanding of the human visual system would be of great help to implement the perfect artificial vision system overcoming the limitations of classical approaches, human vision is still not fully understood. However, inspiration can be taken from the continuous efforts of researchers in the fields of neuroscience and cognitive science who enrich the understanding about the inner workings of the human visual system [1][2]. This work investigates different models of the human vision system and experiments on their possible applications in image processing and understanding. The incorporation of classical image processing techniques along with biological vision elements is explored.

Human vision incorporates several mechanisms in order to support survival. Visual attention is one of the many processes involved in biological vision systems among many different layers of processing that take place in human vision. Visual attention is responsible for identifying important or interesting regions in a scene and for discarding other unimportant or non-interesting parts and thus saving valuable resources and time. Further investigations such as recognition processes are concentrated within those regions of interest. Here “important” or “interesting” regions can also refer to objects of interest. This mechanism greatly helps in efficient scene analysis and hence faster response time. There exist computational models for visual attention that try to emulate such a biological behavior [3][4]. This approach is well suited for high resolution images, where one can direct attention to the relevant regions of the input image and process them for recognition. This strategy can lead to faster processing

by quickly discarding irrelevant image regions. However, the remaining challenge is to identify and classify the “important” image regions. In that respect, goal-directed search is essential. This issue is therefore further investigated in this monograph.

In general, digital image understanding can be used for diverse practical applications that include product manufacturing, automated vehicle navigation, security and monitoring, military defense, computer gaming, medical and surgical procedures. More specifically, one very interesting and challenging application is satellite image processing, which is widely used for earth monitoring, map validation, urban management and development, geographical change detection, agricultural applications, etc. Satellite images and multispectral images are usually different from other natural images in several aspects. They have high spatial resolution, diverse types of image contents, and various image scales, which makes them challenging to deal with when using classical computer vision algorithms. Our recent works in computer vision research show that biologically inspired vision techniques can be used for better performance on satellite images for effective segmentation [5] and target object localization [6]. Throughout this work all experimentations have been conducted on such images due to their practical relevance.

In geo-imaging applications, satellite images contain a complex array of components such as cities, forests, rivers, and desert. Other factors such as cloud, lighting and atmospheric conditions make it even more challenging to work with such images collected in outdoor environments. While the recognition of features characterizing satellite images by a human operator can be quite efficient, provided a given amount of training, the most advanced computational solutions primarily rely on atmospheric and photogrammetric models. Moreover, most of the computational techniques currently used for image feature extraction and classification are generalizations of algorithms which are neither specifically designed for geospatial applications nor fully automated. As a result, the false positive rate of decision is very high and several features of interest remain undetected. According to industry reports, the current algorithms achieve only about 25-30% accuracy when running on large collections of geospatial images, thereby leaving substantial opportunity for new approaches to improve upon the current state-of-the-art techniques in terms of performance. In spite of the promising results of biologically inspired models, following an extensive research in the literature, few papers e.g., [7], was found that applies the concepts of visual attention to aerial images but only shows limited experimental results on a single image.

1.2 Research Objectives

As mentioned earlier, biologically inspired image processing techniques are promising and have the potential of performing better than classical approaches. However, existing computational visual attention systems are still in an embryonic stage and are generally being applied on images with simplistic content and used for simple tasks [3][4][8] and hence opportunities exist to advance them. The primary goal of this research is to explore and enhance existing techniques and to investigate a broader range of applications of computational visual attention.

This research is heavily influenced by the work of Frintrop [4] who attempted to overcome limitations of previous work by Itti *et al*. [3] in the field of computational visual attention, and used the approach for object detection. Frintrop's attention system utilizes three features – intensity, orientation and color, and encourages experimentation with other features as well in order to compute attention maps that are suitable for a given context. In this work, experimentation with an extra entropy feature has been conducted and its effects have been analyzed. It is important to realize that there are several other ways to discover interesting regions within an image that depend on image statistical or functional properties (see [9] for instance); but in this work the focus is placed on biologically-inspired approaches.

Many applications of visual attention have been mentioned in [4] including object tracking, image compression, and image matching. Among them application in image segmentation was experimented with in the present work. In that respect, elements of visual attention are seamlessly integrated with elements from classical image processing techniques that are capable of generating realistic image segmentation.

Visual attention has two denominations – bottom-up and top-down, where the former one is generic in nature and is derived directly from scene contents, and the latter one is more biased towards the objects of interest. Top-down attention is of primary interest, since it has the potential to locate the target objects. However, it remains a great challenge to determine how one can combine the elements of bottom-up attention to obtain an effective top-down attention model that would be able to locate objects of interest. Frintrop also proposed a model for top-down attention in [4], which is highly dependent on background contents of the input image thus making the model unstable on images which were not part of the training set. In this work, this aspect is further investigated and a new technique for top-down attention is proposed. The proposed technique is capable of automatically

identifying important object features from a set of training images and to locate objects having similar features in a given test image.

One of the fascinating characteristics of computational visual attention is its application in the area of object recognition. Although visual attention alone cannot be used to identify objects, when it is combined with some object learning techniques, it can boost the performance of that learning mechanism since only relevant regions are processed. In this work, an object recognition technique has been devised which is coupled with the proposed top-down attention model. With such object recognition ability, it becomes possible to label an input image for pre-learned objects that it contains. Lastly, satellite images are given most importance throughout this work. They will be used to demonstrate the potential of the refined visual attention and object learning techniques, given their inherent large dimension, complexity and practical importance.

In summary, the primary goal of this work is neither to improve nor to devise a new psychological attention model that overrides existing ones. But rather to explore, extend and improve existing computational attention methods and to demonstrate the potential of such attention systems for better image understanding of satellite images. The following aspects represent the specific objectives of this research:

- The definition of, and experimentation with, extra features on top of the ones classically found in the literature in order to compute suitable attention maps in the context of satellite imaging;
- The introduction and experimental evaluation of a new satellite image segmentation technique based on saliency maps;
- The adaptation of top-down attention map computation to enhance goal-directed object search operations;
- The design and validation of an object recognition system for satellite images that exploits the power of visual attention and its incorporation with elements from classical computational methods.

1.3 Structure of the Thesis

In Chapter 2, a comprehensive review of literature is provided. At first, some background information such as terminologies, along with mathematical background about tools that are used throughout the thesis, as well as general applications, are discussed. An overview of the state-of-the-art and detailed descriptions of Frintrop's system (both bottom-up and top-down attention) are provided in sections 2.1, 2.2 and 2.3 respectively. An overview of traditional techniques for processing satellite images is provided in section 2.5. Chapter 3 provides details about the techniques considered in this work and the way they have been expanded and refined. An extensive experimental evaluation of performance is also reported in that chapter. Specific details about the software implementation of the proposed algorithms are covered in Chapter 4 along with an overall description of software design issues and the graphical user interface. Chapter 5 concludes the thesis with a summary of findings, a list of contributions and suggestions for potential future work.

2 Literature Review

2.1 Background

2.1.1 Visual Attention

Millions of years of biological evolution have made our vision system both intricate and effective. The primary reason behind this effectiveness is that human beings pay close attention only to regions of interests in their field of view. Deploying detailed analysis to a small region one at a time and swiftly switching to other regions gives the sense of rich scene understanding and results in a gradual update of the knowledge about one's surroundings [10]. When humans perform their everyday activities such as walking, driving, or playing, they do not need to know or see everything. Only the objects that are related to their current activities are most important. Attending to them one-by-one and interpreting their relations to the scene eventually provides enough information about the surroundings. Visual attention is broadly classified as *Bottom-Up* and *Top-Down* [11] attention which are discussed in subsequent sections. According to [12], two independent and interacting brain areas exist that are responsible for these two types of attention mechanisms.

2.1.2 Bottom-Up Attention

The elements of bottom-up attention are directly related to scene contents and independent of the observer's internal mental state. The bottom-up factors refer to image conspicuity, and include image features such as strong brightness or color contrast. Image conspicuities can be thought of as regions with conspicuously different properties than their respective surroundings, that is, regions that put themselves in evidence. Image regions having such properties are going to attract an observer's attention naturally and are called *salient* regions. For example, if there is large tree in the middle of a desert then the tree is salient and has high potential of attracting the observer's attention. Similarly the full moon over a clear night sky is also highly salient. There exist different computational approaches that attempt to mimic such biological behavior in order to rank the level of conspicuousness of different regions in a scene. The most conspicuous region is believed to be at the focus of attention and a mechanism called *selective attention* guides the order in which the conspicuous regions are attended. One interesting issue about bottom-up attention is that a highly salient region cannot be suppressed voluntarily [13] even when the observer is driven by some other goals. This effect is called *attentional capture*. For example, if an observer is looking for the Andromeda galaxy on a clear night sky with a full moon, it is not possible for the observer to completely ignore the moon. The bottom-up factors that are

considered in this work will be discussed in later sections. Figure 2-1 shows an example of bottom-up map computed by Frintrop's system [4] which will be discussed in depth in section 2.2. Figure 2-1(a) shows an input image that contains few noticeable objects such as the traffic lights and gate, signs saying "Railroad Crossing", and train windows. All interesting objects discovered by Frintrop's system are shown in Figure 2-1(b) where the level of brightness corresponds to the level of saliency. It can be observed that parts of the traffic lights and the traffic signs which are laid on a plain light-blue region (sky), the train windows, the monogram of Caltrain, and the red sections of the traffic gate receive higher saliency. On the other hand, the plain sky and other uniformly textured regions receive lower saliency (depicted by darker shades). Anything which is significantly different from its surroundings has been identified by Frintrop's system.

(a) Input image (b) Bottom-up saliency map

Figure 2-1: (a) An input image and (b) bottom-up saliency map computed by Frintrop's system.

2.1.3 Top-Down Attention

Unlike bottom-up attention which is derived from scene contents, top-down attention is influenced by the current mental state of the observer. The mental state encompasses elements such as current goal, context of the current situation, and expectation. The effects of these elements come from higher brain areas and are influenced by the observer's knowledge [12]. For example, an observer driving a car is better prepared to spot stop signs or traffic lights than other passengers who usually don't pay that much attention to the road. There could be other salient regions in the driver's field of view but the regions containing traffic signs will attract more attention and further analysis will be deployed on those regions, which will lead to eventual realization of safe vehicle navigation. The complexity of the driving

forces of top-down attention such as expectation, knowledge, and context remain the main obstacles to a better understanding of top-down attention; since very little is known about these aforementioned elements. Consequently it is hard to model these driving factors and their interactions, which would be useful in practical applications. In [4], a top-down model is described which will be validated in section 2.3. In this model Frintrop combines the bottom-up features to compute a top-down attention map. Different weights are assigned to each bottom-up features; where the weights are subjected to a learning mechanism.

The following subsections present various metrics, such as moments, invariants, and texture classifiers, that have been proposed in the literature as potential means to encode the observer's knowledge and to guide top-down attention. These metrics are defined here as they will support the experimental investigation conducted in the following chapters.

2.1.4 Moments of Functions

2.1.4.1 Definitions

Image moments play an important role in classical image processing techniques. They find applications as shape or texture descriptors, and for character recognition, among others. A comprehensive treatment on this subject can be found in [14]. Moments are generally defined well for continuous functions but their respective discrete versions exist as well. Equation 2-1 shows the definition of a two dimensional general moment of order *(p, q)* of a function *f(x, y);* where *g(x, y)* is the basis function.

$$M_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_{pq}(x,y)\, f(x,y)\,dxdy \qquad 2\text{-}1$$

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q\, f(x,y)\,dxdy \qquad 2\text{-}2$$

$$c_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x+iy)^p (x-iy)^q\, f(x,y)\,dxdy \qquad 2\text{-}3$$

The basis function determines what type of moment is being applied. In Eq. 2-2 and 2-3 the basis functions are replaced by standard power series and complex polynomial respectively which lead to geometric and complex moment respectively. Geometric moments are probably the most popular moments. Calculating different image properties such as center of mass of a shape or the texture of an

image using geometric moments is trivial. For example, m_{00} is the mass and $(m_{10}/m_{00},\ m_{01}/m_{00})$ represents the center of gravity or centroid.

2.1.4.2 Moment Invariance

Moment invariants are a set of mathematical formulae which remain constant for a particular pattern even if the pattern undergoes some transformations to which it is invariant. The transformation can be any basic transformation (translation, rotation and uniform scaling) or a combination of such basic transformations, uniform contrast change or affine transformation depending on the invariants being used. For example, the magnitude of complex moments are rotation invariant which means that for a particular pattern, the value of the moment invariants are the same even if the pattern goes under rotational transformation. Translational invariance can be obtained by translating the basis function to the centroid and such moments are known as central moments. To obtain scaling invariance, central moments have to be divided by a normalizing factor which is usually a power of another similar moment of low order (e.g. (0, 0)). Similarly, invariance to uniform contrast change can be achieved but the normalization factor has to be of different order than the one used for scale invariance. Equation 2-4 shows the translational invariance for geometric moments where $x_c = m_{10}/m_{00}$ and $y_c = m_{01}/m_{00}$ are coordinates of the centroid. Equation 2-5 shows the normalized central geometric moment.

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-x_c)^p(y-y_c)^q\, f(x,y)dxdy \qquad 2\text{-}4$$

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\,1+(p+q)/2}} \qquad 2\text{-}5$$

Deriving rotational invariance can be a daunting task, depending on what basis function is being used and up to what order of moments are needed. Complex moments are intrinsically invariant to rotation when only the magnitude of the moment is considered, i.e. $|c_{pq}|$. Some simple example of complex rotational invariant moments would be – c_{11}, $c_{02}c_{20}$, etc. Other types of moments can also be made rotation invariant. Simple rotation invariant geometric moments include $m_{20} + m_{02}$.

2.1.4.3 Hu Invariants

In 1962, Hu derived seven moment invariants based on geometric moments that are invariant under Euclidean transformation (translation, rotation, uniform scaling) [15]. The last one among them switches sign under mirror transformation and thus can be exploited to determine if a construct is mirror

reflected. Hu invariants use moments up to the third order and his original work does not provide directions on how to derive invariants from higher order moments. Equations 2-6 to 2-8 show the first three of the seven invariants. It can be noticed that the invariants depend upon normalized central moments; thus making them already invariant to translation and uniform scaling. Rotational invariance is achieved by the specific mathematical operations as depicted in the equations. Hu invariants have been successfully used in many practical applications since they can serve as a shape descriptor. In this work, Hu invariants are also used as a shape descriptor. A Hu shape descriptor is a seven dimensional vector that describes an arbitrary shape where the components of the vector come from the seven invariants.

$$I_1 = \eta_{20} + \eta_{02} \quad \text{2-6}$$

$$I_2 = (\eta_{20} - \eta_{02})^2 - (2\eta_{11})^2 \quad \text{2-7}$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad \text{2-8}$$

2.1.4.4 Legendre Moments

Moments in general are numerically unstable due to large exponents and numerous multiplication, division and summation operations involving real numbers. Hence, using moment-based approaches can be tricky and one has to be very careful at how it is implemented in software. Legendre moments in this respect (at least) are more desirable since they are computed using Legendre polynomials which are numerically well-behaved. Legendre polynomials are orthogonal to each other on [-1, +1] and thus deemed to retain optimum classification capabilities. Rotational invariance using Legendre moments is however difficult to achieve but the possibility is demonstrated in [16]; which also proposes a technique to compute affine invariant Legendre moments. Although image moments are primarily used on images, they can also be applied on any arbitrary function to unearth its internal structure and shape properties and thus help in representing arbitrary functions compactly. Such compact representations help in achieving efficient implementation as well. In this respect exploiting one dimensional Legendre moments can be advantageous. Equations 2-9 and 2-10 define one and two dimensional Legendre moments of order *n* and *(m, n)* of functions *f(x)* and *f(x, y)* respectively; where *P(x)* is the Legendre polynomial which is defined in Eq. 2-11. Actual computation of Legendre moments is achieved using Eq. 2-12 which is a recursive definition of the Legendre polynomials and allows efficient computation of the

polynomial by using a pre-computed table. In order for the moments to be accurate, the function *f(x)* or *f(x, y)* have to be scaled to [-1, +1] in the directions of the independent variables.

$$L(n) = \int_{-1}^{+1} P_n(x) f(x) dx \quad \text{2-9}$$

$$L(m,n) = \int_{-1}^{+1}\int_{-1}^{+1} P_n(x) P_m(y) f(x,y) dx dy \quad \text{2-10}$$

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}\left(x^2 - 1\right)^n \quad \text{2-11}$$

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x) \quad \text{2-12}$$

The function on which Legendre moments are being computed can be arbitrary and it can be represented compactly by taking a small number (e.g., for *n* belonging to the range [0, 15]) of moment values. This is particularly useful to represent histograms which can be large in size (usually 256 for 8-bit gray scale images and 768 for three-channel color images). In this case the histogram is the *f(x)* function in Eq. 2-9.

2.1.5 Local Binary Pattern

Local binary patterns (LBP) were shown to be effective for texture classification in [17]. The computation of LBP is simple and fast. A 3x3 window is considered which slides over the input image and computes a value associated with the center pixel of the window. The neighbors of the center pixels (there are 8 such neighbors) are assigned, based on their respective position, to an integer weight number which is a power of two. It is important to note that the assigned weights have to be identical for all possible positions of the sliding window. Figure 2-2(a) shows a possible assignment of weights. It can be observed that the weights form an 8-bit integer and their sum is 255. According to number theory, any number in the range [0, 255] can be formed using various combinations of the weights. Next, the intensity of the center pixel is compared to that of each of its neighbors. If any neighbor's intensity is higher than that of the center pixel, then the associated weight of that neighbor is added to the LBP value of the center pixel (which is zero initially). Figure 2-2(b and c) show two sample windows where the center pixels have intensities of 20 and 23 respectively. Their corresponding LBP value is shown in the caption.

1	2	4
128	0	8
64	32	16

(a) A possible assignment of weights to the neighbors.

25	18	40
17	20	29
15	30	24

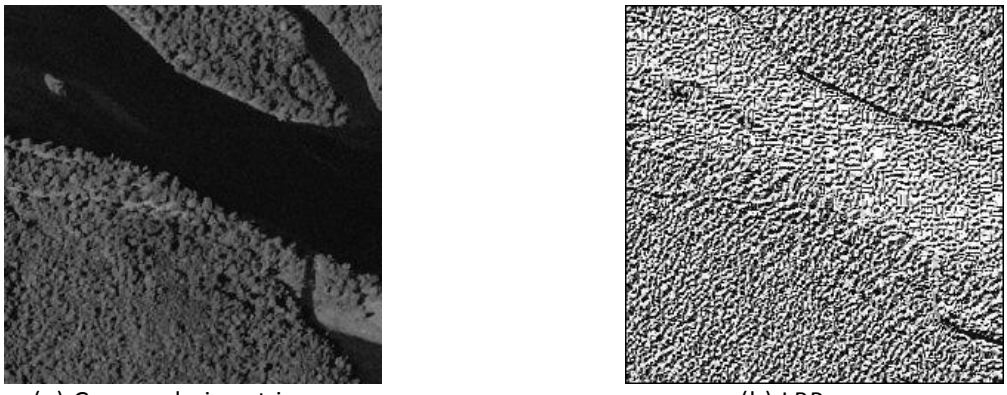
(b) A 3x3 window. LBP of the center pixel: 1+4+8+16+32=61

13	18	15
35	23	19
17	26	20

(c) A 3x3 window. LBP of the center pixel: 32+128=160

Figure 2-2: (a) Weight assignment of neighbors, (b) and (c) example windows and their respective LBP value according to the weight assignment.

Pixels whose respective neighbors have similar intensity configuration will have identical LBP values. When computation for all pixels is completed, several techniques can be applied to differentiate textures. Visually similar texture will tend to share similar LBP values and their statistical properties will be comparable. For example, two similar but different textures will have similar LBP histogram. Figure 2-3 shows a color image and its gray scale version on which the corresponding LBP map is computed. In the original image the forest and the river are dominant. The corresponding local binary patterns are also distinguishable.



(a) Gray scale input image (b) LBP map

Figure 2-3: Local Binary Pattern (LBP) map of a gray scale image.

2.1.6 Early Age of Attention Models

The human race has long been attempting to better explain and understand itself, starting from human biology to human psychology. A wide variety of psychophysical models on visual attention thus exist in the literature. A review on many of them can be found in [18]. A description of Treisman and Gelade's

model [1] which is called “feature integration theory” is presented in the next subsection which is followed by a brief overview of the development trend in the area of visual attention, along with other notable works relevant to this undertaking. In sections 2.2 and 2.3, Frintrop’s attention system is studied in depth. Following that, the main differences between the work of Itti *et al*. [3] and that of Frintrop [4] are described.

2.1.6.1 Feature Integration Theory

Treisman’s *Feature Integration Theory (FIT)* is one of the most popular and widely used visual attention models. It first appeared in 1980 [1] and has been subjected to adaptation. Many of the approaches documented in the field of visual attention are actually derivatives of this model. In Treisman’s work it was claimed that in an early stage of human visual perception, different features are automatically registered in parallel and recognition occurs at a later stage with focused attention. The features are combined in a map that shows where the objects are located. This map is known as the saliency map. The latter highlights different regions of a scene according to their level of conspicuity.

2.1.6.2 Development Trend of Computational Attention Systems

Treisman’s FIT was mostly of theoretical interest but served as the basis for more advanced research. Koch and Ullman first introduced computational approach of visual attention in their work [2]. Although it was not implemented at that time, their framework inspired some attention models. One of the earliest implementations of computational attention systems was developed by Milanese [8]. This implementation was based on the model of Koch and Ullman and uses classical image processing tools such as filter operations and edge detection for feature map computation. The system relies on features such as opponent colors, oriented edges, and intensity variations.

The *Neuromorphic Vision Toolkit (NVT)* [3], which was later devised by Itti *et al*. is one of the most popular attention systems and is based upon Koch-Ullman’s model [2]. Itti’s system was initially developed as a bottom-up attention system but was later extended to incorporate top-down attention as well [19][20][21]. The initial system uses intensity, orientation and color as the bottom-up features to compute the final saliency map. It produces results that are coherent with human attention as shown in [22]. Later, Frintrop devised another visual attention system, *VOCUS*, [4] that brought several improvements to Itti’s approach. VOCUS conceives both the bottom-up and the top-down attention along with a learning mechanism. This framework plays a major role in the work described in this thesis. Therefore, a detailed description of Frintrop’s approach will be provided in sections 2.2 and 2.3.

2.1.6.3 Data Complexity and Testing Scenarios

Most of the proposed computational solutions found in the literature have been tested mainly on indoor scenes or for a limited number of images. Moreover, in many cases the tested images are simplistic in nature with their content limited to a single object shown over a simple uniform background. It is only in the recent years that attention-based computational systems started to be studied in practical applications which introduced the need for dealing with real data. In this context, Frintrop and Jensfelt [23] use a sparse set of landmarks based on a biologically attention-based feature-selection strategy and active gaze control to achieve simultaneous localization and mapping of a robot circulating in an office environment and in an atrium area. In a similar manner, Siagian and Itti [19][20] use salient features derived from attention together with context information to build a system for mobile robotic applications that can differentiate outdoor scenes from various sites on a campus [20] and for localization of a robot [19]. Rasolzadeh *et al.* [24] propose a stereoscopic vision system framework that identifies attention-based features that are then utilized for robotic object grasping. Rotenstein *et al.* [25] propose the use of mechanisms of visual attention to be integrated in a smart wheelchair for disabled children and to help with visual search tasks.

2.2 Frintrop's Bottom-Up Saliency

Frintrop's approach [4] is one of the latest in the gamut of visual attention systems. This system is highly influenced by the earlier proposal of Itti, et al. [3]. The features that are used for bottom-up saliency map computation in Frintrop's system are intensity, orientation and color. The computation for different features is based upon filter-based operations. Different image pyramid techniques are used as an attempt to make the system scale invariant. The system first produces an image pyramid of five layers (S0, S1, S2, S3 and S4) where S0 is the input image and subsequent layers are half the size in each spatial direction of their respective previous layers. Then the two bottom-most layers (S0 and S1) are eliminated as a measure to reduce noise. The pyramid structure uses a down-sampling technique to generate successive layers and depends on the feature being computed. For example, a Gaussian pyramid is used for the intensity feature but Filter-Subtract-Decimate (FSD) pyramid [26] is used for the orientation feature. See Appendix A for a discussion on image pyramids.

2.2.1 Feature Selection

Among several bottom-up features proposed in the literature, intensity, orientation and color are the most popular, largely due to the influence of biological and psychological research works such as [27] and [28]. There are also a number of bottom-up features, apart from above mentioned ones, which are explored and reported in the literature, these include entropy, eccentricity, ellipses, symmetry, curvature, and optical flow. In general the larger the number of considered features, the more accurate the system is supposed to be. Some features are inherently complicated to deal with, such as motion which involves object dynamics. On the other hand, intensity, orientation and color remain simpler and efficient to compute, which also explains their popularity.

The intensity feature involves the computation of contrast in different regions of the image. There are two variants inspired by biology, namely on-center and off-center. On-center retinal Ganglion cells fire when a cell receives brighter light than its surroundings, and vice-versa for the off-center scenario. The higher the light contrast the stronger the firing response. The orientation feature, on the other hand, is consigned to orientation of edges present in a scene. Certain orientations of edges, such as vertical and horizontal edges, are known to respond stronger than others in the human vision system [29]. The color feature also plays an important role in bottom-up attention. In the visual field, color is represented in terms of redness-greenness and yellowness-blueness. Details on the computation of each feature used in Frintrop's system are provided in subsequent sections.

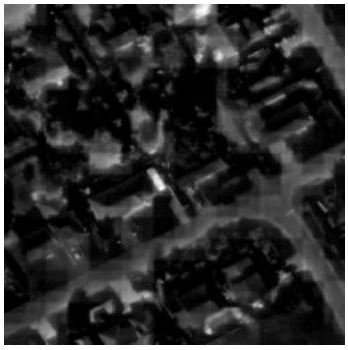
2.2.2 Intensity Feature Maps

As mentioned earlier, the intensity feature map depicts intensity variations present in the scene. First, the color input image is converted to gray scale and subsequently a Gaussian pyramid is computed based on the gray scale version of the input image. A 3x3 Gaussian filter is used in the computation of the pyramid. Down-sampling is achieved by taking out every second pixel (in both directions) after the smoothing operation. All computations regarding the feature maps are performed on every layer of the pyramid (except on the first two layers, S0 and S1, which are removed to reduce noise). To compute the on-center map, a square region is considered around each pixel from where the average intensity within the square region is computed. If the intensity of the center pixel is higher than the average intensity, then the corresponding pixel in the output map is replaced with the positive difference between the two values. Otherwise, a zero value is assigned to the corresponding output pixel. Conversely, in the computation of the off-center map, a zero value is assigned to the corresponding output pixel when the center pixel has higher intensity than the average intensity, and the positive difference is assigned if the

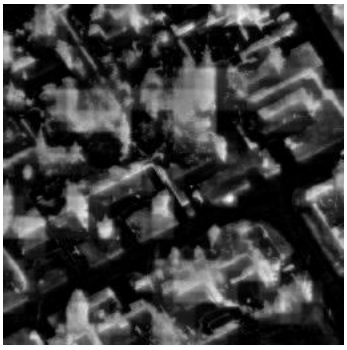
center pixel has lower intensity than the average. These calculations are applied for two different sizes of the square region: 3 and 7 pixels respectively. Hence, in total there are 12 maps generated (3 scales corresponding to the S2, S3 and S4 layers of the pyramid, 2 square region sizes and 2 types, respectively on-center and off-center). Out of these 12 maps, 6 belong to the on-center and the other 6 belong to the off-center categories. The 6 maps in each respective group are summed up yielding two maps; one for on-center and another for off-center. In the last step, addition cannot be performed in a straight forward manner since the intermediate maps are of different sizes (since they come from different layers of the pyramid). A technique called *across-scale addition* is employed where the cells of smaller scales are enlarged to fit the largest scale available (in this case S2) and then they are summed up together. The resize operation is performed using pixel replication. For example, resizing an image at scale S_i to the scale S_{i-1} is done by duplicating each pixel in both dimensions resulting in four pixels with the same intensity value in place of one. Finally, the two intensity feature maps obtained are stored for later integration in the final saliency computation.

Figure 2-4 shows an example satellite image obtained from MapQuest [30]. The image depicts a typical residential area with houses, trees, and streets. Noticeable points include the following: a few roofs near the bottom-left corner of the image are reddish while some trees are yellow and orange. There is also a large part of the surface showing grass. Streets are well separated from their surroundings. All of these factors influence intensity maps due to their strong intensity variations. The intensity feature maps obtained from that original image are depicted in Figure 2-5. Figure 2-5(a) shows the on-center map and Figure 2-5(b) shows the off-center map. Streets on the input image are relatively brighter compared to their surroundings and hence streets are highlighted in the on-center map. On the other hand, the off-center map highlights regions that have shadows, dark green trees in the middle of light green areas, and similarly contrasting regions. As expected, these two maps are sort of complements to each other.

Figure 2-4: A satellite image depicting a residential area with houses, trees, and streets.



(a) On-Center



(b) Off-Center

Figure 2-5: Two intensity feature maps computed from the input image in Figure 2-4: a) on-center intensity map, and b) off-center intensity map.

2.2.3 Orientation Feature Maps

Physiological studies demonstrated that the human visual system's orientation selectivity responses the most to the four orientations of 0, 45, 90 and 135 degrees [29]. This motivates the use of orientation selective edge detectors in the computation of the orientation feature maps. *Gabor-like* filters were identified as the best fit for this purpose [31][32]. The computation of the orientation feature is heavily influenced by a procedure known as *Overcomplete steerable pyramid filter,* described by Greenspan *et al*. [33]. The procedure employs a Filter-Subtract-Decimate (FSD) pyramid [26], a variant of Burt and Adelson's pyramid [34], and complex sinusoidal modulation. First, the input image is converted to gray scale and a FSD pyramid is constructed based on the gray scale version. Each layer of the FSD pyramid is constructed as follows: The original image is first low-pass-filtered (LPF) and subtracted from itself. The result becomes the first layer. The low-pass-filtered version is down-sampled for the next layer and the process continues. Hence at each layer of the FSD pyramid, one ends up with image details

corresponding to higher frequency contents, since the low frequency signal (after applying LPF) gets subtracted from the original signal. The level of details varies according to the scale and the image is decomposed in different frequency bands. Next, on each layer of the FSD pyramid, complex sinusoidal modulations at four different orientations (0, 45, 90 and 135 degrees) are applied. Complex sinusoids have two components – real and imaginary. Both components are again low-pass-filtered and their power is computed by summing of the square of both components. The process is equivalent to applying a Gabor filter since Gabor filters are oriented sinusoids with a Gaussian envelop. Application of complex sinusoids at a particular orientation followed by low-pass-filtering thus draws the equivalence. Overall, this process results in a total of 12 maps (4 orientations for 3 scales) and they form 4 pyramids (one for each orientation). Each pyramid is then merged into a single map by applying across-scale addition as described in the previous section. Finally, four orientation maps are obtained, each depicting edges with their respective orientation. Figure 2-6 presents the orientation maps computed from the original input image shown in Figure 2-4. The highlighted regions correspond to strong edges matching their respective orientations.

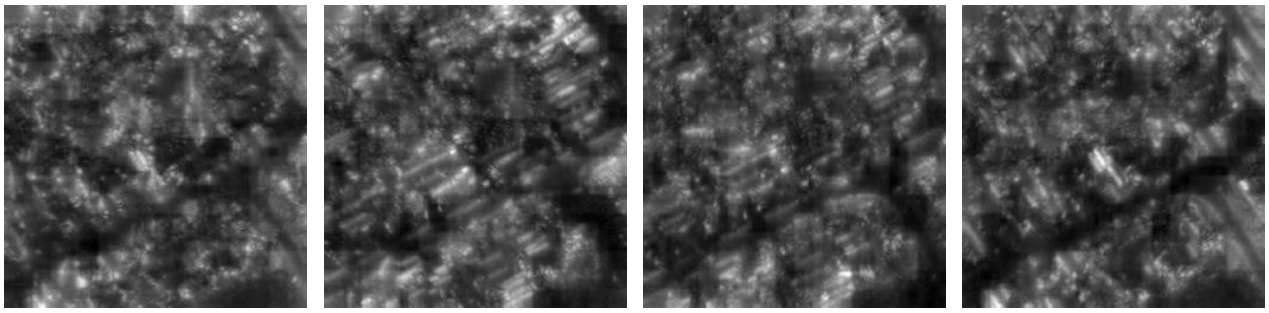
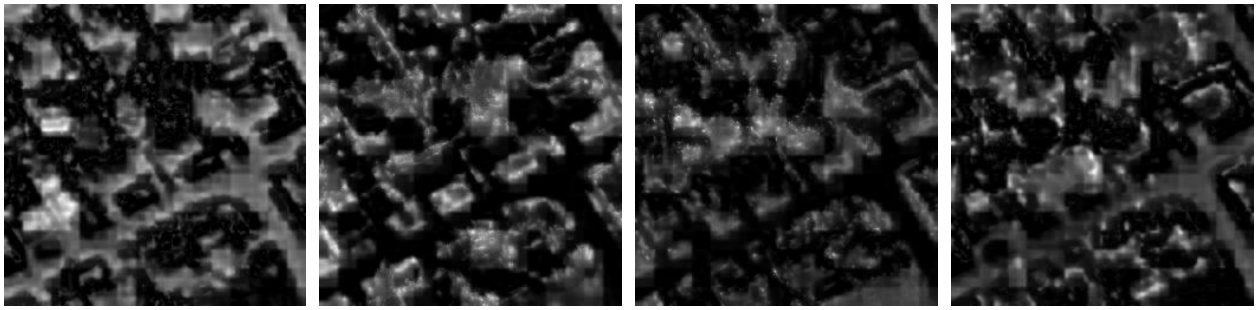


Figure 2-6: Orientation maps for orientations of 0, 45, 90 and 135 degrees respectively computed from the input image in Figure 2-4.

2.2.4 Color Feature Maps

From everyday life experience it is evident that color plays an important role in the human vision system. One interesting aspect is that, the way human beings perceive color difference is not reflected in many popular image representation techniques using color coordinates systems (e.g., RGB, HSV) [35]. The CIE LAB color space is a good fit with the human visual system's color perception. In this color space, the coordinate *L* represents a luminance value which can be related to brightness. *A and B* coordinates represent greenness against redness and blueness against yellowness respectively. A negative value of *A* represents greenness and a positive value stands for redness. Similarly, negative and positive values of *B*

represent blueness and yellowness respectively. The higher the absolute value of a particular component, the stronger is the color represented by that component. In the process of calculating color feature maps, the input color image is first converted to the LAB space (if not already in that space) and a Gaussian pyramid is built using the LAB version of the image. Following that, the *L* component is eliminated since intensity is already considered in the intensity feature maps described in section 2.2.2. This leads to a 2-component color representation. Four color pyramids are constructed for each of the base colors (i.e. red, green, yellow and blue) using the following procedure. Each pixel of each layer of a color pyramid corresponds to the amount of the associated color present at that pixel in the input image. Since the 2D coordinates represent in total 4 colors, measuring the amount of a particular color is not straightforward. The measurement of redness present in the color represented by *(a, b)* coordinates is achieved by taking the Euclidean distance between the *(a, b)* coordinates and the coordinates that represent maximum redness which is (1.0, 0) (or any other uniformly scaled or translated coordinates that represent the maximum red). This computed distance will be larger for colors that are far from red and smaller for colors that are closer to red. In other words, this procedure measures the inverse of the amount of color. For this reason the computed distance is inversed by subtracting it from the maximum distance between any two colors; and hence the final value reflects the closeness rather than distance. Similar operations are performed for the other three color pyramids at all scales. After that, to obtain color specific pop-out, an on-center map is computed (following the procedure described in section 2.2.2) on top of the computed color-closeness maps at all scales and for all color pyramids. The last step helps in identifying regions with abrupt color variations, such as a green tree in the middle of a desert. Finally, a single map is computed from each of the four color pyramids by performing across-scale summation. This results in four color feature maps, one for each of red, green, yellow and blue colors, as shown in Figure 2-7.



Red Green Yellow Blue

Figure 2-7: From left to right, color maps for colors red, green, yellow and blue respectively, computed from the input image in Figure 2-4.

In Figure 2-7, the red color map highlights the red roofs. Similarly the green color map shows the regions that are covered with grass or trees. The input image also has few yellow and orange colored trees, which are highlighted in the yellow color map. The input image does not contain much blue, but some objects have light shades of blue which can be seen from the blue color map. It can be observed that red and green color maps are somewhat complementary to each other. This is because the red and green colors are opponents to one another as described earlier. A similar phenomenon can be observed by comparing the blue and yellow color maps as well.

2.2.5 Saliency Computation

The fusion of different feature maps is one of the difficulties in computational visual attention. Fusion integrates all ten feature maps (two intensities, four orientations and four colors) into one map called a saliency map. The most trivial way of integration could be simply summing up all feature maps, but that would mean assigning equal weight to all maps. In Frintrop's model, a weighting function is rather applied on each map to compute its weight. The weighting function is also called the uniqueness function. A map with too many large bright peaks is considered less important than a map with fewer large peaks, and is therefore assigned a lower weight, *W(X)*, defined as follows:

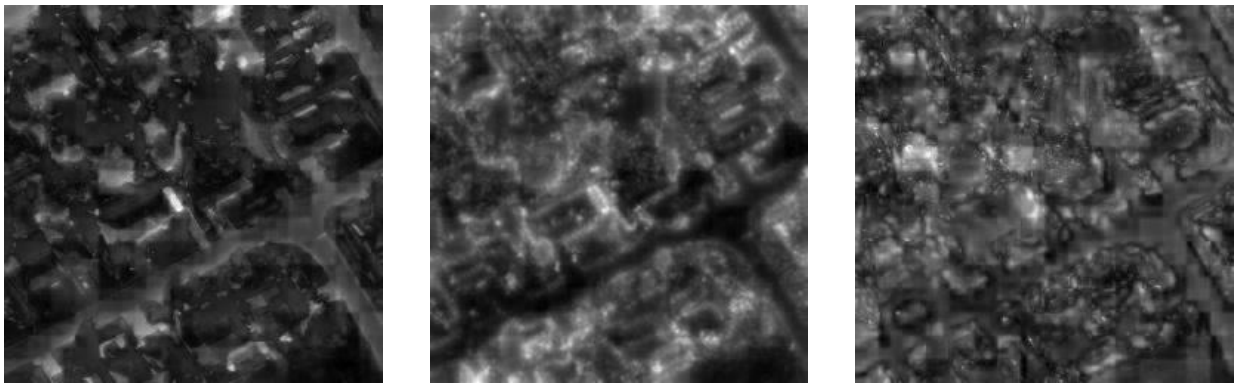
$$W(X) = X / \sqrt{m} \qquad 2\text{-}13$$

Equation 2-13 is the so called *uniqueness function* where X is a given feature map and m is the number of local maxima within a range. The range was chosen to be at least half of the global maximum. Therefore, if m is large, which means several local peaks, then the map X receives a lower weight and vice versa. The final saliency map generation is performed in two steps. First, feature maps, X_k, are fused together into separate conspicuity maps, X_c, for their respective feature. For example, the four color maps (with $k \in [0, 3]$) are fused into one final color conspicuity map, and so on for intensity (with $k \in [0, 1]$) and orientation (with $k \in [0, 3]$). Then the three conspicuity maps, one for each of the three features, are fused into the final saliency map. The fusion for each conspicuity map is performed by first applying Eq. 2-13 on each of the feature maps belonging to a certain feature (e.g., intensity) followed by their summation. Next the summed up map, X_c, is normalized to the range $[0, M]$ where M is the global maximum of all feature maps that belong to the feature in question. The normalization is necessary in order for different conspicuity maps to be comparable to each other, since the number of maps is different for different features. Finally, three normalized conspicuity maps, $\hat{X}_c$, are fused together

following a similar procedure (i.e., weighted summation followed by normalization). The general rule of fusion is shown in Eq. 2-14:

$$X_c = \sum_k W(X_k), \qquad \hat{X}_c = Normalize(X_c, 0, M) \tag{2-14}$$

Figure 2-8 shows the three respective conspicuity maps computed from the feature maps previously shown in Figure 2-5 to Figure 2-7. The intensity conspicuity map of Figure 2-8(a) shows all outstanding intensity variations. The orientation conspicuity map of Figure 2-8(b) contains a number of dominant areas, since its corresponding feature maps had several peaks except for the case of 135 degrees orientation where there is only one strong peak; and that peak is also present in the conspicuity map depicting that it received larger weight. Other orientations having too many peaks close to each other led to smaller weights and were deemed to be less important features in this context. Finally, the color conspicuity map of Figure 2-8(c) is showing a few regions that are really conspicuous due to strong color contrast, resulting mostly from colored trees and the white edge of a house roof located near the center of the image.

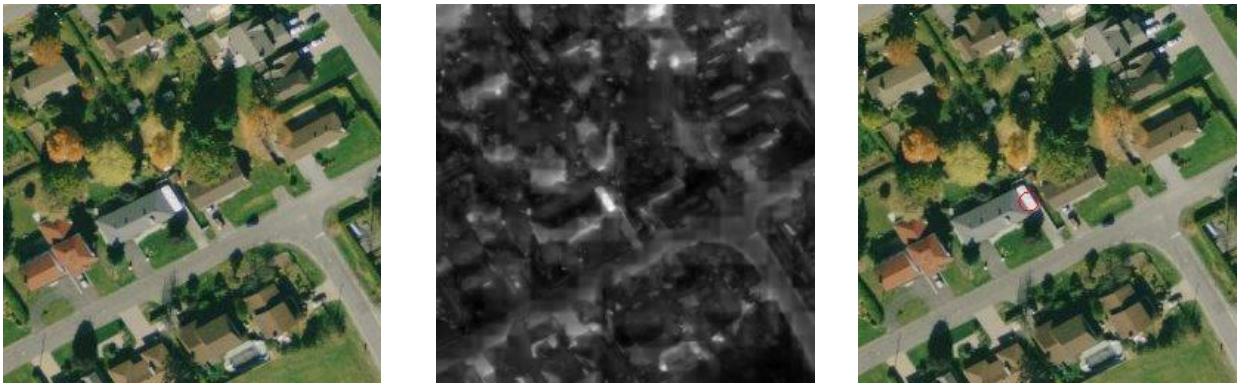


(a) Intensity (b) Orientation (c) Color

Figure 2-8: Conspicuity maps for a) intensity, b) orientation, and c) color, generated from the corresponding feature maps. The maps are normalized to [0, 1] for the sake of presentation clarity.

Lastly, the final saliency map is presented along with the input image in Figure 2-9. The sudden white regions, orange colored trees, and streets are highlighted. When the computation of the saliency map is completed, the location of the focus of attention (FOA) is performed in only one step. In Frintrop's approach, the FOA is defined as the most salient region (MSR), which is the region that contains the

largest value in the saliency map along with its surroundings. Therefore, the largest value is searched for within the saliency map, and subsequently a flood fill mechanism (see Appendix B) is applied on near pixels that have values within 25% of the largest value. The corresponding group of pixels is considered to form the MSR. The threshold of 25% was chosen by the author based on trial and error method. It is evident that if the threshold is increased; the area of the MSR will tend to increase and conversely a decrease in the threshold value will cause the area of the MSR to decrease. The attention is then directed towards the MSR, which is depicted by a red circle In Figure 2-9(c) where the MSR corresponds to the white edge of the house near the center of the image.



(a) Original input image (b) Final saliency map (c) Most Salient Region

Figure 2-9: a) Input image and b) its final saliency map. The saliency map is normalized to [0, 1] for the sake of presentation clarity. c) The input image with MSR depicted using a red circle.

2.3 Frintrop's Learning Method

2.3.1 Learning Phase

The bottom-up saliency map depicts the level of interest associated to regions in a scene to an unbiased observer. On the other hand, when the observer is looking for something specific, for example a red car on a street scene, then to that observer red objects and car shaped objects will be more attractive, even though such objects might not belong to top salient regions in the scene. This type of educated behavior is modeled using top-down saliency where features belonging to objects of interests are deliberately given more weight such that they become more salient. Frintrop's approach also incorporates a learning method that tunes a set of weights associated to each feature map and each conspicuity map (in total 13 maps) in order to facilitate a later top-down map computation. At first, a training image is provided along with a selected region of interest (ROI). Each ROI is represented with a rectangle manually selected and identified by the image coordinates that surround an object of interest. The algorithm

attempts to learn the object within the given ROI. There are a few restrictions on how to choose the training images and the ROI. In order to learn the object in the ROI, the system first computes all thirteen maps by following the procedures described in section 2.2. Next, for each feature map and conspicuity map, the MSR is determined, but only regions of the image within the specified ROI are considered. A MSR outside of the ROI is not considered even if that MSR has a higher saliency than that of the inside MSR, since the target object is known to lie inside the specified ROI. Next, the mean value within the ROI's MSR, $\overline{m}_{MSR_i}$, and the mean value of the rest of the feature and conspicuity maps, $\overline{m}_{(outsideMSR)_i}$, are calculated and the ratio between them corresponds to the weight associated with each of the corresponding maps.

$$w_i = \frac{\overline{m}_{MSR_i}}{\overline{m}_{(outsideMSR)_i}} \qquad 2\text{-}15$$

Equation 2-15 summarizes the strategy. The numerator stands for the mean value within the MSR that is contained inside the specified ROI, and the denominator stands for the mean value of the rest of the i^{th} feature or conspicuity map. The ratio gives the weight of the corresponding feature or conspicuity map. It can be seen that if a particular feature or conspicuity map has larger values within the specified ROI than in the other parts of the map, then the weight, w_i, would be higher for that particular feature or conspicuity map, and vice versa. Therefore it is possible to rank the maps according to their respective level of "importance".

2.3.2 Recognition Phase

Following the learning phase comes the recognition phase when a test image (which was not used for training) is provided to the visual attention system. The system computes all thirteen feature and conspicuity maps on the new test image and the previously computed weights are applied following Eq. 2-16, where Y_i symbolizes one of thirteen feature or conspicuity maps, w_i is its corresponding weight computed in the previous step, and I_{TD} is the final top-down map.

$$I_{TD} = \sum_{w_i>1} w_i Y_i - \sum_{w_i<1} \frac{1}{w_i} Y_i \qquad 2\text{-}16$$

The maps whose weights are larger than 1 are multiplied directly with the corresponding weight and their pixel-based contribution to the top-down map, I_{TD}, is constructively added. On the other hand, the components having a weight smaller than 1 are inversely multiplied and subtracted from the top-down map, I_{TD}. The former corresponds to excitation while the latter implements inhibition. The maps with higher weights are privileged whereas the maps with smaller weights are given less attention in the final result. Lastly, any negative value resulting from the above step is replaced with zero.

2.3.3 Limitations

There are a few inherent limitations that can be observed in VOCUS's learning mechanism. Among other things, the latter is strongly dependent on the background contents. This can be seen from Eq. 2-15 where the denominator is the mean value of the entire feature or conspicuity map excluding the MSR that is contained within the ROI. This affects the computed weight. Hence, the system also learns the background as it learns the object in the specified ROI. The computed weights also affect the final top-down map computation.

Another issue is the selection of the ROI. It is unclear how large or small the ROI should be. The numerator of Eq. 2-15 is the mean value of the map within the ROI and hence the size of the ROI would affect its value. This is because a larger ROI is susceptible to inclusion of other objects (especially when the target object has irregular shape) which would perturb the mean value and is even capable of relocating the MSR that may fall on another object. On the other hand, the MSR is restricted to a small continuous region and hence, in certain applications, insufficient to represent an entire object, especially one with a complex shape. It is expected that an object can have a complex signature which would naturally take more variables to represent. Frintrop's model does not address this case. Figure 2-10 shows an example of such limitation. Figure 2-10(a) shows an input image with a manually selected red rectangle specifying the ROI, and Figure 2-10(b) shows the result of a self-test for the recognition of grass areas (as marked by the ROI). It can be observed that most of the green field is correctly highlighted since the ROI contains a green surface. Figure 2-10(c) shows a test image which is different from the input image, and finally Figure 2-10 (d) shows the result of applying the learned weights to retrieve regions similar to the signature of the ROI. It is visible that the system fails to detect green-like objects and wrongly highlights streets.

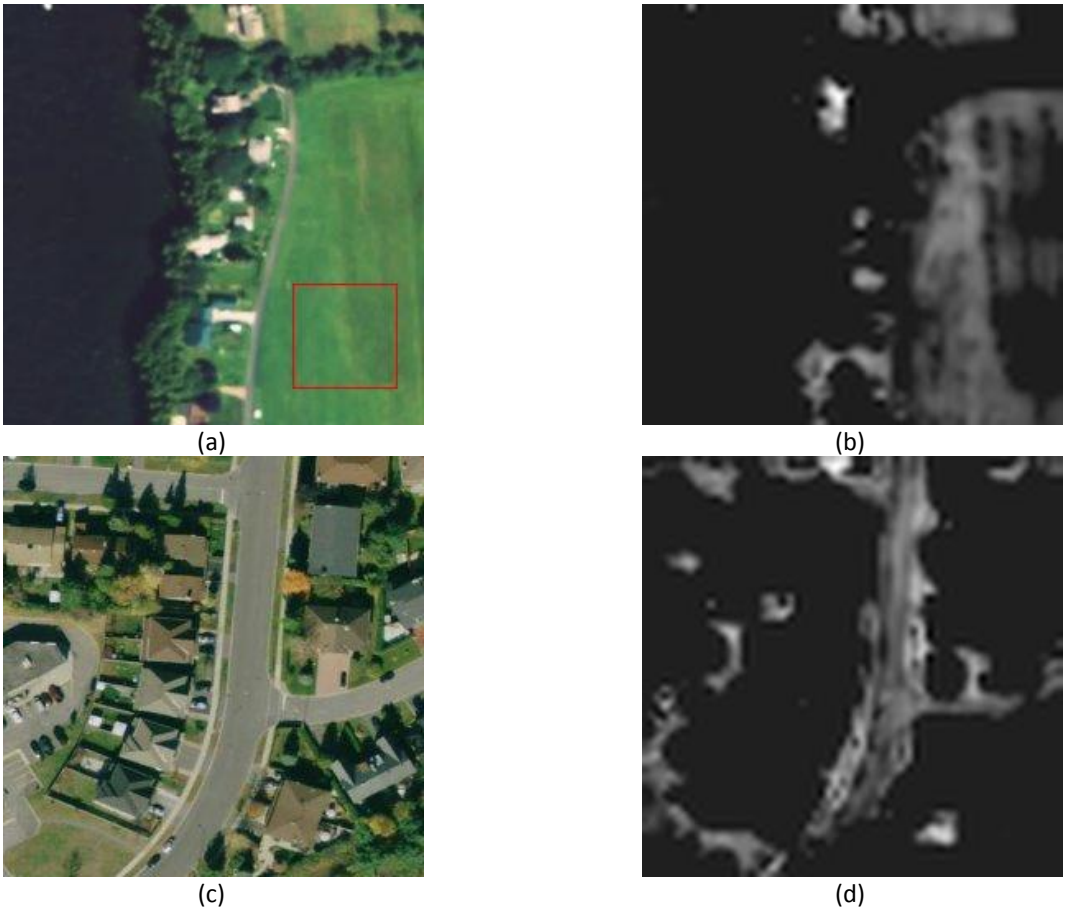


Figure 2-10: Limitation of Frintrop's top-down map computation.

2.4 Comparison between Itti's and Frintrop's Saliency Estimators

As mentioned earlier, Frintrop's system [4] represents an evolution over an earlier approach proposed by Itti *et al*. [3]. The work of Frintrop contains a complete and comprehensive description about the evolution. In this section the key differences between both methods are described. Itti's original system does not contain any top-down map computation and hence only the differences in the computation of bottom-up features are described.

2.4.1 Feature Map Computation

While computing intensity feature maps, Itti's system computes only one type of center-surround whereas Frintrop's system computes both on-center and off-center. Itti's system attempts to acquire similar effects by taking the absolute difference between the intensity of the center pixel and the mean intensity of the current window. Frintrop shows how this simplification can affect Itti's system making it ignorant to certain intensity pop-out to appear. Moreover, intensity variation in Itti's system is computed by taking the differences between two different scales, which results in blocky feature maps. On the other hand, during the top-down map computation, Frintrop's approach relies on two different intensity maps, which enables two different weights. This provides the learning system with more flexibility. Lastly, although Frintrop's system is a bit slower than Itti's, it produces a more accurate solution. In orientation feature map computation, Itti's system first applies a Gabor filter and follows with a center-surround technique. But, Frintrop's system skips the latter step, claiming that Gabor filter already simulates such an effect. All computations are performed one nine scales in Itti's system, while Frintrop's system uses only five. Itti's system uses two color maps that merge red and green into one map, and yellow and blue into another map. Frintrop's system rather uses four color maps: one for each of red, green, yellow and blue colors. As a result, Itti's approach is more restricted in color specific pop-out, since Frintrop capitalizes on two extra weights associated with two more color maps, providing a more flexible learning scheme.

2.4.2 Map Fusion

The fusion of different maps into conspicuity or a final saliency map is also performed differently. Itti's system uses Eq. 2-17 to weight a map where *M* is the global maximum and *m* is the average of local maxima, and *X* is the map itself.

$$N(X) = X * (M - \overline{m})^2 \qquad 2\text{-}17$$

On the other hand, Frintrop's system applies the uniqueness function defined in Eq. 2-13 as a weighting function on every map. It is pointed out by Frintrop that Eq. 2-17 works well only when there is only one strong peak. If there are more than one equally strong peak in the map, the resulting weight is zero. Itti's system normalizes each conspicuity to a fixed range whereas Frintrop's system normalizes to *[0, M]*, where *M* is the global maximum within all feature maps for a particular feature.

2.5 Classical Satellite Image Processing

Applications of traditional digital image processing techniques in satellite images and remote sensing have been around for nearly half a century. Satellite images not only include the color images of earth's surface taken from above, but also other types of images such as those that capture other invisible waves (multispectral imagery) that can be used to estimate altitude, land properties such as desert, forest, wetness, etc. These data can be sampled and eventually be represented as two dimensional signals or as a series of such signals. Hence traditional image processing and computer vision approaches also apply in those cases, including image quality enhancement, image registration and stitching, texture identification and classification. Considerations on these classical satellite image processing techniques are provided in the following subsections. There are also a multitude of publications regarding remote sensing and satellite image understanding. Among them, solutions have been proposed for much diversified applications such as the estimation of land surface temperature, or velocity estimation, from satellite images. But those are beyond the scope of this research work and are therefore not discussed in this monograph. A comprehensive treatment on remote sensing can be found in [36] and [37]. More general discussions on image processing and computer vision problems can be found in [38], [39] and [35].

2.5.1 Image Enhancement

Satellite images are usually taken from several miles above the ground and the resulting images can be very noisy. Interference from other sources can often cause further complications. The distance between the sensors and the ground is the primary hindrance for fine scale images and causes a single pixel to represent a large ground area, ranging from a few centimeters to several meters. There are techniques that can fuse multiple images in order to generate a higher resolution satellite image. A comparison among such fusion techniques can be found in [40]. In many cases the sensors also produce noisy, too dark or too bright images. As a result, satellite images almost always undergo an enhancement process before being used.

Image enhancement is a vast subject and is usually one of the first image processing steps that needs to be completed before other techniques are applied. Image enhancement techniques can be broadly classified into two categories: those that operate in the spatial domain and those that work in the frequency domain. Among the spatial domain techniques, the power-law transformation, log transformation, contrast stretching, and histogram equalization, are very popular. The idea behind these techniques is to transform the image dynamic range to a suitable range. There are also several denoising

techniques that attempt to reduce the noise from the signal. Image blurring through the use of different filters (e.g., Gaussian, Mean, or Median) is a well adopted solution. But, blurring techniques come with a disadvantage of also blurring away the edges present in the signal. To reduce such effect, edge-preserving denoising techniques have been proposed. One of those works by minimizing the total variation in the image. On the other hand, frequency domain related techniques work by first transforming the signal into its frequency domain by means of a Fourier transformation. Usually the noise is associated with high frequency content in the signal and hence reducing high frequencies from the signal and transforming the signal back to the spatial domain is considered as an effective denoising technique. A number of standard low pass filters (LPF) are available to reduce the high frequency contents. Techniques in both domains have high correspondences with each other.

2.5.2 Image Segmentation

In many cases it is desirable to group together similar objects present in a satellite image. Partitioning pixels (in other words grouping) in an image in different sets is known as image segmentation. The union of those partitions would give back the original image. For example, a satellite image can consist of water bodies, agricultural fields, buildings, etc., and one would want to group similar objects together (see [41] and [42] for instance). Usually, these partitions of pixels work as input to other algorithms. Object location identification and their boundary estimation are the primary goals of image segmentation. Numerous techniques are available to address this difficult problem. Defining the partitioning criteria for the pixels that would work effectively all over the image is the main difficulty. Often, the simplest technique is to use thresholding that maps pixel intensities in different preset ranges to different groups. For example, pixels having intensities in the ranges [0, 100], [101, 200] and [201, 255] would map to partitions 1, 2 and 3 respectively. However, finding the number of ranges and their boundaries is challenging. More sophisticated approaches consist of defining a parametric model (e.g., a Gaussian with a particular mean and variance as parameters) to identify segments where each segment would have a different set of parameters. Another technique is to use the k-means algorithm that can be employed to find object centers according to spatial proximity and intensity similarity. But, a k-means algorithm must be provided with the expected number of clusters (the value of k) and their initial position as input. The end result is highly dependent on these parameters.

Among many other techniques, region growing is also a very popular technique and requires a set of initial seeds to be provided as input. It is also assumed that the seeds lie within all the target objects. The regions around the seeds grow by comparing the neighboring pixels of respective seeds based on

similarity criteria. One effective criterion is to compare the intensity of a neighbor to the region mean. Split-and-merge is also a very effective technique to solve the image segmentation problem and it employs quad-tree partitioning of an image where the root of the tree is the entire image. At each step the current sub-image is divided into four sub-images if it is found heterogonous, and this subdivision goes on until the sub-windows become small enough. Next, the neighboring sub-images are merged together into one sub-image if they are similar based on some criteria. This technique produces good segmentation provided that the similarity criterion reflects the context reality. There are numerous other techniques available in the literature, such as graph theoretic based, partial differential equation based, watershed, multi-scale segmentation and energy minimizing segmentation. The choice of a particular approach always depends on the situation and the properties of the input image. In this context, section 3.2 will introduce a novel segmentation technique that uses saliency map to process satellite images.

2.5.3 Image Registration

Satellite images taken from different locations in space and time must be accurately registered in order to be comparable or to create concatenated maps of expanded areas. This step is especially important in order to detect geographical changes ([43] and [44] depict such applications). Image registration is the process where images taken from different points of view are brought under a common frame of reference. Usually this is performed by computing transformation matrices that would take a set of data in one frame of reference to another frame of reference such that two different sets of data can be integrated. It is one of the fundamental problems of machine vision. It also has applications in 3D reconstruction, image stitching, and motion measurement, among others. Image registration techniques start with determining one-to-one mappings between regions of two images. There are two main methods to find these mappings: intensity based and feature based. Intensity based methods look for similar intensity patterns between the two images via one of several cross-correlation functions. A small window is selected from the first image and its closest match is found in the second image thus establishing a mapping between them. On the other hand, feature based techniques look for significant features (e.g., points, lines, polygons, contours) occurring in both images and establishing correspondences between them. In [14], Flusser *et al*. showed how mappings between two satellite images can be found depending on local moment invariants present in them. Once the mapping is built, various techniques from linear algebra can be applied to solve for the transformation matrix that would register the two sets. The Singular Value Decomposition (SVD) technique is widely adopted in this regard since it provides a solution which is optimal in the sense of least squared error (LSE) minimization.

Depending on the context, different transformation models can also be considered, including rigid, affine or projective transformation. In rigid transformation only rotation and translation can be registered. A more general model is the affine or projective transformation that provides the flexibility of taking images from many different points of views.

2.5.4 Image Classification

In satellite image processing, image classification is often the ultimate goal. Since satellite images are generally of a multispectral nature, the problem is not only constrained within texture classification but can also relate to the classification of surface temperature, altitude, and many other geographical parameters. Given its dependency on previous enhancement and registration steps, and also because of its inherent complexity, image classification is often related to machine learning. The latter is a vast field by itself and finds many practical applications. Machine learning encompasses numerous methods, ranging from decision tree, neural network, support vector machine, and probabilistic methods. Alternatively, there is also a number of parametric models, such as Markov processes and its variants, e.g. Hidden Markov Model (HMM), that can be exploited to solve the classification problem [45][46]. Classification of high resolution satellite images using neural networks is demonstrated in [47]. One interesting classification strategy for hyperspectral imagery using Gabor wavelets is also presented in [48].

Specifying a feature vector is one of the important steps in any machine learning system and it can come in many flavors. Feature vectors can contain components that are continuous or discrete and can have different cardinalities. A training phase is necessary where the system would learn a model that would associate feature vectors to one of the classes. Selecting a proper learning model is one of the main challenges in machine learning and the selection has to be done based on the nature of the problem at hand. Depending on the choice of the learning model (e.g., Bayesian learning model), the feature vectors may have to be transformed (e.g., discretization of continuous variables). When the learning phase is completed, the learned model is used at the classification stage by feeding the model with a new feature vector which is then labeled to one of the classes. In image classification problems, the feature vectors can contain measurements ranging from simple pixel intensity up to more complex representations that include length, area, contour or other elaborate configuration descriptors. The features can also be traditional intensity or LBP (see section 2.1.5) histograms, or local moments of various types (see section 2.1.4). Classification can be performed at pixel level or at region level depending on the learning model. For example, if the feature vector is based on pixel intensities, then a

classification at pixel level is possible. On the other hand, contour based feature vectors need region based classification in which case locating a potential target region is another challenge. One of the simplest classification strategies is the nearest neighbor classification. In this strategy, a test feature vector is compared to all classes' representative feature vectors by means of some notion of distance, often based on Euclidean distance. The class closest to the test feature vector is declared as the class label. Computing a representative feature vector for a class can be achieved by averaging the feature vectors of all instances belonging to that class. Section 3.5 will propose a novel object recognition approach for satellite images that is based on an integration of visual attention with classical computational methods.

3 Methods and Experimental Evaluation

This chapter reports on several experimentations conducted regarding various aspects of computational visual attention and proposes innovative ways to analyze satellite images. Diverse attempts have been made to apply and improve both bottom-up and top-down saliency maps in order to advance image understanding capabilities using computational visual attention. Combinations of different measures and elements have been studied and the results are presented in this chapter.

At first, the effects of incorporating entropy as an additional feature for bottom-up saliency estimation are investigated. Following that, a novel technique that renders an application of bottom-up saliency in automatic image segmentation (based on visual similarity among regions) is discussed. A search technique for a target object by means of its bottom-up saliency using various moments is also introduced. Finally, a novel top-down saliency map computation technique is devised for automatically identifying locations of target objects which is subsequently exploited by an object recognition process in order to label objects.

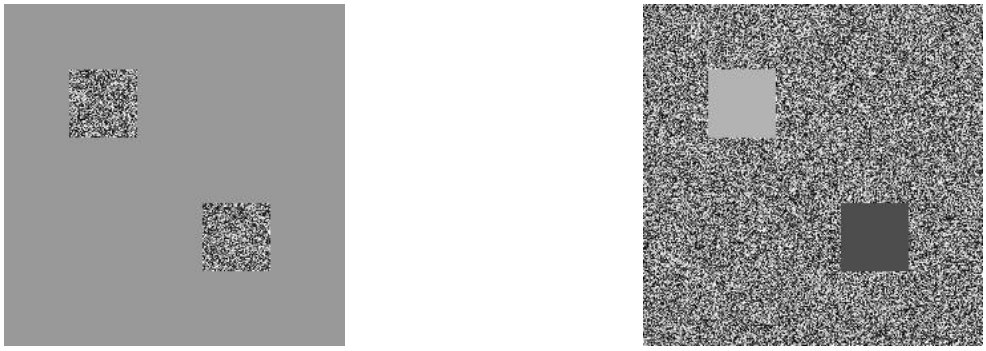
3.1 Incorporation of Entropy in Saliency Map

Entropy is a texture related feature which was not considered in Frintrop's bottom-up saliency computation algorithm. Entropy was used in other (both biological and non-biological) saliency computation systems and reported in the literature [49][9]. In [49], which is a biologically motivated approach, Heidemann *et al*. consider an entropy map as a feature for determining the presence of an object, assuming that target objects have high information content. In [9], Kadir and Brady adopt a more sophisticated statistical approach to find the optimal radius of a window within which entropy is computed, as an attempt to discover the boundaries of salient objects. None of them operate under the umbrella of biological visual attention and especially within the context of feature integration theory (FIT). It is therefore interesting to study the effects of incorporating entropy within Frintrop's system. In this monograph, variations of entropy throughout the scene rather than pure entropy are considered. A technique is devised to compute an entropy feature and integrate it within the framework of Frintrop.

3.1.1 Significance of Entropy

Entropy is a well understood mathematical quantity that can be applied to measure the amount of disorder present in the outcomes of a stochastic process. From a generic perspective, an image can be considered as the outcome of a stochastic process, corresponding to the image creation process. The entropy of an image can then be computed at either a global or local scale. Local entropy is more

interesting than global entropy in the context of an image since it shows the amount of disorder present within small regions over the entire image. As an example, a completely uniform texture with one specific color has zero entropy, which can also be construed as total predictability. Changes in colors (or gray scales) over spatial dimensions result in increased entropy; implying less predictability and hence increased disorder. On the other hand, an image constructed using totally random gray values will have very high entropy values at all scales. But, within that entirely random image, if there exists a region which is uniform (for some reason) then the uniform region is conspicuous, since its entropy differs from that of its surroundings. Conversely, a high entropy region within a low entropy region is also conspicuous. Figure 3-1 shows two example images depicting the two above mentioned scenarios. In both cases the contrast in entropy would be an effective measure to identify conspicuous regions. It can be concluded that although local entropy is useful to identify texture difference, it cannot alone be very useful to discover conspicuous regions within an image. A center-surround structure of local entropy is required to achieve this goal.



(a) (b)

Figure 3-1: a)Two high entropy regions in the middle of a low entropy region, and b) two low entropy regions in the middle of a high entropy region. The contrast of entropy is useful in discovering conspicuous regions within an image.

3.1.2 Entropy as a Feature

The entropy feature can be computed in two steps: first, a local entropy map is estimated, and second, the variations of local entropy are measured. Following the strategy of computation of other features introduced in section 2.2, it is proposed to perform all computation on every layer of the image pyramid. At first, the input image is converted to gray scale and a Gaussian pyramid is constructed based on the gray scale version. On each layer of the pyramid, an entropy map is computed as follows: a small square window of odd size (e.g., 7x7) is translated over the image, and for each position of the window the

histogram of pixel distribution is obtained. Subsequently the histogram is converted to a probability distribution function (PDF) and the entropy is calculated using Shannon's entropy equation shown in Eq. 3-1 where p_k represents the probability of the k^{th} bin (a pixel having the intensity value k within the current window) and E is the entropy of the current window.

$$E = -\sum_{k} p_k \log_2(p_k) \qquad 3\text{-}1$$

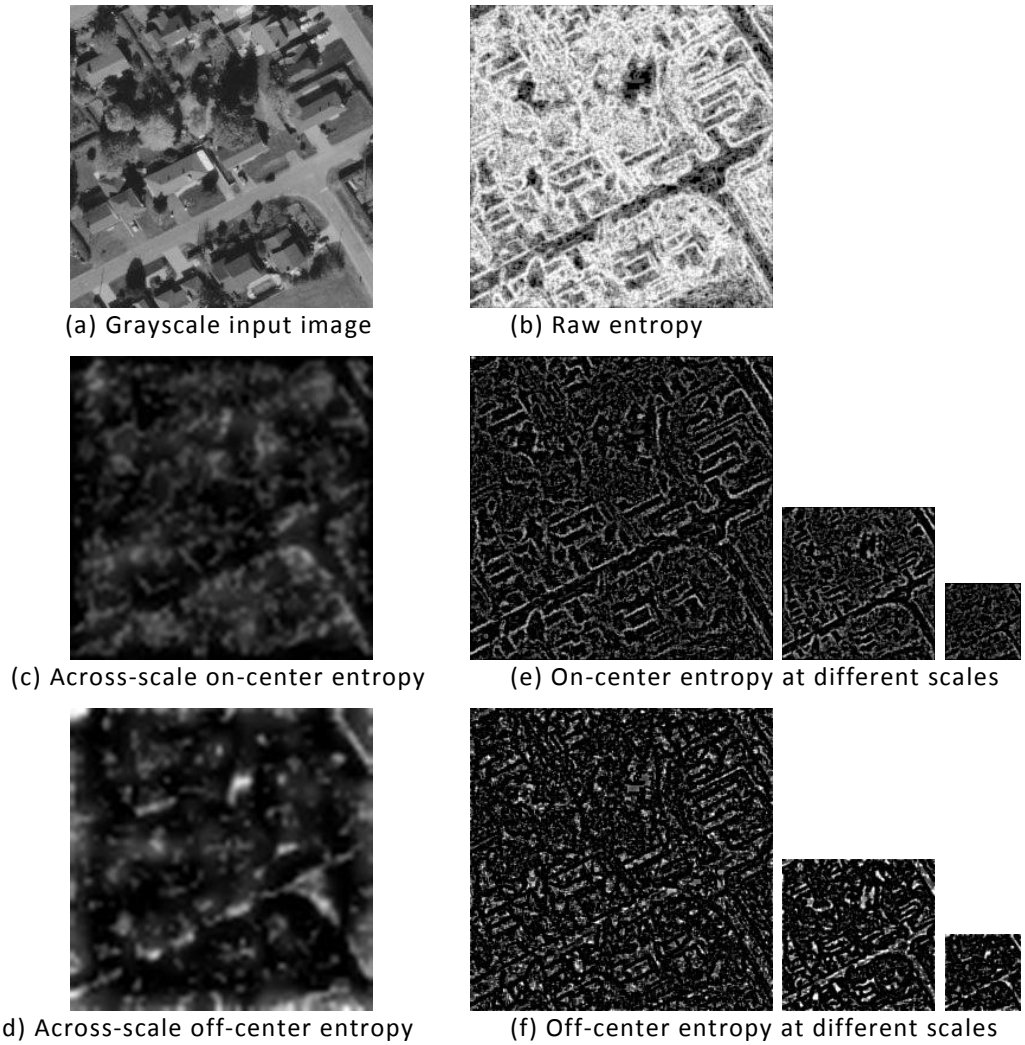
The value of the calculated entropy is recorded at the center pixel of the current window. The maximum possible entropy depends on the size of the sliding window and such a maximum occurs when all p_k are equal. The maximum possible entropy is at least n^2 for a window of size n. The entropy map is divided by the maximum value found in the map for normalization purpose and thus completing the local entropy map computation. Next, in order to compute the entropy variations, both on-center and off-center maps are computed on the local entropy map following the procedure described in section 2.2.2. Both kinds of center surround maps are required since high entropy regions in the middle of low entropy region and vice versa should be considered (as illustrated in Figure 3-1). While computing the entropy from the histogram of intensities contained in the sliding window, the number of bins of the histogram is reduced from 256. Since the window size is small (e.g., 7x7), the number of pixels inside of it is much smaller than 256. This leads to high entropy since each bin of the histogram tends to contain a small value and most of them are actually empty. To alleviate this situation the bins are clustered together for similar values. For example, a cluster size of four will lead to a histogram size of 256/4 = 64 and pixel values in the ranges 0 – 3, 4 – 7, 8 – 11 are sent to bin 0, 1, 2 respectively, and so on. An example of this clustering process is given in Table 3-1 for an artificial histogram of intensities in the range 0 – 9 (hence size is 10) shown on the upper two rows. The cluster size is assumed to be 2 and hence the clustered histogram is of size 5 which is shown on the bottom row. It can be observed that values from the original histogram are paired up and added together to form the clustered histogram.

Intensity	0	1	2	3	4	5	6	7	8	9
Frequency	7	6	2	0	2	1	6	9	0	0
Clustered	13		2		3		15		0	

Table 3-1: Illustration of a simple range based clustering process.

Figure 3-2 shows examples of the above mentioned procedure with window size and cluster size assigned to 3x3 and 4 respectively. The input image in Figure 3-2(a) is identical to the one used earlier in the computation of other features (Figure 2-4). Figure 3-2(b) shows the raw entropy calculated from the

input image. It can be noticed that within a given object the entropy values are similar while close to objects' boundaries the entropy is very high, resulting in saturated white lines. After computing the on-center entropy from the raw entropy map at the three different scales, Figure 3-2(e) is obtained, which shows the regions that have high entropy and are surrounded by low entropy. Conversely, the off-center entropy of Figure 3-2(f) shows the regions having low entropy while being surrounded by high entropy. Figure 3-2(c and d) show the on-/off-center entropy feature maps obtained by computing the across-scale sum respectively from the maps in Figure 3-2(e) and Figure 3-2(f). As mentioned in section 2.2, within the framework of Frintrop's system, final feature map is computed using the across-scale summation which is performed as follows: lower resolution scales are enlarged to fit the highest resolution available using pixel replication and subsequently added up together. This approach makes the inner parts of objects in the final map start to fill up.



(a) Grayscale input image (b) Raw entropy

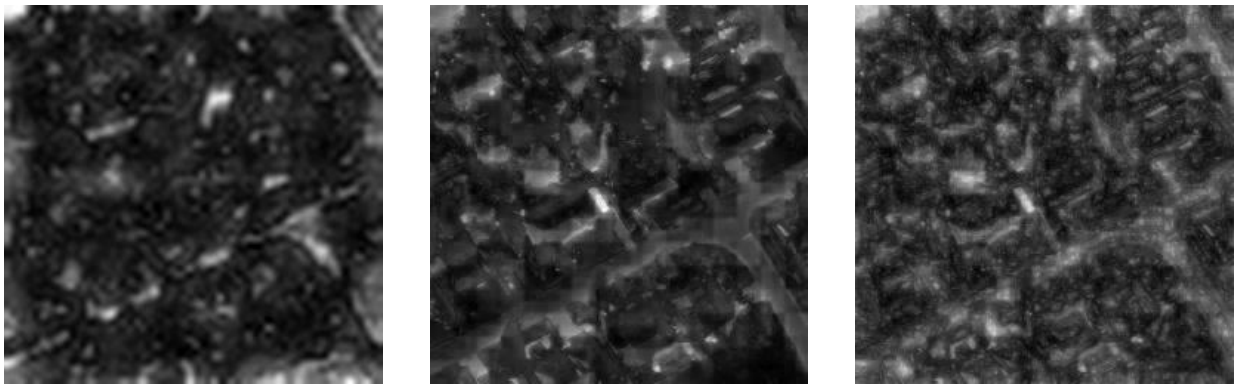
(c) Across-scale on-center entropy (e) On-center entropy at different scales

(d) Across-scale off-center entropy (f) Off-center entropy at different scales

Figure 3-2: Entropy feature and its related maps.

3.1.3 Experimental Results

In order to incorporate the computed entropy maps as extra features into the final saliency map, the feature map fusion procedure described in section 2.2.5 is also applied on the entropy maps. Figure 3-3(a) illustrates the final entropy feature map after fusing the on-center and the off-center entropy maps. It shows all entropy variations over the input image. Figure 3-3(b) is identical to Figure 2-9(b) which is the final saliency map obtained from the same input image when only intensity, orientation and color features maps where considered. It is reproduced here for comparison purposes. Finally, Figure 3-3(c) shows the saliency map obtained after incorporating the entropy map along with the other three feature maps. The resulting saliency map appears slightly noisier which also signifies that saliency has become sensitive. Small changes in the input image are also producing salient regions. This saliency can be used in detailed scene analysis to identify fine changes of the input image contents while the earlier one, without entropy, can be used for higher level analysis of the image.



(a) (b) (c)

Figure 3-3: (a) Entropy feature map after fusing the on-/off-center entropy maps, (b) saliency map without entropy (shown earlier in Figure 2-9(b)), and (c) saliency map after incorporating the entropy as an extra feature map.

After incorporating the entropy feature, the focus of attention (FOA) defined by the most salient region (MSR) can change its location, as is the case for the input image shown in Figure 3-4(a). The MSR after the incorporation of entropy feature is shown in Figure 3-4(c). Red and Green circles show the first and second MSRs respectively. It can be observed that the single MSR of Frintrop's system (red circle in the center of Figure 3-4(b) becomes the second MSR (green circle in Figure 3-4(c)) while the dominant MSR (red circle in the upper part of Figure 3-4(c)) moves to a totally different region after incorporating the entropy feature.

(a) Input image (b) MSR with Frintrop's system (c) MSRs with entropy feature

Figure 3-4: Red and green circles show the first and second MSRs respectively, when entropy is considered, or not, to measure saliency.

The change in location of the MSR happens due to the significant contrast of entropy, present within the scene which may assign higher weights for entropy feature maps resulting in different a saliency map and hence different locations of the MSR. The uniqueness function of Eq. 2-13 is responsible for assigning such weights to different maps and it is possible for the entropy conspicuity map to receive a larger weight than other features, given it has fewer strong peaks than other conspicuity maps. The saliency map can be seen as a scoreboard where scores are assigned to all regions by different features. The region having highest score is the MSR. The contribution of entropy feature has a direct influence on the scoreboard. For example, in the above cases, Figure 3-4(b) has its first and second MSR values equal to 0.1582 and 0.1545, respectively, and Figure 3-4(c) has its first and second MSR values 0.1977 and 0.1749, respectively. Hence, incorporation of entropy does not only change MSRs but also the difference between the saliency levels of the MSRs.

In this experiment, entropy is not utilized directly; rather, its variation is considered and is one of its originalities. Also the contrast in the information content is considered in both on-/off-center sense to make it possible to detect both high entropy regions in the middle of low entropy regions and vice versa. The entropy feature is seamlessly integrated within the framework of Frintrop's system along with other features. The resulting bottom-up saliency maps are more sensitive than that of Frintrop's original system. In a situation where such capability is required, one can exploit the power of the entropy feature.

3.2 Automatic Image Segmentation Based on Visual Similarity

In previous sections, different ways of computing a bottom-up saliency map have been presented. In this section, an application of such a saliency map for automated image segmentation is introduced. The work related to this contribution has been published in [5]. In the proposed approach, the saliency map is viewed from a different perspective. While the MSRs show regions that attract attention, the least salient regions (LSR) also direct the observation towards regions that are homogeneous. Their homogeneity makes them less salient. The study of less salient regions is an aspect that is largely ignored in Frintrop's VOCUS. However, the LSRs reveal to be good locations for the system to learn about the visual appearance of the scene, particularly due to their homogeneity. If a model is learned from an LSR in training images and used later on to search for similar model parameters in a test image, similar regions would be grouped together. This leads to the introduction of an original segmentation scheme for satellite images based on visual similarity.

3.2.1 Proposed Algorithm

The novel algorithm that is proposed is based on the bottom-up saliency map built by VOCUS. Legendre moments are used as a tool for compact yet effective representation of colored texture to measure visual similarity and are applied on the probability density function of histograms (see section 2.1.4.4). Any saliency detection approach could be used instead of VOCUS without affecting the proposed segmentation mechanism. After the computation of the bottom-up saliency map, as detailed in section 2.2, the proposed solution determines the LSR of the map. In order to find the LSR, at first the global minimum point in the saliency map is found, which is followed by the application of a flood-fill technique (see Appendix B) to locate the neighborhood where the values are very similar to that of the minimum saliency point. That neighborhood is considered to be the LSR and it is marked such that this region does not come up next time when another LSR is requested. If the size of the LSR is not sufficiently large (the next section details how all parameters are set), it is discarded and the next LSR is found according to the procedure described above but considering only the unmarked pixels. A sufficiently large region is required for the learning scheme to work properly. The proposed learning scheme is based on the RGB color histograms of the region in the original image that corresponds to the LSR in the saliency map. After the LSR is found, its corresponding region is extracted from the source image which is expected to be homogenous (as discussed earlier), and a color histogram (one for each color channel) is computed for the region extracted from source image. The histograms are subsequently converted into a probability density function (PDF) by normalizing the histograms (i.e.,

dividing each bin by the sum of values of all bins). From the PDF, *f(x)*, Legendre moments are computed up to a certain order *n* using Eq. 2-9. The minimum size of the LSR and the number of orders up to which the Legendre moments of the PDF are computed are parameters to be tuned in the proposed approach. All the parameters and their effects are discussed in the next section.

Once the Legendre moments of the PDF for the color histograms are computed, the image can be analyzed. The entire image is searched for similar Legendre moment values up to the order used in the learning phase. For this task, a sliding window of a given size is considered and for each position of the window the Legendre moments of the PDF of three color histograms (for red, green and blue) of that window are computed. If the moment values are similar to those of the learned LSR, then the center pixel of the window is marked to be similar to the LSR. The measure of similarity is based on Euclidean distance considering that the Legendre moment values form a vector having *n* components. If the distance is less than a threshold, it is considered to be similar.

The whole procedure is repeated for the rest of the unmarked regions of the image. Gradually, the marked regions grow and unmarked regions shrink. In those cases where there is no region similar to the current LSR, the latter is considered a segment by itself. It is also possible to have small regions scattered around the image that may or may not be similar to other parts of the image. They remain unmarked and are called orphan regions. One of the major advantages of the proposed approach is that the algorithm does not need any seed point or any user intervention, since it selects the best region to learn a model from by itself, based on visual attention, encoded in the saliency map. After properly setting the parameters, it operates in a fully automated manner and reports on the visually similar segments of an image. Morphological operators can be applied as a post-processing step to fill up small holes. The pseudo code of the proposed algorithm is given below:

BIO-SEGMENT (I)

1. Compute saliency map M of image I
2. Find the LSR of M and compute the area S of the LSR
3. If S >= minimum required LSR area:
    a. Then Ref_Moments = compute histograms of the region corresponding to LSR from I and subsequently Legendre moments up to order n for all color channels
    b. Wnd_Moments = slide a window of a given size W over I and compute Legendre moments as in step 4
    c. D = compute Euclidean distance between Ref_Moments and Wnd_Moments
    d. If D < threshold (T):
        i. Then add the center pixel of the current window to the current segment and discard the pixel from I
4. Repeat steps 2 – 3 until no more LSR can be found

3.2.2 Setting the Parameters

There are five parameters to be set in the proposed method, namely the minimum size of the LSR (*S*), the number of bins for histograms (*B*), the maximum order used for Legendre moments (*N*), the similarity threshold (*T*), and the size of the sliding window (*W*). The minimum size of the LSR is crucial. If it is too large, the system may not find any suitable region, resulting in an increased number of orphan regions. On the other hand, too small LSR regions may not provide adequate support for learning. The number of bins used in the histograms can be used to tune robustness. A large number of bins may capture too much detail about the region, leading to a larger number of segments and an increase in computation time. A smaller number of bins may increase robustness, but too few bins may falsely match visually different regions and lead to a reduced number of segments. The maximum order of moments used has similar effects to the choice of the number of bins. A similarity threshold between Legendre moment vectors is also used to tune robustness/rigidness. A too large threshold value may match visually different regions and a too small value may force the system to match only regions looking exactly the same. But overall, the most important parameter is the size of the sliding window. It determines the fineness of the segmentation. If it is too large then close region boundaries include a larger portion of other regions and the detection fails. If the window is too small, it does not provide enough pixels to compute representative color histograms. Among all these restrictions, it is still possible to find a proper set of parameters that work over a given class of images. Parameters values used during the experiments for a broad set of satellite images, estimated by trial-and-error, are: S = 225, B = 128, N = 10, T = 2.0, W = 9x9.

3.2.3 Experimental Results

Figure 3-5 presents three input satellite images with various levels of complexity and their corresponding segmented images. In the latter, different shades of red and green represent visually similar regions. Black regions are orphan regions. Figure 3-5(d) highlights that the empty field as well as the lake entirely share the same shades of red, respectively. The trees are also properly segmented along the shore although different shades of red are visible in some places. This is due to the effect of the size of the sliding window. In Figure 3-5(f) the green field and the bushes are also nearly properly segmented. The trees are also segmented well with few holes and noise which may be removed by the application of morphological operators. Overall, the proposed algorithm provides fairly good results for the segmentation, even for relatively complex images such as the one illustrated in Figure 3-5(f).

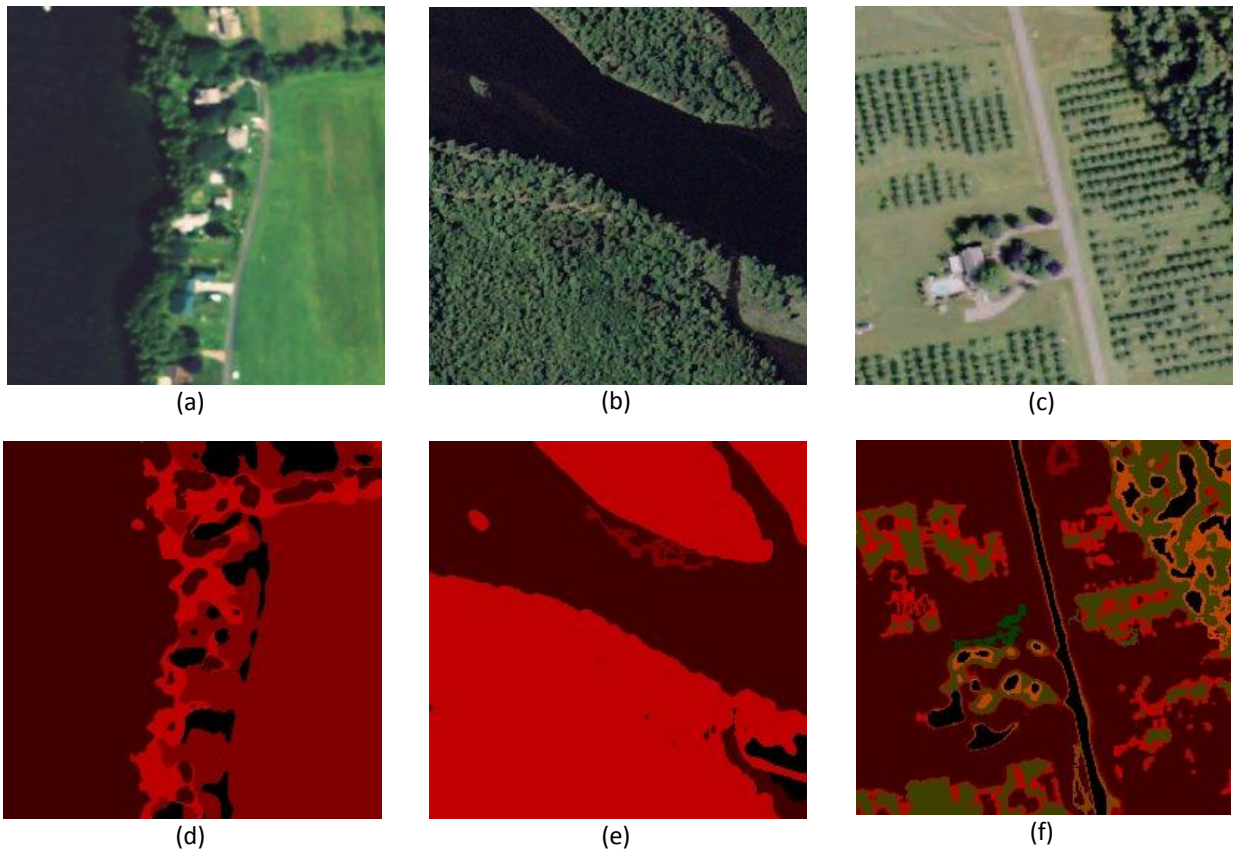


Figure 3-5: (a) – (c) Input images, and (d) – (f) corresponding segmentation where shades of red and green depict visually similar regions.

3.2.4 Effects of Varying Parameters

In section 3.2.2, the effects of various parameters were discussed. In order to provide visual evidence, more experiments were conducted. Three most important variables were experimented with; namely, matching threshold (T), minimum LSR area (S) and size of sliding window (W). For each parameter three different values were chosen while keeping other variables constant (as given earlier).

The first case deals with the required minimum LSR area (S). An increase in this value will tend to increase orphan regions since enough regions may not be found that are suitable for learning. Figure 3-6 shows a test run of this experiment. It can be observed that orphan regions (dark areas) are increasing from left to right as the value of S is increasing.

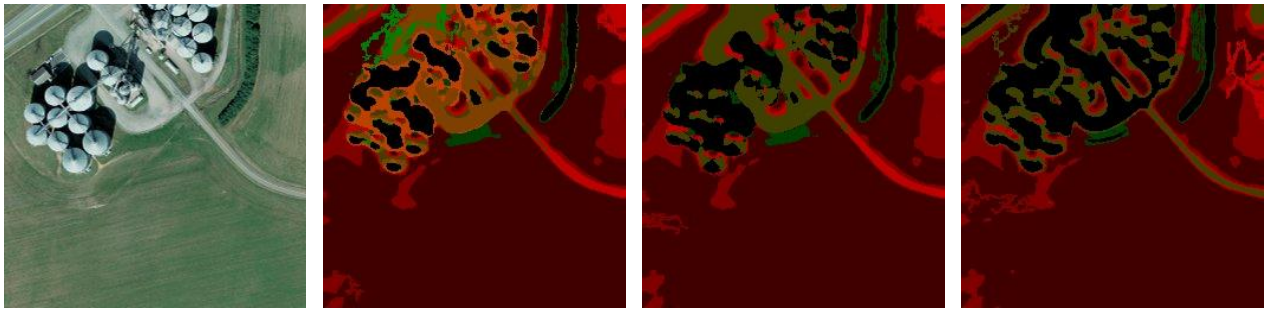


Figure 3-6: Input image and segmented images with S = 100, 225, 350 from left to right, respectively.

The next experiment was performed with varying threshold parameter (T). It is expected that a higher threshold would lead to easier matches. Figure 3-7 shows the effect of increasing the threshold from 1.5 to 2.5. It can be seen that the trail and the dark green region on top-middle are shrinking gradually. Also the number of different colors is decreasing which implies that more regions are grouped together as a result of increased threshold value.

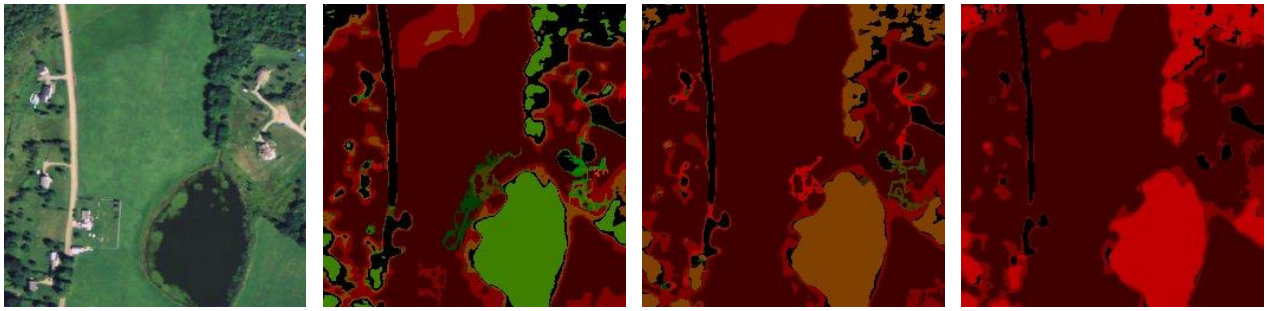


Figure 3-7: Input image and segmented images with T = 1.5, 2.0, 2.5 from left to right, respectively.

The last experiment was conducted using the window size parameter (W). Figure 3-8 depicts a test run that provides the effects of such changes. From left to right, the window size parameter was changed from 7x7 to 11x11. It can be noticed that the gray trail beside the green field gradually vanishes as the sliding window size increases. Other small details such as small orphan regions also disappear as window size increases. Such phenomena occur since a larger window includes more surrounding regions of a pixel and hence tend to match regions that only slightly differ more easily.

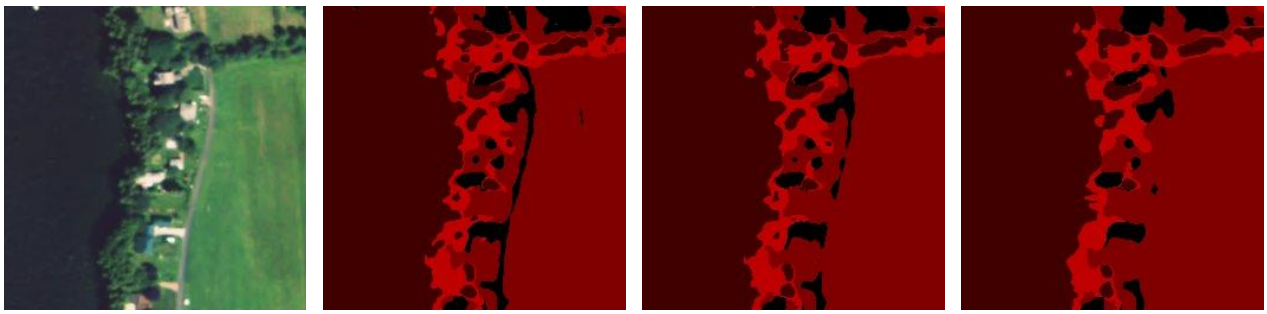


Figure 3-8: Input image and segmented images with W = 7x7, 9x9, 11x11 from left to right, respectively.

3.2.5 Alternate Application for Region of Interest Search

The proposed segmentation algorithm detailed in section 3.2.1 also allows searching for a given region, defined from a training image, in another test image without computing the saliency map of the training image. In this case, the selected ROI is learned in the same way as described in section 3.2.1 by computing the Legendre moments of the color histograms' PDF from a training image. The ROI is manually selected according to the operator's expectation rather than being automatically selected from the LSR of the training image. The entire test image is then searched for similar Legendre moments. It therefore allows for the identification of visually similar regions in the test image, rather than aiming at a self-segmentation of the input image. Experimental results are presented in Figure 3-9. The training image with the selected ROI depicted by a red rectangle is shown in Figure 3-9(a). Binary images on the bottom row of the figure show the results of a search for the learned ROI over different images depicted in Figure 3-9(d) (self-test) and Figure 3-9(e and f), with white regions indicating visual similarity to the learned region (a green surface corresponding to grass). It can be observed that in Figure 3-9(f) the top-middle region where the grass is darker was not detected by the system. While, it is also possible to detect that region along with the grass on the field by increasing the threshold value (T), but then other areas start to match and merge. A trade-off is therefore necessary. This phenomenon is shown in Figure 3-10. It is interesting to notice how white regions gradually grow from left to right as the threshold value increases, as a consequence of the matching criteria being relaxed.

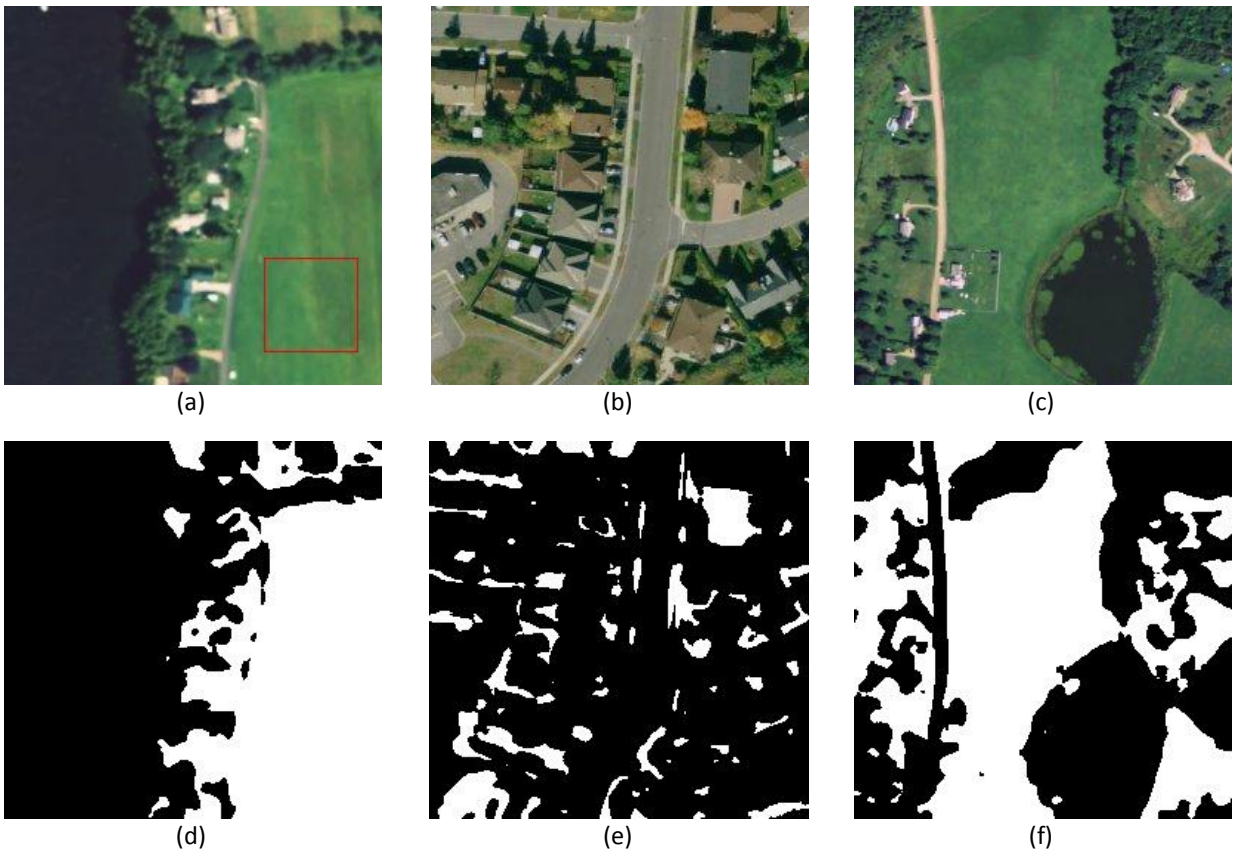


Figure 3-9: (a) Training image with selected ROI in red rectangle, (d) result of self test where the test image is the training image itself. White regions indicate a successful match; (b) and (c) different test images, and (e) and (f) results of the search for similar ROIs.

Figure 3-10: Variations in the detected region of interest as threshold values vary: from left to right T=1.5, 2.5, 3.5 and 5.0.

The proposed segmentation technique described in this section makes use of the least significant regions of bottom-up saliency maps which are largely ignored in the literature. It is demonstrated that the LSRs can be very useful as well. The segmentation results obtained with this method are plausible and the technique can also be used to effectively identify targeted regions in complex satellite images. Such targeted regions can represent vegetation, water source, agricultural fields, etc., that have a defined texture. The selected ROI is confined to a rectangle and can be of any shape such as polygon, circle or any arbitrary shape.

3.3 Moment-Based Saliency Search for Objects of Interest

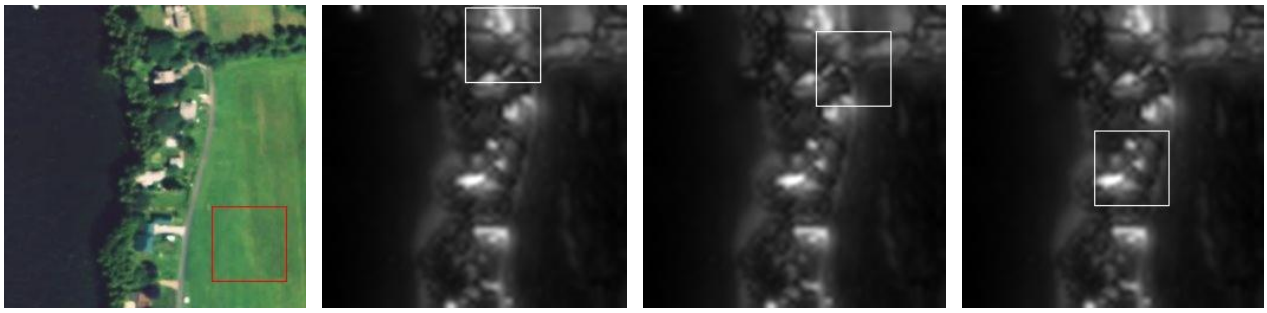
As discussed in the previous section, the capability to search for a specific target object is a very useful capability when analyzing satellite images. Applications of such functionalities can easily be found in intelligence and military activities, territory management, environmental conditions monitoring, and emergency preparedness. Given that the original VOCUS system, detailed in sections 2.2 and 2.3, faces challenges in the search mode when the background in the analyzed image differs from the background in the image used for training, as was demonstrated in section 2.3.3, an extension is considered here to Frintrop's approach that is expected to make the search independent from the background information. The strategy uses moment invariants.

Image moments of various types and their invariants have been discussed in section 2.1.4. Experimental tests were run in the context of satellite images with a set of classical moment invariants, namely, complex moments [14], Hu invariants [15] and Legendre moments [14][16], in order to evaluate their respective performance. The Legendre moment invariants considered are invariant to translation, scaling and uniform contrast changes but they are not rotation invariant. On the other hand, the complex moment invariants considered are rotation/translation/scaling invariant but are not invariant to uniform contrast changes.

At first, given a training image along with a ROI containing the desired subject, the image region within the given ROI is extracted and the rest of the image is discarded to make the subject definition independent from its background. Next, Frintrop's bottom-up saliency is computed only on the extracted region considering that smaller image as a standalone image. After that, the above mentioned moment invariants are computed on the computed saliency map. Once the visual appearance of the subject of interest is defined via the moment invariants, a test image is processed to determine its bottom-up saliency, and the same moment invariants are computed. Finally, similar moment invariants to those defining the subject of interest are searched for in the saliency map of the test image to retrieve regions that correspond to the target object.

Figure 3-11 shows a test run with the procedure described above. Figure 3-11(a) shows the training and the test image (self-test) along with the ROI depicted in the red rectangle. Figure 3-11(b-d) show the results of applying respectively Hu, Complex and Legendre moment invariants for the detection of similar regions. The white rectangles illustrate the best match found that correspond to the saliency of the original ROI when this saliency is represented via the three different moments. Unfortunately none

of the moment invariants allow for a reliable retrieval of the original ROI in spite of its simple signature (all green grass), not even for the self-test scenario.



(a) (b) (c) (d)

Figure 3-11: (a) Input image along with ROI in red rectangle. Test image is identical (self-test). White rectangles show the best match found by (b) seven Hu invariants, (c) five Complex moment invariants up to the 3rd order, and (d) three Legendre moment invariants up to the 3rd order.

A closer look into the ROI reveals a small area where the grass is yellowish. This part of the ROI happens to become the MSR of the ROI. Since Frintrop's method only retains one MSR from the saliency map, the actual green grass signature is neglected during the learning phase of the visual appearance of the region of interest that gets defined via the moment invariants. However, when the entire image is considered during the search phase, the yellowish grass that generated dominant saliency almost vanishes due to the normalizing functions used by VOCUS and the attraction that results is directed toward other more salient regions, such as the bright houses and the road. Although Legendre moments are invariant to uniform contrast changes, the method also fails to find the original ROI. This suggests that the resulting bottom-up saliency of the ROI when the entire image is considered has changed non-uniformly. This experiment demonstrates that a direct use of moment based saliency search for objects of interest is inadequate since the bottom-up saliency of an object also depends on the presence of other objects including the background. Later in section 3.5, different usages of moments are shown to be effective in object recognition.

3.4 Proposed Energy-Based Top-Down Saliency

Locating a target object in an image is often a prerequisite for successful recognition. A moment-based approach, building on saliency, was discussed in the previous section to locate a desired object, but provided doubtful performance due to a biased representation of knowledge. Alternatively, and as discussed in subsection 2.1.3, top-down attention can be considered as a goal-directed procedure that offers some potential to locate objects of interest. In her original approach, Frintrop's system was highly background dependent as illustrated in section 2.3.3. This section introduces a different top-down learning method, where objects' feature learning is viewed from a different perspective. The work reported in this section was published in [6].

Specifying an object of interest through a ROI for learning purposes is not always an appropriate procedure. Indeed, it is sometimes difficult for the operator to identity which objects would be of the highest interest, especially in the context of complex and very large satellite images. With the proposed learning method, one does not have to specify any particular object. Rather the system automatically identifies them from a set of training images. Unimportant objects are automatically discarded. The features that are strong are discovered and assigned higher weights while weak features are assigned lower weights.

3.4.1 Learning Mechanism

In Frintrop's learning mechanism, during the computation of the top-down map, the importance of a feature is estimated from the feature and conspicuity maps as the ratio between the mean value of MSR contained within the specified ROI that marks the subject and the mean value of the rest of the feature and conspicuity maps (that is the regions excluding the MSR). This mechanism is detailed in section 2.3. In the proposed learning mechanism which does not involve any ROI specification, the importance of a feature map is defined in terms of the energy of the signal. A high energy feature map is given more importance than a low energy one. The energy for a discrete signal is defined in Eq. 3-2.

$$E(X) = \sum_{x} \sum_{y} X[x, y]^2 \qquad 3\text{-}2$$

High energy values in a given feature map, *X[x, y]*, signify that from the perspective of bottom-up attention, the map is important. For example, if the training image contains objects that have strong vertical edges, then the feature map for the 90 degree orientation would have higher energy values than those of other orientations. Conversely, if the training image does not contain a red object then the energy of the feature map for the red color would be very weak. In both cases the energy of a map is

capable of encoding such information. At the same time, since the proposed approach does not involve any formally defined MSR, it is not restricted to a single region that defines an object. This represents a major benefit over Frintrop's original learning method described in section 2.3. The proposed learning mechanism is well-suited for learning from a large number of images, since the algorithm only uses feature maps. Therefore the computation of conspicuity maps is not necessary.

The proposed solution can be described as follows: for each image in the training set, ten feature maps (two intensity maps, four orientation maps and four color maps) are computed according to the process described in section 2.2. The uniqueness function of Eq. 2-13 is then applied on each of them. Next, the energy of each of the ten maps is computed using Eq. 3-2, where *X* represents each of the feature maps respectively. An energy vector having 10 components, *B*, is formed for each training image where each component of the vector represents the energy of the respective ten feature maps. It is important to note that all these vectors, each corresponding to one image, would not be too dissimilar as will be further discussed in the next section. Next, all these energy vectors are combined to compute the average energy vector, *R*, for the training set using Eq. 3-3 where *n* represents the number of training images. Finally, the resulting 10-dimensional average energy vector, *R*, is mean-shifted using Eq. 3-4 where r_i are the 10 components of *R* corresponding to the 10 different feature maps considered and $\bar{r}$ is the average vector computed over the components of *R*.

$$\boldsymbol{R} = \frac{1}{\boldsymbol{n}} \sum_{k=1}^{n} \boldsymbol{B}_k \qquad 3\text{-}3$$

$$r_i = r_i - \bar{r}; \quad \bar{r} = \frac{1}{10} \sum r_i \qquad 3\text{-}4$$

The resulting r_i components from Eq. 3-4 compose the final learned weight vector, *W*, defined in Eq. 3-5.

$$W = [r_1, r_2, ..., r_{10}] \qquad 3\text{-}5$$

Note that the sum of the r_i components of the weight vector is zero (since they are mean-shifted), which also means that at least one of them is negative and at least one of them is positive. This is helpful to inhibit the features from the final top-down map that are weak and excite the features from the final top-down map that are strong. After the weight vector is learned for a training set that represents a class of images (see next section for details), the top-down saliency map, *TD(I)*, of a given image, *I*, is computed using Eq. 3-6, where X_i are the respective ten feature maps (two intensities, four orientations, and four colors) of *I* , and r_i are their corresponding learned weights from Eq. 3-5.

$$TD(I) = \sum_{1 \le i \le 10} r_i X_i \qquad 3\text{-}6$$

The inhibition and excitation are applied automatically in the summation due to the signs of the respective weights. Given that the test image, *I*, belongs to the same class as the training set, then the objects that were interesting in the training set will be highlighted in the test image. In case the test image belongs to some other classes, the top-down map will highlight similar looking objects. The proposed technique is advantageous in that the system does not need to know explicitly what features of the objects to learn from. Based on the strengths of bottom-up features, the proposed energy-based system identifies and enhances the "important" (from an energy perspective) features that characterize the objects in images from a given dataset automatically.

3.4.2 Selection of Training Set

For the significant regions to be identified properly, the training images are chosen such that their contents are similar and relevant for the task that the system is trained for. For example, if one wants to process satellite images of cities, the training set should ideally contain only images of cities and not contain other types of images such as desserts, farm lands, or sea. The use of other images than those relevant for the task results in changes in the 10-dimensional vector, *B*, that characterizes each image and that, in turn, leads to a bias in the resultant average energy vector, *R*, which then inaccurately represents any of the classes of interest. When they are representative for the task, the "objects of interest" are automatically learned by the system. The latter objects are defined by features that are strong with respect to their neighborhoods and they therefore get highlighted in the bottom-up saliency map, and as a result, are further enhanced by the strategy proposed in the previous section. For example, in images of a city, among the many objects present, objects of interest can include houses, roads, or parking lots.

3.4.3 Experimental Results

In the initial experiments, a set of 15 images (of resolution 256x256) generated from [30] over residential areas were used. A sample of them is depicted in Figure 3-12. Some other test images (also of resolution 256x256) that did not belong to the training set are illustrated in Figure 3-13 along with their corresponding computed top-down saliency map, *TD(I)*, as defined in Eq. 3-6. Brighter areas in the top-down maps are associated with regions where the underlying feature map values are well aligned with the learned weight vector (i.e., high value for positive weights and low value for negative weights). It can be observed that computed top-down saliency maps can be used for effective object localization which would later be helpful for object recognition. The system identifies houses and streets as

"important" objects, which means that visual properties of these objects were stronger in terms of bottom-up feature maps.

Figure 3-12: A subset of training images that depict top views of an urban residential area.

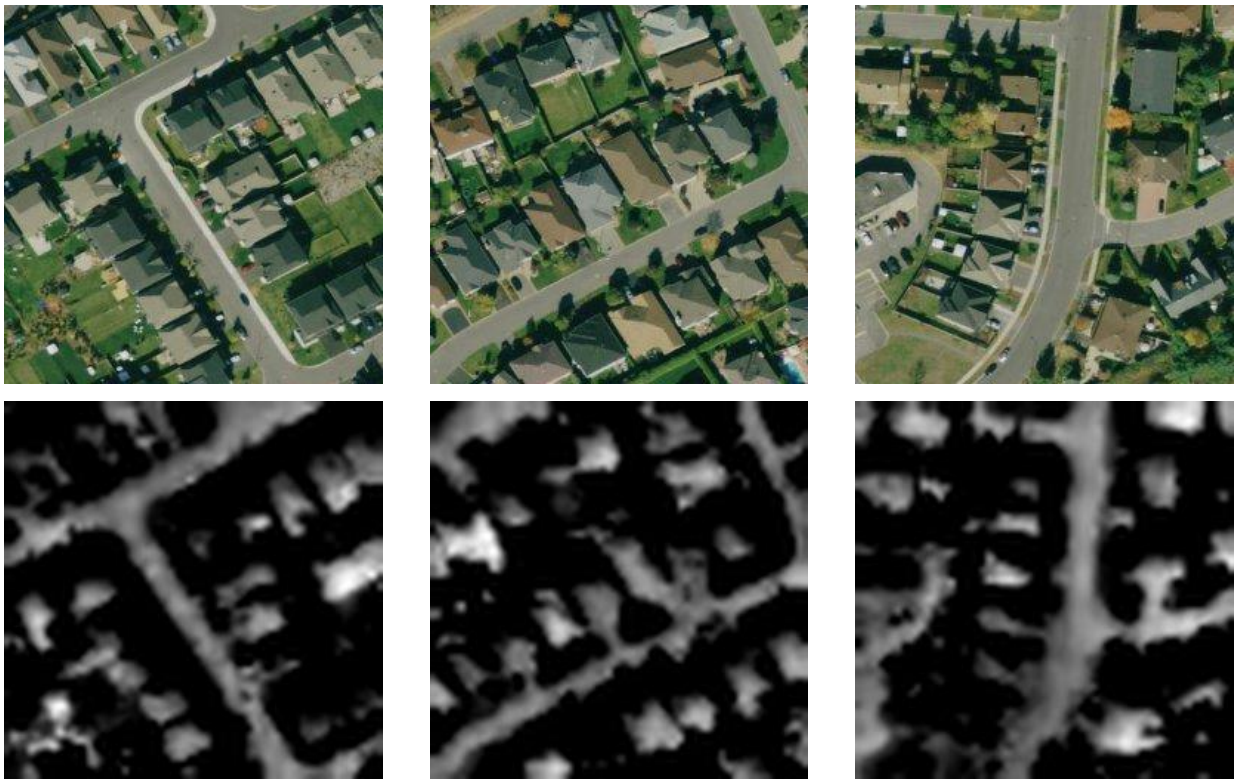


Figure 3-13: Test images (which were not part of the training set) are shown on the top row and corresponding top-down saliency maps on the bottom row. The proposed technique was able to identify most of the objects of interest, here corresponding to houses and streets.

In order to evaluate the generalization capabilities of the proposed energy-based top-down saliency detection method, more experimentation was conducted with images from different classes and at a much larger scale where individual houses or streets are not recognizable. In these cases, recognizable elements are rather city area, river, etc. These images were generated from Google Maps [50] and correspond to various places around Ottawa, Canada. The training set, test images and their corresponding top-down saliency maps are presented in Figure 3-14. The spatial resolution of these images is of 512x512. All computations were done following the procedure described in section 3.4.1.

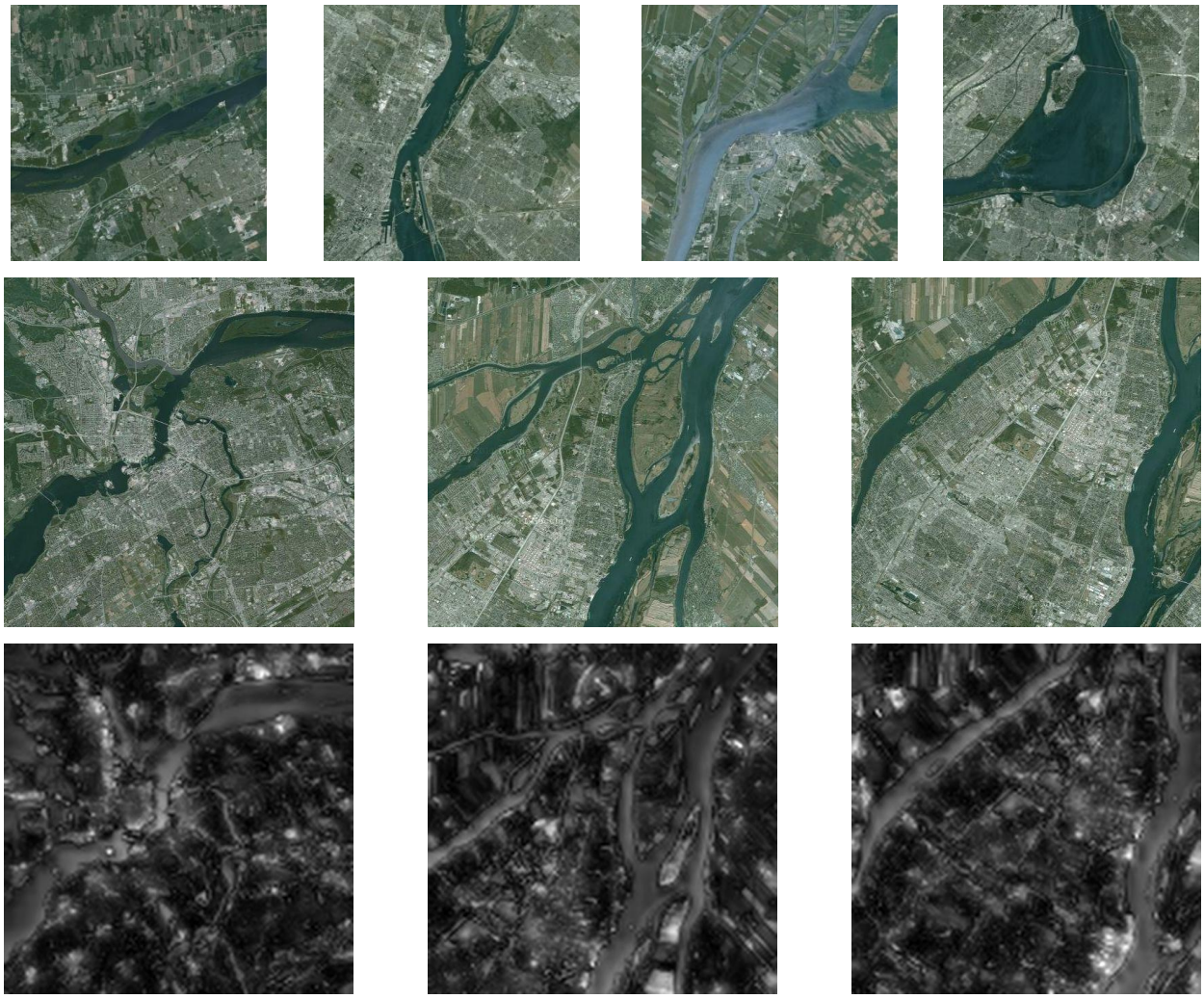


Figure 3-14: Top-row: training set; middle-row: test satellite images; bottom-row: corresponding top-down saliency maps highlighting regions with high energy feature maps.

In the training set (depicted in the top-row of Figure 3-14); bottom-up features related to the rivers and city areas are salient and hence their respective feature maps received larger weights. The test images which are depicted in the middle row of Figure 3-14 belong to the same class of images as of the training set. It can be observed that objects such as most parts of the rivers and city areas are highlighted against more agricultural regions in the final top-down saliency maps (depicted in the last row of Figure 3-14). The features corresponding to built areas were deemed important by the learning system, in proportion with the construction density. Similarly, the rivers with their more uniform visual appearance and directional alignments are assigned relatively uniform energy-based saliency, which make them fairly easy to locate, even for narrower channels of water. Simultaneously, islands are put in evidence. Conversely, the majority of the green fields and regions with low-density construction are inhibited since features corresponding to those regions were identified as less relevant by the system, as they did not cover massive sections of all images used in the training set.

This experimental evaluation demonstrates that the proposed technique for energy-based top-down saliency map computation offers a solid potential in automatically identifying target areas provided that the selected training set reflects the importance of those target regions among other scene elements. It is not required to have specific ROIs associated with each image, as imposed in Frintrop's system. Also the proposed learning approach is not confined to a single MSR. It rather considers the global impact of intensity, orientation and color features and determines their respective weights accordingly. Finally, the computation of conspicuity maps is not required, which contributes to saving valuable computational resources when analyzing massive databases of satellite images.

3.4.4 Top-Down Saliency Map and Entropy Feature

In the previous section, top-down saliency map computation has been performed using only the usual three features of intensity, orientation and color, as retained by Frintrop. In this section, an investigation is conducted to examine whether benefits are provided by the introduction of the extra entropy feature in the computation of the top-down saliency map, as an extension of the idea that was studied in section 3.1 for bottom-up saliency. The introduction of entropy feature would enlarge the size of the weight vector to 12 (it was previously 10), since the entropy feature has two feature maps: one for on-center variation and another for off-center variation. Other energy-based top-down learning related computations are performed following the process described in subsection 3.4.1 with consideration of two extra feature maps. Experimental results obtained with and without the use of the entropy feature are presented in Figure 3-15. The figure considers a set of test images which are processed based on the

training set showed in Figure 3-12. The top row presents the test images. The gray scale images given in the middle row represent the energy-based top-down saliency map computed without entropy feature, and the lower row depicts the top-down saliency map obtained with the addition of the entropy feature.

In all cases the top-down saliency maps that incorporate the entropy feature are more sensitive to small details. They tend to generate maps with several small spikes that correspond to small changes in texture. This happens because of the entropy changes which can then be detected even at fine scales. Another interesting observation is that when entropy is taken into account, highly salient regions correspond to objects' edges, since over edges the entropy changes abruptly. Edges mark the boundary between two regions and two different textures.

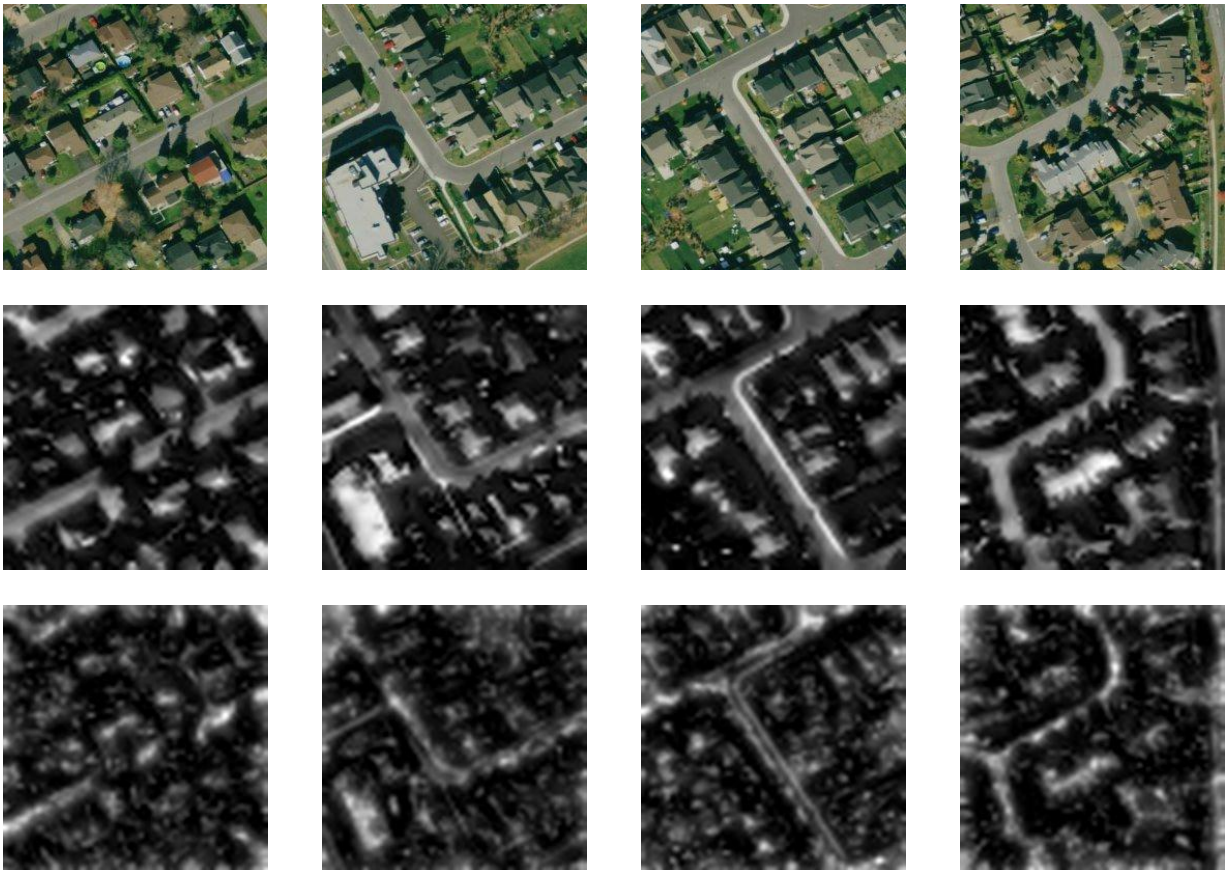


Figure 3-15: Top row: two sets of test images; middle row: top-down saliency maps without entropy; and lower row: top-down saliency maps with additional entropy feature.

3.5 Object Recognition

3.5.1 Process Overview

Beyond the segmentation of an image and the automated location of objects with similar visual appearance, as were discussed in the previous sections, the recognition of objects in a scene is another fundamental problem in image understanding. Usually the first step towards successful object recognition is object localization. In the context of satellite imaging, it was demonstrated that localization can be accomplished with the help of the proposed top-down learning mechanism described in section 3.4. Here, the latter approach is extended to provide a means for reliable recognition of objects in given categories. This work was also partially reported in [6].

To achieve such recognition, the system is first trained with a suitable set of training images according to the procedure described in section 3.4.1, without considering entropy as features. Next, a test image, *I*, is fed in the system and its top-down saliency map, *TD(I)*, is obtained, as defined in Eq. 3-6. The brightest regions in this map are expected to highlight the objects of significance, as detected through energy mapping in the training images. The top-down saliency map is converted to a binary image by replacing small values with 0s and all other values with 1s. The threshold for binarization was determined by trial-and-error and set to 0.25. Figure 3-16 shows some examples of test images that are identical to the ones depicted in Figure 3-13 and the training set considered here is that of Figure 3-12. Next, continuous regions filled with 1s are extracted one after another, thus completing the object localization step. Region extraction is performed by first applying flood-fill on the binary map producing all disjoint regions, followed by the extraction of corresponding regions from the input image. The resulting localization maps displayed on the bottom row of Figure 3-16 are actually binary versions of the top-down saliency maps shown in Figure 3-13. White regions correspond to areas of interests, in this case mainly roads and houses, given the nature of the input images that represent residential areas.

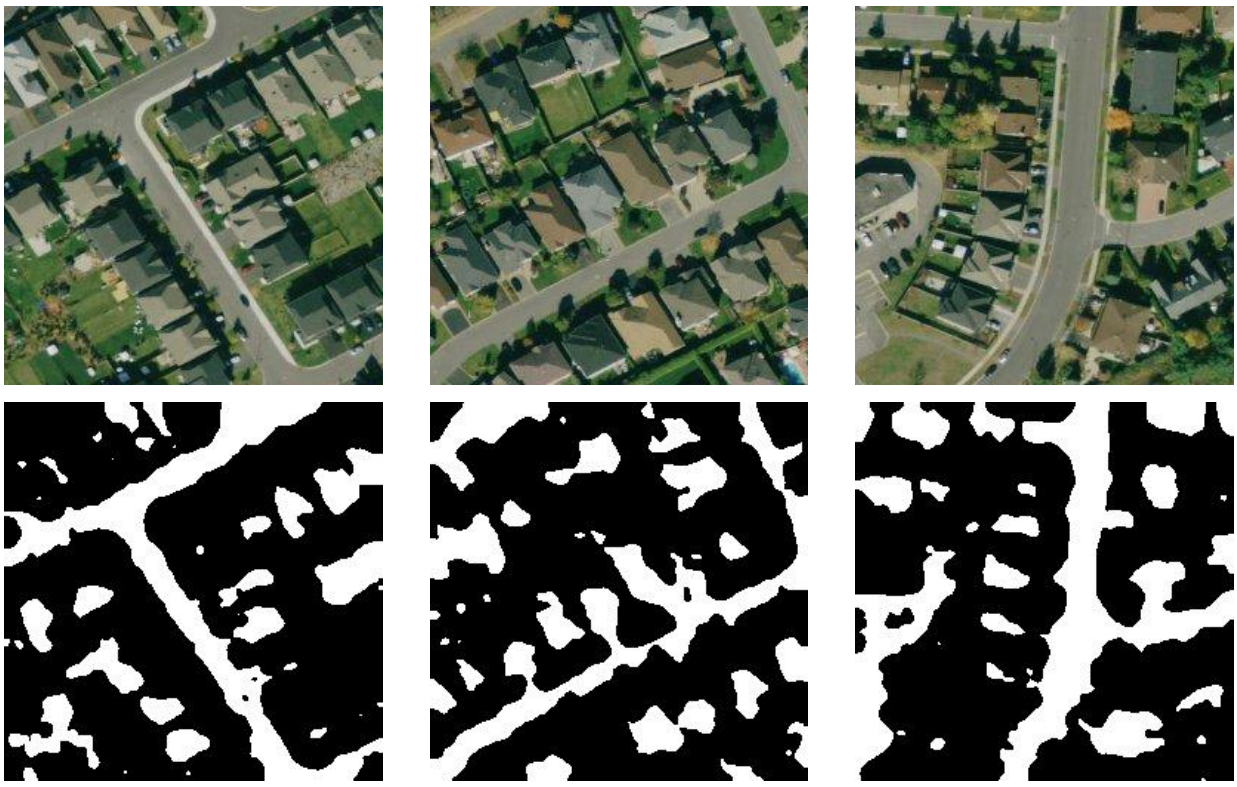


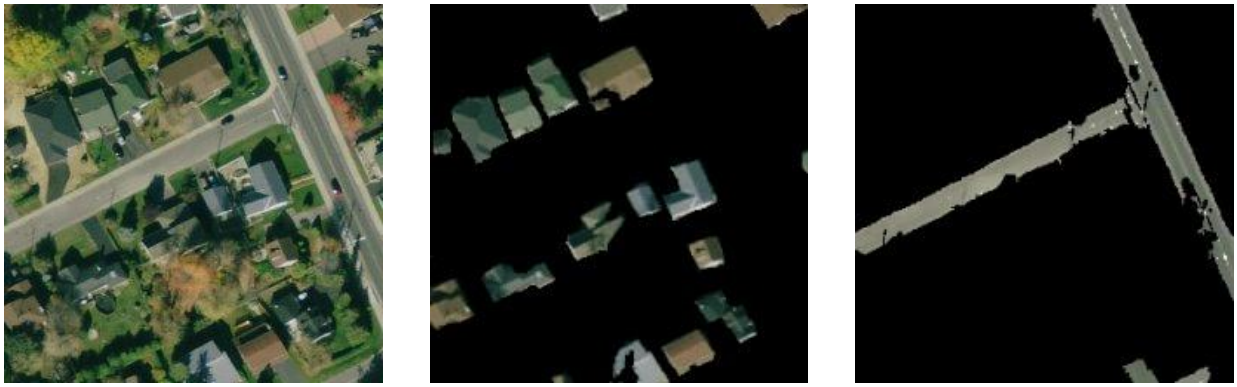
Figure 3-16: Top row illustrates input images and bottom row illustrates the corresponding maps that highlight regions of interest as obtained after binary conversion of the computed top-down energy-based saliency maps.

Following the extraction of regions of interest, further examination is conducted for recognition purpose. The detected regions may encompass different types of objects and in order to classify them an object signature must be known beforehand. The top-down saliency map is solely meant to direct the search towards the interesting regions but does not tell which region corresponds to what type of object. A separate object signature learning procedure is proposed and deployed for this purpose.

It is well understood that both the texture and the shape of an object are important for successful recognition, since two different objects may have similar texture but different shapes, or conversely, similar shape but different textures. In the proposed recognition technique, two measures are used: Local Binary Pattern (LBP) [17] as a texture signature (see section 2.1.5), and image moment invariants [14] as a shape signature (see section 2.1.4). The object of interest signature learning process is detailed in the following section.

3.5.2 Object Signature Learning

Learning the object of interest signature is a separate and independent process whose goal is to describe objects based on their physical appearances rather than on bottom-up or top-down features. Training images for object signature learning are in the form of mask images. Figure 3-17 shows such mask image examples where respectively houses and roads are manually pre-segmented. Mask images specify the shape and texture of the objects of interest in the training images. For example, if one wants to learn roads from satellite images, then the mask images contain only roads and black pixels everywhere else. Mask images are prepared by manual segmentation of training images that are representative of the class. Preferably each class of object is represented in multiple training masks. The system goes through each mask and computes a signature of the object associated with the mask in the form of a vector, and finally computes the resultant vector as a representative signature of that class of object. Computed object signatures consist of texture and shape descriptors and are discussed in the following sections.



a) b) c)

Figure 3-17: a) Original image, b) segmented mask image of houses, and c) segmented mask image of streets.

3.5.3 Texture Signature

Local binary patterns (LBP) are recognized as reliable texture descriptors (see section 2.1.5 for details). They are therefore used in the context of this work as a basis to define a texture signature. Although LBP is usually defined for gray scale texture, it can be extended for three channel color images. In that case instead of one, three LBP maps would be produced. Here, the LBP coding is initially computed over the entire source image for each of its red, green and blue color channels, resulting in three maps containing values in the range [0, 255]. Computing LBP only over the regions of interest would introduce undesirable artifacts since, in the mask images, regions of interest are bounded by black pixels. Next, a

histogram of the LBP values for each color channel is calculated but only over the regions of interest in an image (i.e., where the corresponding mask value is non-zero). Finally, the histogram is converted to a probability density function (PDF) by dividing the frequency of each bin by the sum of all frequency values and a 256-dimensional vector is formed for each color channel. As a result, a 3-color image requires a 768-dimensional vector which would slow down the system severely. To work around this problem, 1-dimensional Legendre moments up to order 10 for each color channel are calculated on top of the histogram PDF using Eq. 2-9. Ultimately, the texture is described by an effective but low dimensional vector.

3.5.4 Shape Signature

Incorporating shape information is advantageous as it can be observed from our everyday life experience. Human beings in many cases can recognize objects even from their outlines only. Moment invariants have been successfully used as shape descriptors in many applications as described in section 2.1.4. In that section, moments for continuous functions are defined and discussed. These definitions need to be adjusted so that they can be applied on discrete data as for images. Discrete versions of the equations for geometric moments are provided below.

Basic geometric moment (GM) of order (p, q)

$$M_{pq} = \sum_x \sum_y x^p y^q f(x,y) \qquad 3\text{-}7$$

Centroid defined in terms of GM

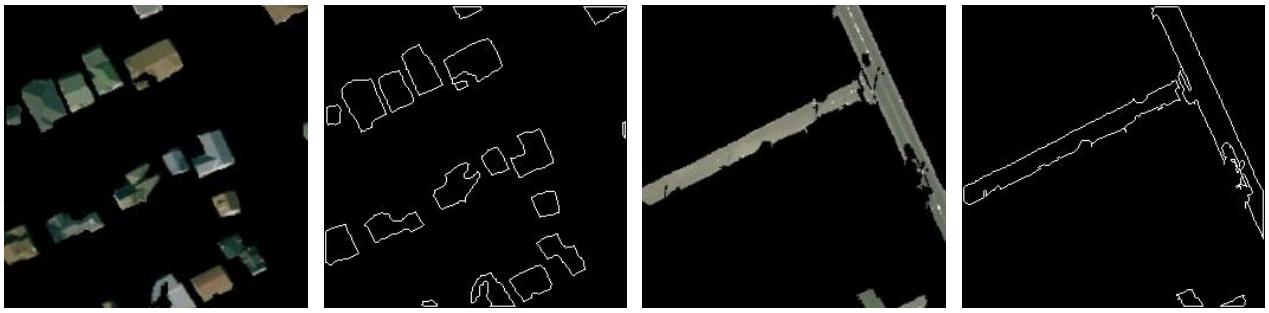
$$\bar{x} = \frac{M_{10}}{M_{00}} \qquad \bar{y} = \frac{M_{01}}{M_{00}} \qquad 3\text{-}8$$

Central GM (w.r.t. centroid)

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \qquad 3\text{-}9$$

Each of the functions needs to be evaluated in order to successfully calculate the required moment invariants. Each function is defined in terms of the one above it. From the training mask images illustrated in Figure 3-17, each contiguous region is extracted and its contours are identified. Figure 3-18 shows examples of such contours for the mask images shown in Figure 3-17. The seven Hu moment invariants are computed from those contours and a 7-dimensional vector is formed for each object present in the mask. Absolute value is taken in the case of the seventh moment invariant to make it mirror invariant as well. Average values of such moment invariants for the contours of objects shown in Figure 3-18 are provided in Table 3-2 to provide an illustration of the intermediate representation of the shapes. Differences between them are distinguishable. Houses, that are closer to a square shape, produce a different distribution of values in the seven Hu moments than streets that are more

rectangular. Finally, the resultants of all vectors are averaged for all objects in a given category to encode the shape of that type of object.



(a) House segments (b) House contours (c) Street segments (d) Street contours

Figure 3-18: Segmented mask images along with the corresponding contours of objects of interest.

Invariant	1	2	3	4	5	6	7
House	0.23	0.03	0.00	0.00	0.00	0.00	0.00
Street	0.64	0.53	0.02	0.02	0.00	0.02	0.00

Table 3-2: Average Hu moment invariants for contours of houses and streets shown in Figure 3-18.

3.5.5 Final Recognition

The vectors representing texture and shape are combined to obtain the final object signature. Learned object signatures are stored in a database. Next, given a test image, its bottom-up feature maps are computed first. Then using top-down weights learned for the class of the training images, its top-down map is obtained and converted to a binary image as detailed in section 3.5.1. The regions extracted from the binary image then undergo the same procedure as for object signature learning and several vectors of unknown classes are obtained from that test image. The database of pre-learned vectors is then iterated once for each of the objects found in the test image and the closest matches are found by means of Euclidean distance. The corresponding objects are then labeled with the color associated to their recognized category. The entire recognition procedure can be summarized using a flow chart presented in Figure 3-19.

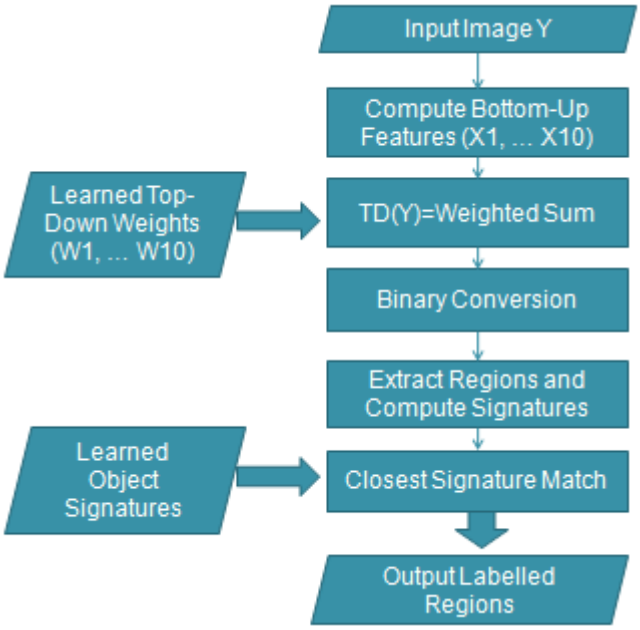


Figure 3-19: Flow chart for the proposed recognition system.

Figure 3-20 shows a test run on the three input images that have been used in Figure 3-16 to maintain consistency. The top row contains the test images, whereas the bottom row depicts the obtained labeled images after recognition. The experiment was performed on a binary case where the goal was to separate houses from streets. Green color signifies streets whereas blue color signifies houses. It can be observed that most of the houses and streets are classified successfully. In a small number of cases, the system gets confused between streets and houses. Texture similarity between roofing and asphalt is the primary reason behind this behavior. But due to the incorporation of shape information, the system also learned that blob-like shapes are houses and longer shapes are streets, which help in the classification. However, in some cases, the binary converted top-down saliency maps contain regions that are thin and long and with a texture similar to that of a street causing the system to classify them wrongly as a street, instead of houses. On the other hand, the shapes of the recognized objects do not match perfectly those of the source image. To overcome this problem, some additional post-processing steps are required and will be added as a future extension to this work. It can also be noticed that the houses with dark greenish roofing were ignored by the system. It happened due to the similarity of the texture to that of grass which was inhibited by the system. The corresponding top-down saliency maps (Figure 3-13) and the binary maps (Figure 3-16) also attest to this.

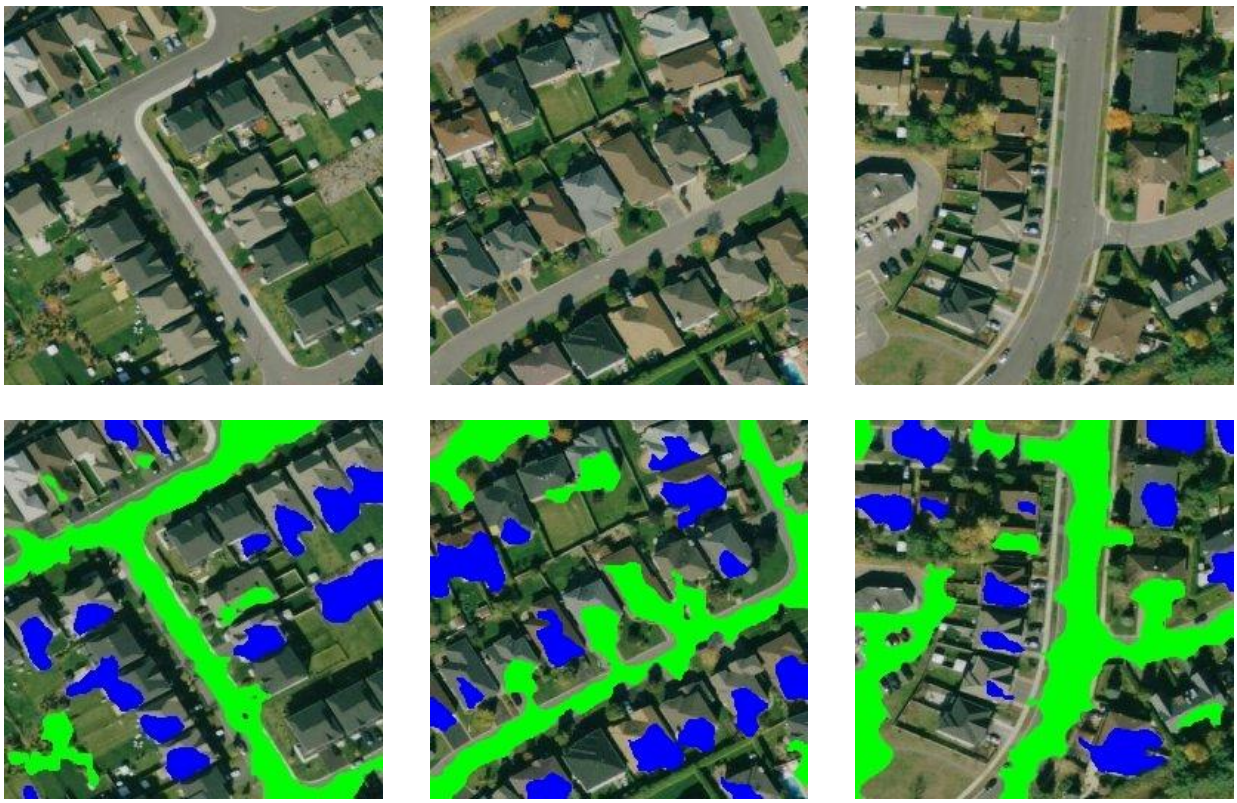


Figure 3-20: Top row: test images that are identical to images on the top row of Figure 3-16. Bottom row: labeled images where green and blue colors represent streets and houses respectively.

Supplementary object recognition experiments were conducted on larger scale satellite images to evaluate the generalization capability of the proposed recognition approach. The results are shown in Figure 3-21 where the top two rows depict two examples of mask images for both city areas and rivers. The training set used for this experiment is identical to those illustrated in the top row of Figure 3-14. The third row of Figure 3-21 depicts test images and the corresponding images in the last row present the recognition results. In the classification, green color corresponds to rivers and blue corresponds to city areas. In order to separate city areas from rivers, similar steps to those described in section 3.5.2, which include mask image preparation for both city areas and rivers and the determination of their shape and texture object descriptors. It is important to notice that city areas and rivers have very diverse shape, unlike the previous cases for houses and streets recognition. For example, the mask images of city areas shown in Figure 3-21(b and e) have highly irregular shapes. In the case of Figure 3-21(b) the masked areas are extended to the image boundaries; which further complicates the situation. Similarly, Figure 3-21(c and f) contain river mask images illustrating serpentine courses of the

river. Also in many cases the textures of cities are not uniform. Cities typically have high density of buildings near the center with a gradual decrease as the distance from the center increases. In some cases, city area can have multiple centers. On the other hand, rivers may have an arbitrary number of branches as can be observed from the images in Figure 3-21. In spite of these large variations is the topology of regions of interest, the proposed recognition method could still correctly recover significant sections of the city, especially the areas with higher density, as well as the river areas.

It can also be observed that, there are a number of misclassifications such as city areas being classified as river and vice versa. In some cases, farm lands also get classified as cities. Beyond the difficulties mentioned above, these erroneous classifications can be associated with the fact that when the region extraction process extracts regions that constitute parts from multiple objects, the computed object descriptor is a hybrid descriptor and hence fails to concretely represent any of the real objects. Regions are extracted directly from the binary image which is generated from the computed top-down saliency map based on a static threshold (0.25) and hence it is likely that nearby potential target objects get connected to each other. This produces misclassified objects and hence the presented quantitative performance ratio in the next section may not be in accordance with all classes of images. In order to alleviate the misclassifications in these erroneous cases; region extraction can be made more intelligent by incorporating object edge information on top of the top-down saliency map instead of simply thresholding the map. This can lead to a more reliable region extraction procedure that would be able to better recognize the objects in question. On the other hand the classified images can be used as high level guidelines and the classified regions can be further examined using more zoomed images of the respective regions. A more zoomed image will reveal more details and the texture and shape are expected to be more defined. A richer database containing object descriptors for possible objects present in a class of image would also help in better classification rate; since an unknown object will get the label of the nearest object in the database. Overall, further refinement is necessary in order to overcome the limitations.

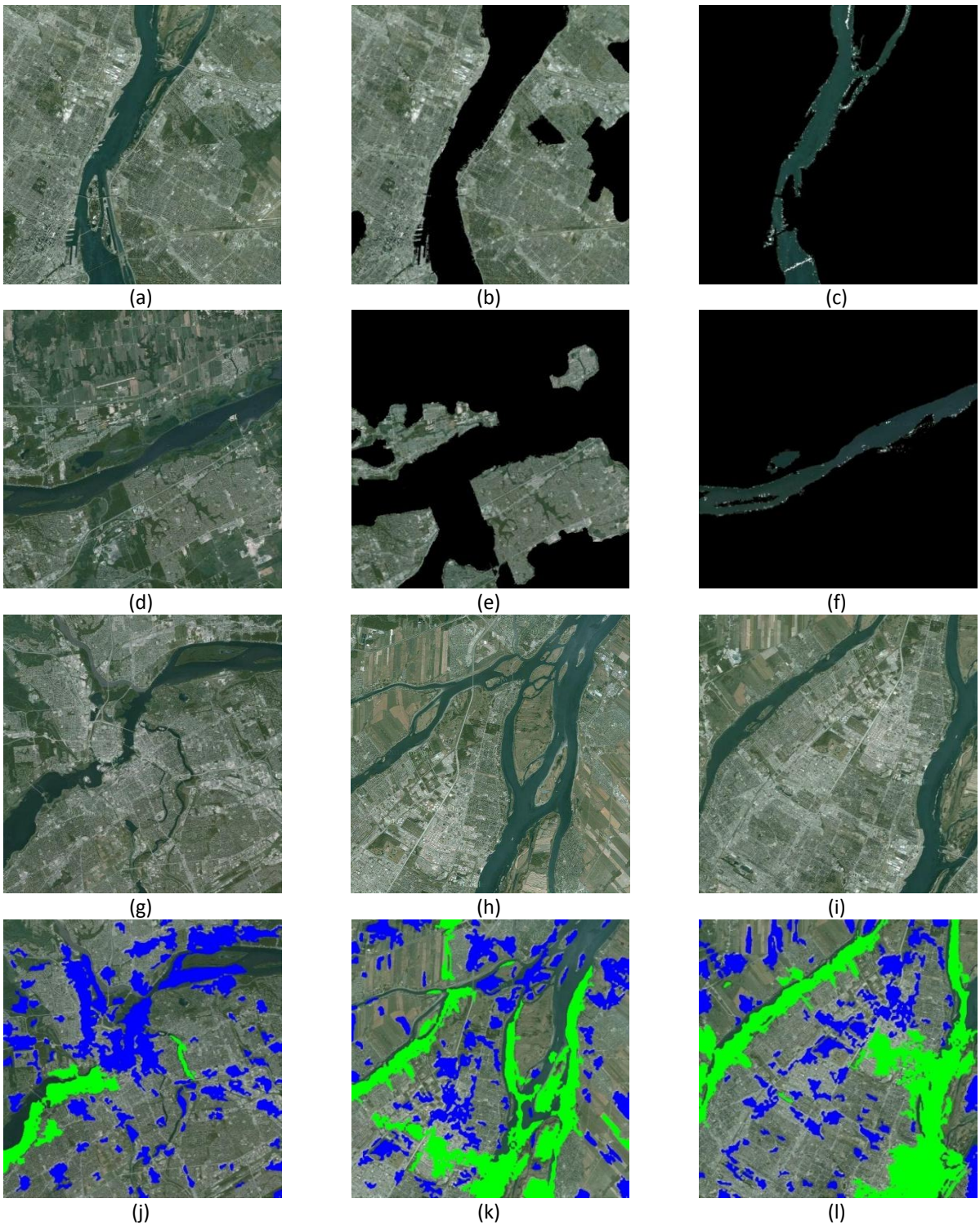


Figure 3-21: First and second row: training images (a and d), along with their mask images for both city areas (b and e) and river (c and f); Third-row: large scale satellite test images. Bottom row: recognition results where green and blue colors represent river and city areas respectively.

3.5.6 Performance Analysis

3.5.6.1 Classification Rate

In order to derive a quantitative performance evaluation of the recognition technique described in section 3.5.5, the following criterions are defined. The ground truth of number of houses present in each test image is manually counted and the total street area in pixels is computed from the mask images of streets. For each test image, the correctly classified houses are also manually counted from the labeled output images. Correctly classified street areas are determined as follows: The intersection between the ground truth street mask image and the total classified street areas is measured (in pixels) for each test image.

The results from ten test images on which houses and streets were searched for and recognized with a system that was trained with the training set depicted in Figure 3-12, are reported in Table 3-3. A more elaborate illustration of results along with training images and their corresponding mask images are provided in Appendix C.

Image Number	Ground-truth Houses [nb of houses]	Correctly Classified Houses [nb of houses]	Ground-truth Street Area [pixels]	Correctly Classified Street Area [pixels]
1	16	7 (43.8%)	6316	5539 (87.7%)
2	13	5 (38.5%)	8652	5131 (59.3%)
3	19	9 (47.4%)	8786	6849 (78.0%)
4	20	11 (55.0%)	8524	7787 (91.4%)
5	9	7 (77.8%)	6137	5181 (84.4%)
6	6	4 (66.7%)	16235	12300 (75.8%)
7	8	7 (87.5%)	9501	7715 (81.2%)
8	19	12 (63.2%)	1959	1704 (87.0%)
9	8	5 (62.5%)	5770	3926 (68.0%)
10	10	7 (70.0%)	3944	3418 (86.7%)
Total	**128**	**74 (57.8%)**	**75824**	**59550 (78.5%)**

Table 3-3: Performance of proposed recognition method when applied on satellite images of residential in search for houses and streets.

Over the entire set of test images considered, globally 74 out of 128 houses (57.8%) are correctly detected, which corresponds in fact to an average image-based success rate of 61.2%. Similarly, for street areas, 78.5% of total street areas are correctly classified by the proposed recognition technique, which corresponds to an average image-based success rate of 80.0%. It is noticeable that correct classification rate of streets is higher than that of houses. The reason behind this is the following. Houses have very diverse shape and texture whereas streets are more uniform in texture and shape.

Also there are shadows associated with almost every house due to their inherent elevation, which is not the case for streets. Shadows tend to have a significant impact on satellite image analysis independently of their content. Overall, the performance achieved represents an improvement over the industry reported performance which lays around 30% to 40% on an average [51]. This shows that the proposed system offers a promising direction to further develop satellite image processing techniques that will better meet the needs of the industry.

3.5.6.2 Comparison against Frintrop's System

In Frintrop's system, an operator must manually select a ROI from every single training image and be able to decide how large or small the ROI needs to be in order to obtain possible optimum result. Also, at a given time only one target area can be learned and searched. In Frintrop's system, the performance criterion is defined in terms of hit-number which is eventually defined as the rank of the MSR within the computed top-down saliency map that corresponds to the target object. For example, if the system is trained to detect a key; then in a top-down saliency map computed from a given test image; all top MSRs are found. Among those MSRs, the rank of the MSR that is the order in decreasing saliency values that correspond to a key is declared as the hit-number for that test image. Ideally the hit-number would be 1 for a successful identification of the target object, which means that the object of interest would specifically be the one with the highest saliency over the entire scene. Obviously this can hardly be the case for any realistic scene, but still the hit-number must remain as low as possible for the method to be of any value. It is important to notice also that the original Frintrop model does not verify that the found MSR really corresponds to the subject of interest based on the physical appearances (e.g., a key shape). Also multiple appearances of the target object within the test image are not considered. By definition, the system does not attempt to recognize all target objects present in a given test image. The quantitative performance measure reported in Frintrop's work is in terms of average hit-number which can be as low as 1.09 (i.e., the object of interest is the true dominant object), and as high as 6.91 (i.e., the object of interest gets detected only in the 6 or 7 positions in terms of dominant saliency). The performance varies drastically for different objects and according to the complexity of the scene considered. However, none of the tests reported by Frintrop are related to satellite images, and the images considered are largely of much lower complexity in terms of contents.

The characteristics of Frintrop's method make it difficult to perform a one-on-one comparison with the proposed object recognition technique. Notwithstanding this, the latter brings several benefits over the original approach. First, the technique introduced in this chapter is unique in the sense that strong

features can be automatically separated from weaker features. As a result, there is no need to specify regions of interest to start from for each of the training images, which otherwise requires operator intervention. The method directs itself toward most salient regions based on the estimation of the energy inherently contained in the feature maps. Second, the proposed approach attempts to recognize all targeted objects that are present in the input image without requiring successive searches for lower and lower top-down saliency values in the map. This brings a major advantage for large satellite images that systematically contain several occurrences of objects belonging to a same category (e.g., houses, roads, lakes, forest areas, fields). Finally, the method not only attempts to detect possible locations of the target objects but also attempts to label them based on pre-learned object signatures that are characterized by shape and texture, as suggested by many biological instances of visual perception. The differences are summarized in Table 3-4.

Frintrop's System	Proposed System
ROI selection is required.	ROI selection is not required.
Single target object learning and searching.	Multiple target objects learning and searching.
Highly dependent on specific background contents.	Not dependent on specific background rather on a class of backgrounds.
Object signature is based on MSR: a small region.	Object signature composed of both texture and shape.
Multiple appearance of a target object is not considered.	Multiple appearances of target objects are inherently considered.
Does not physically verify that the generated MSR really corresponds to the target object.	Attempts to verify object signatures.

Table 3-4: Frintrop's system against the proposed system.

3.5.6.3 Comparison against Recent Results

There are classical techniques that can be specialized to detect a particular object. In [52], authors presented a building recognition technique on NADIR images. The proposed technique first segments the input image by reducing its dynamic range, followed by removal of vegetation and shadows. Finally, building recognition computation was done as follows. The input image is entropy filtered and thresholded. The resulting image is then distance transformed and finally segmented using watershed technique. Different regions are extracted from the segmented image and their corresponding solidity value is computed which was defined as the ratio of the area of the region to the area of the convex hull of that region. The regions with high solidity value (> 0.7) were declared to be buildings. This implies that buildings are assumed to have convex shape.

The approach presented in [52] is highly specialized for recognizing only buildings exclusively. Also, the resolutions of the input images were very high comparing to the current case. The above mentioned paper also compares its results with other techniques for building recognition. Table 3-5 summarizes the performance measures of related techniques as reported in [52] along with the results achieved in this thesis. Row 1 (technique "Shorter") provides performance measure of the above mentioned procedure and row 2 depicts performance measure of a related technique mentioned in [52]. The bottom two rows represent the techniques where object signatures were either only based on texture or shape (see section 3.5.7 for more details).

	Technique	Correct Classification Rate (House)	Correct Classification Rate (Street)
1	Shorter	48.2%	X
2	Liu	83.4%	X
3	Proposed	57.8%	78.5%
4	Proposed-Texture only	4.7%	84.1%
5	Proposed-Shape only	72.7%	68.6%

Table 3-5: Performance measure comparison.

It can be observed that, the proposed technique (3rd row) provides a well balanced classification rates for both houses and streets comparing to the other techniques. Especially, the house recognition rate is higher than that of Shorter's approach. The classification rate for houses using only shape information (5th row) is close to Liu's approach which is exclusive to house recognition. Moreover, it provides high classification rate for street recognition. The texture-only technique (4th row) performs extremely poorly on houses whereas it performs well on street classification due to the fact that house textures in many cases are very similar to that of streets (further discussed in section 3.5.7).

3.5.6.4 Computational Complexity

The proposed method involves several components that have different computational requirements. There are learning processes that operate separately and independently from the recognition processes. The computational complexity also largely depends on contents of the scene. For example, in the recognition phase, the number of times the object signatures are computed is roughly equal to the number of objects present in the image. Also the size of each object influences the total time it takes, since a larger object requires more time to be processed. In general, the number of objects and their respective sizes are unknown; hence it is not possible to derive a deterministic time complexity for the entire system. On the other hand, some parts of the system involve more predictable complexity.

Finally, there are also some fixed time processing components. For example, in order to compute Legendre moments, a pre-computed table is required which is a fixed (or no) cost since it can be computed off-line. A computational complexity analysis is proposed below for the main functionalities discussed in this chapter.

The computation of intensity feature maps represents a computational complexity of O(WH) where W and H are the width and height of the input image, respectively. At first, an integral image is computed which takes O(WH), followed by two applications (for two different radii) of center-surround algorithm which also have O(WH) complexity each. Finally, all computations are performed on a constant number of scales of a pyramid; hence the total time complexity is O(WH) for the intensity features. Similarly, the complexity of the orientation and color feature maps computation is also O(WH) for each, but with larger associated constants since they need more iterations (four maps instead of two maps). The uniqueness function, fusion of different feature maps, normalization, and maxima determination, bear O(WH) complexity. Finally, computing the bottom-up saliency map requires O(WH) time since the elements that are required are applied a constant number of times in order to obtain the final saliency map.

In the top-down weights learning phase, at first a bottom-up saliency map for each image in the training set is computed. This is followed by energy computation for each feature map, which takes O(WH). The transformation of weights takes a constant time, O(1). Hence the learning phase takes O(nWH) in total where n is the cardinality of the training set.

Total number of pixels in an image having width, W, and height, H, is WH and usually all pixels must be visited in order to solve a given classification or recognition task. Hence, from the perspective of image processing, a time complexity of O(WH) is usual and is considered as efficient since it means a constant number of operations per pixel. The proposed methods do not differ much, in terms of computational complexity, from classical image processing techniques, which represents an interesting advantage.

3.5.7 Elements of Object Descriptors

In the proposed object recognition technique, local binary pattern and Hu moment invariants were amalgamated in order to effectively describe the objects' appearance. In this section, the results obtained when using only one of the descriptors are compared against the performance achieved when using both a texture and a shape descriptor. In Figure 3-22, the first, second, third and fourth rows respectively show the input images, the recognition achieved with only the use of the Hu moments

shape descriptor, the recognition with only the use of LBP texture descriptor, and the recognition achieved when using both descriptors. Here again, the training set considered is that of Figure 3-12.

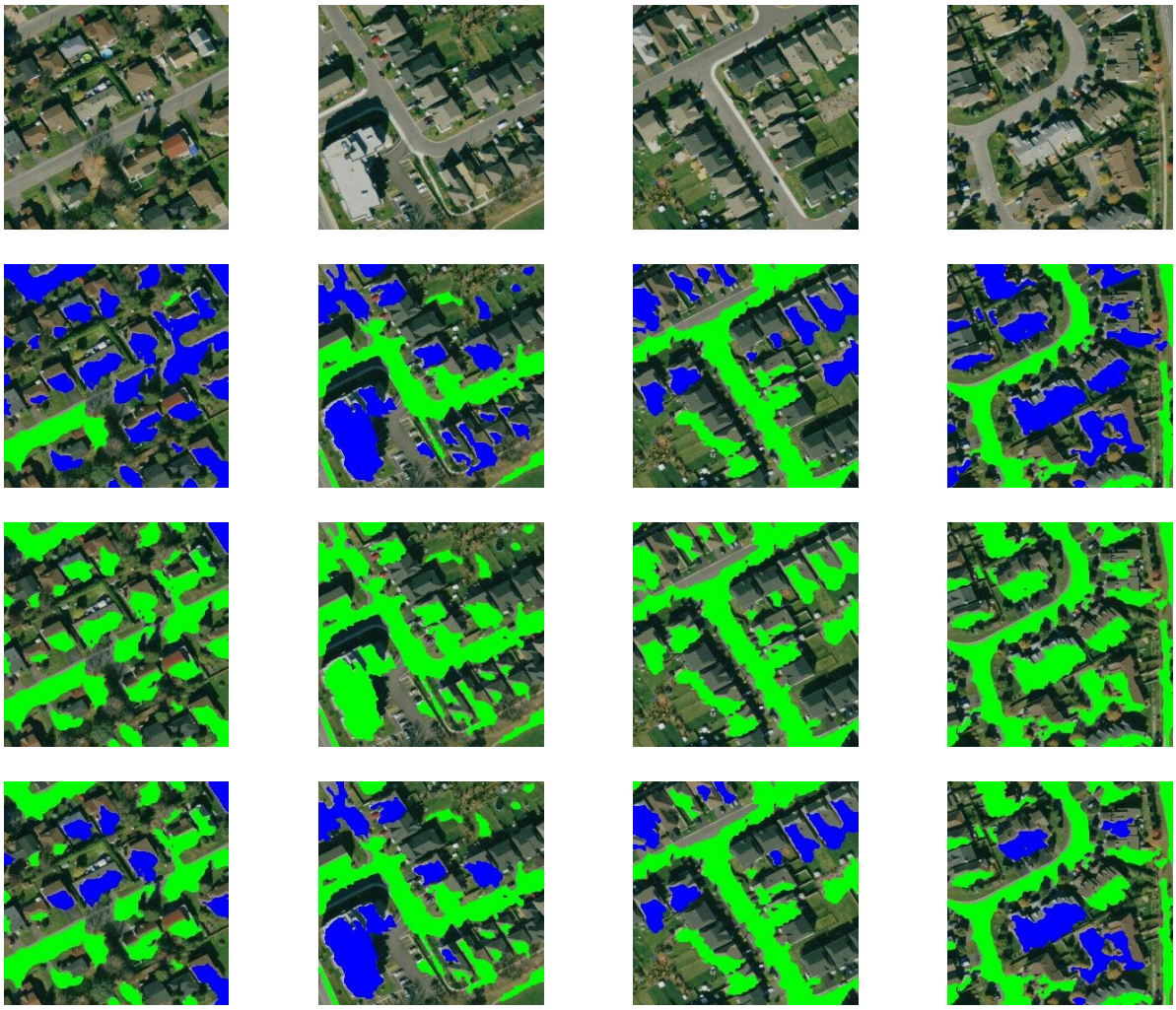


Figure 3-22: Different combinations of object descriptors for the recognition of houses (blue) and streets (green). First row: test images; second row: recognition with Hu moments shape descriptor only; third row: recognition with LBP texture descriptor only; fourth row: recognition with both descriptors.

In order to quantitatively evaluate the impact of the use of the shape descriptor and that of the texture descriptor, Table 3-6 and Table 3-7 compare the classification accuracy against the ground truth for the same set of 10 images considered in Table 3-3 and depicted in Appendix C. Using only the shape descriptors encoded in Hu moment invariants, 72.7% of the houses (or 74.7% on average image-based)

and 68.6% of the street areas (or 63.2% on average image-based) are accurately classified. On the other hand, using only LBP texture descriptors, the experiments reveal that only 4.7% of the houses (or 5.3% on average image-based) and 84.1% of the street areas (or 63.2% on average image-based) are classified properly.

As reported in Table 3-3, when using both the shape and the texture descriptor simultaneously, the classification performance reached 57.8% for houses (or 61.2% on average image-based) and 78.5% for street areas (or 80.0% on average image-based). With shape descriptors only, a significant improvement is observed on house recognition, but simultaneously street area classification is degraded. Using only LBP texture without shape information gives very poor results on house classification. As exemplified in Figure 3-22, the system classifies almost everything as streets and that is the reason for such a high street classification rate. On the other hand, using only shape descriptors classifies most of the objects as houses since houses have better defined shapes. This also supports the higher house classification rate achieved with shape descriptors only. In both situations, the winning category (house or streets) adversely impacts, to various extents, the opposite category. Let A and B be the correct classification rates of houses and streets, respectively. Then Shannon's information content based on these two variables, calculated as: $-(A \log_2(A) + B \log_2(B))$, are 0.724 (both texture and shape), 0.701 (shape only) and 0.429 (texture only). Thus using both measures gives the highest information content which means that statistically this would provide more reliable results than either one alone.

Image Number	Ground-truth Houses [nb of houses]	Correctly Classified Houses [nb of houses]	Ground-truth Street Area [pixels]	Correctly Classified Street Area [pixels]
1	16	15 (93.8%)	6316	2184 (34.6%)
2	13	8 (61.5%)	8652	5083 (58.7%)
3	19	11 (57.9%)	8786	6849 (78.0%)
4	20	12 (60.0%)	8524	6681 (78.4%)
5	9	8 (88.9%)	6137	5181 (84.4%)
6	6	5 (83.3%)	16235	12300 (75.8%)
7	8	5 (62.5%)	9501	7714 (81.2%)
8	19	14 (73.7%)	1959	350 (17.9%)
9	8	6 (75.0%)	5770	2518 (43.6%)
10	10	9 (90.0%)	3944	3137 (79.5%)
Total	**128**	**93 (72.7%)**	**75824**	**51997 (68.6%)**

Table 3-6: Performance of proposed recognition method with only Hu moment invariants for shape signature when applied on satellite images of residential areas in search for houses and streets.

Image Number	Ground-truth Houses [nb of houses]	Correctly Classified Houses [nb of houses]	Ground-truth Street Area [pixels]	Correctly Classified Street Area [pixels]
1	16	0 (0.0%)	6316	5539 (87.7%)
2	13	0 (0.0%)	8652	6533 (75.5%)
3	19	0 (0.0%)	8786	6991 (79.6%)
4	20	1 (5.0%)	8524	7787 (91.4%)
5	9	0 (0.0%)	6137	5816 (94.8%)
6	6	0 (0.0%)	16235	12300 (75.8%)
7	8	2 (25.0%)	9501	8639 (90.9%)
8	19	2 (10.5%)	1959	1873 (95.6%)
9	8	1 (12.5%)	5770	4877 (84.5%)
10	10	0 (0.0%)	3944	3418 (86.7%)
Total	**128**	**6 (4.7%)**	**75824**	**63773 (84.1%)**

Table 3-7: Performance of proposed recognition method with only LBP texture signature when applied on satellite images of residential areas in search for houses and streets.

3.5.8 Object Recognition and Entropy

The effects of the incorporation of the entropy feature when performing object recognition as proposed in section 3.5 are inevitable, since object recognition is based on the top-down saliency map in the proposed approach. Since top-down saliency maps vary with the incorporation of the entropy feature, the corresponding binary maps vary accordingly. This eventually affects the extracted regions. Hence object localization is different. Figure 3-23 shows a direct one-to-one comparison between cases where entropy is incorporated in top-down saliency computation and where it is not.

For each input image in the top row, the middle row depicts recognition result achieved without entropy, and the lower rows show recognition results with the entropy feature incorporated. The input images used are identical to that of Figure 3-15. It can be observed that the recognition results obtained with the incorporation of the entropy feature are somewhat different. The shapes of recognized objects are narrower than when the entropy is not included. Also there are more holes within the recognized objects when using entropy. However, in both cases, this problem can be overcome by using morphological operations (e.g., hole-filling). The consideration of entropy provides more conservative pointers toward objects of interest, which is beneficial for the detection of large streets, but in some cases also seems to degrade the recognition of smaller objects, in this case exemplified by houses.

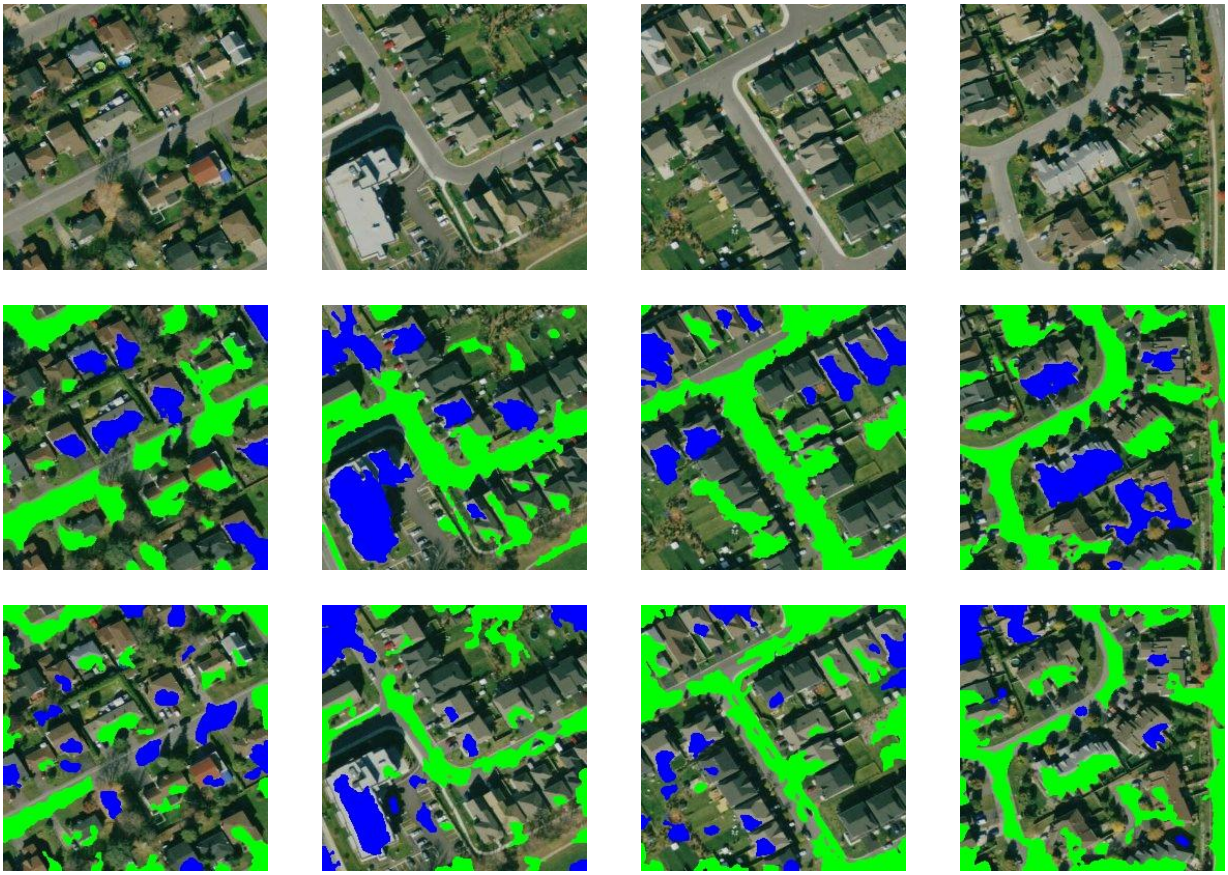


Figure 3-23: Effects of the entropy feature on houses (blue) and streets (green) recognition. Top row: two sets of test images; middle row: recognition from top-down saliency maps without entropy; lower row: recognition from top-down saliency maps with entropy.

3.6 Summary

In this chapter, computational visual attention techniques have been explored, experimented and their potentials evaluated in solving real world problems related to satellite image understanding. This work expands on Frintrop's system [4] which exploits the power to three classical features namely, intensity, orientation and color.

On top of these three features, an entropy feature is incorporated within the framework of Frintrop's system. Entropy measurements are associated with on-center and off-center entropy variations. It is observed that the addition of entropy in bottom saliency maps makes the system more sensitive to fine variations that are present within the scene contents. This capability can be exploited in a situation where a detailed analysis of a scene is needed. An application of bottom-up saliency in satellite image segmentation is also presented and it is demonstrated that the technique is capable of producing plausible segmentation of satellite images. The use of Legendre moments of histograms of color texture, to compactly encode the visual appearance yields to an efficient computational approach to encode complex object descriptors for segmentation purposes. The technique is further extended with the introduction of a method to search for objects of interest through the use of bottom-up saliency and with the help of various image moments. It is observed that the bottom-up saliency of an object varies according to where it is placed (background dependence) and hence it is not a viable standalone solution to search for an object in a background independent manner when only relying on saliency.

In order to overcome the limitations mentioned above, a novel top-down saliency map computation technique was presented. The latter can automatically separate important features from unimportant ones from a given training set of images and without the need of ROI specifications for each image in the training set. Once trained, given a test image, it can automatically excite the important features and inhibit the weak features, thus highlighting desired objects. This technique is especially useful in handling large satellite images since it is capable of locating several target objects in a single pass, therefore relieving the recognition system from looking everywhere and only investing computational efforts on selected areas of the image. As an immediate extension, an object recognition system is introduced that uses the proposed top-down saliency system to first locate target objects within the input image and then deploy the recognition procedure only in those locations. The recognition procedure attempts to recognize all present entities of the target objects at once and labels them accordingly. The system advantageously uses both texture and shape information to describe an object, which has shown to be effective. For the texture component, local binary patterns compactly encoded

by Legendre moments are used. On the other hand, the shape is described by Hu moment invariants. It was experimentally demonstrated that using only shape descriptors can provide satisfactory results, unlike the use of texture descriptors alone. The effects of using only one of the two descriptors, or the combination of both are also studied and demonstrated that the use of both descriptors is more reliable to detect object with various characteristics.

Overall, the presented techniques provide a contribution to the general area of image understanding and especially in the context of satellite imaging. The proposed framework provides a comprehensive image analysis strategy which includes novel methods for defining and learning object features and signatures which are then integrated for final recognition over large images with complex contents. In that, the proposed approach differentiates itself from previous works found in the literature where computational visual attention is often applied to fairly simple scenes.

4 Implementation

In this chapter, the design of the implemented software that supports the functionalities discussed in Chapter 3 is presented along with a discussion about the related issues. The details of key functions are provided. The tools used in the implementation are mentioned along with references to them.

The software is carefully designed so that it retains modularity, extensibility and maintainability in order to make future development in this research track easier. The system works as a framework for further development and hence experimenting with different features and algorithms would be possible to conduct without hassle. Integration of more functionality is also possible under the umbrella of the framework. Many functionalities of the software are made possible to override. For example, adding or removing a feature, redefining importance of a particular map, using different strategies for map fusion, would be programmatically trivial. Besides, several low level operations are implemented in a generalized fashion and are readily available for implementing complex operations.

This chapter, which focuses on implementation particularities, is provided as a complement to the detailed presentation of the proposed algorithms in Chapter 3, with the goal to support future developers who will further expand the biologically-inspired processing platform that was initially developed in the scope of the current research.

4.1 Implementation Environment

4.1.1 Computational Platform

The results presented in this monograph are produced with software that was implemented using the C++ programming language and object oriented approach. The Open Computer Vision (OpenCV) [53] library version 2.3.1 was used for solving fundamental image processing issues such as reading/writing images from disk files, convolving an image with a filter, or image normalization. The software was compiled using the 64-bit version of Microsoft Visual C++ 2010 Ultimate (MSVC) compiler. The 64-bit version allows faster processing since the software can exploit all the enhanced features of 64-bit microprocessors such as an increased number of general purpose registers, and enhanced SIMD (Single Instruction Multiple Data) instructions for floating point processing. Most of the function parameter passing in 64-bit system is done through registers rather than using the program stacks; reducing memory access and hence increasing performance. The compiler can take care of all low level details and use the advanced features.

In some cases, functions are made parallel using Open Multi Processing (OpenMP) [54] library (version 2.0) which can utilize all the available microprocessor cores allowing much faster processing. Parallelization is especially desirable in the current context since satellite images have very high spatial resolution and can be of very large dimensions, reaching up to 30000 x 30000 pixels in some applications. Throughout the developed application there are several opportunities for parallelization. For example, the computation of different feature maps is independent of each other and hence can be performed in parallel. Lower level operations, such as operations at each scale of a pyramid, window-based operations (e.g., local binary patterns), and computation of moments of a function at different orders, can be accomplished in parallel. Given an increased number of available microprocessor cores, the performance of the system will be much better. Using parallelization is highly advantageous especially since there is no quality penalty. Obtained results are identical to those obtained by serial implementation. However, one needs to be careful at how parallelization is done in order to gain optimal performance. If parallelization is applied at every level then the system would have too many internal threads which would take significant amount of time to be scheduled and managed properly especially when few microprocessor cores are available. Moreover, the creation and destruction of threads take significant amount of time (fixed cost). In order to maintain a good balance, often parallelization is applied only at the high level.

In the implemented software, parallelization is applied while computing different features since feature computations are independent of each other. It was found out that, on an average 40.9% performance gain was received on a processor having two cores while computing bottom-up saliency map of a given input image. It is almost never possible to obtain full performance gain due to thread synchronization and scheduling as mentioned earlier. In the current context, some features require more time to compute as discussed in section 3.5.6.4 and hence the threads handling computationally lighter features (e.g., intensity) finish first and have to wait for the threads that are handling computationally heavier features (e.g., color). Obtained results using both parallel and serial approaches were identical in quality. Overall, the acquired performance gain is still desirable.

In a different perspective that is also relevant for the work discussed in this thesis, it is well known that floating point numbers used in microprocessors (or in the Floating Point Unit – FPU) tend to loose precision. Specially, single precision (4-byte float data type) is very unreliable. To alleviate the effects of this problem, 8-byte (double precision) floating point (double data type) is used to represent all images, maps, filters, intermediate values, and local variables. Table 4-1 summarizes the specifications of the

environment that was selected for the development of the computational platform that supported the experimental components of the work.

Tool Type	Specification	Comment
Compiler	Microsoft Visual C++ (MSVC) 2010 Ultimate	One of the best IDEs for development in windows environment
Target architecture	64-bit	Allows faster processing
Image Processing Library	OpenCV 2.3.1	Free, open source and reliable
Parallelization Library	OpenMP 2.0	Standard and supported by MSVC
Floating point precision	Double precision	More accurate, since lots of operations have to be performed.

Table 4-1: Computational platform specifications.

4.1.2 OpenCV

OpenCV is a widely used, free, open source and well documented image processing and computer vision library. It has implemented a wide range of general purpose functionalities starting from low level to high level. It allows computation for standard image processing procedures such as filtering, finding min/max, normalization, or color space conversion. Among mid-level processing, notable functions are Fourier transformation, finding contours, and geometric moment computations. It has also implemented quite a number of machine learning techniques, such as neural networks, and Bayesian learning. It also supports reading and writing of many types of image formats such as JPEG, PNG, BMP, among others. The library has functionalities to read and write AVI (Audio Video Interleaved) video files. It also provides basic user interface elements, but with limited capabilities. In many cases it is sufficient.

OpenCV is written in C/C++ and has a very simple architecture and hence is easy to use. It contains a small number of classes and is mostly function oriented. OpenCV provides few classes (both template and non-template) for representing any matrix. For example the template class Mat_<T> represents a matrix of any type T, whereas the class Mat represents a matrix whose data type has to be provided by OpenCV defined constants, e.g., CV_8UC3 and CV_32FC3 define three-channel 8-bit integer or 32-bit (single precision) floating point matrices respectively. OpenCV does not always support double precision floating point. For example, the function that converts a color image to gray scale or a BGR (default channel order is blue-green-red rather than red-green-blue) image to CIE LAB image, does not support double precision. In those cases, the functions needed to be written from the scratch to prevent loss of precision in the results.

4.1.3 OpenMP

OpenMP is an open standard for software parallelization and is supported by many compilers including MSVC. It is unique in the sense that the library is built-in to the compiler. No explicit thread management, e.g., creation/destruction, work sharing, or scheduling is necessary. Programmers only specify the parallel sections of the software along with some other small number of parameters, e.g. private and shared variables. On the other hand, it is not possible to control the worker threads at the low level using OpenMP. A comprehensive treatment on OpenMP can be found in [55]. An interesting advantage of programming with OpenMP is that it can be used with existing serial code to make it parallel with minimal changes and in most cases just by adding a few lines of code to define the parallelization components.

4.2 Software Modules

One of the main advantages of using C++ programming language is software modularity and reusability [56]. In C++ this is achieved through the use of classes. A class is an abstract data type that represents a programming element, idea or concept. It encapsulates data and its related code and gives the sense of data-code unity. A class typically exposes different interfaces in order for its clients to perform some operations on the data that it encapsulates. A class can use other classes (in other words becomes a client) through pointers, inheritance, or object compositions, and the interfaces are in the form of different types of functions. The overall design of the implementation realized for this research work, along with key functions implementation, is described in the next sections.

4.2.1 Overall Design

The design of the system is simple and intuitive. All classes in the system are derived from a root base class (either directly or indirectly through C++ inheritance mechanism) which defines and declares several elements that are common to all other classes. The elements that are needed in several places such as different image pyramids and feature maps are encapsulated in classes. Each type of image pyramid is represented by an individual class and all of them are derived from one base class. A similar strategy is taken for feature maps. The class which is responsible for bottom-up saliency computation stores objects of feature classes. The inheritance relationships among classes can be described using an UML class diagram. A significant portion of the class diagram is illustrated in Figure 4-1 which depicts the class hierarchy. One interesting aspect can be observed from the diagram, which is that the saliency class is also derived from the feature class. This is because the saliency map of a scene is also a feature of that scene. The primary advantage of using such a hierarchical class structure is that it allows addition

of more software features easily. For example, in order to experiment with the entropy feature (see section 3.1), one needs to derive a class from the class named Feature (see Figure 4-1) and add the necessary code to compute an entropy map (and a few other households). There are other classes that help in accomplishing other tasks such as image moment computation, and complex sinusoid computation. In the following sections, specific details about the key functionalities are provided.

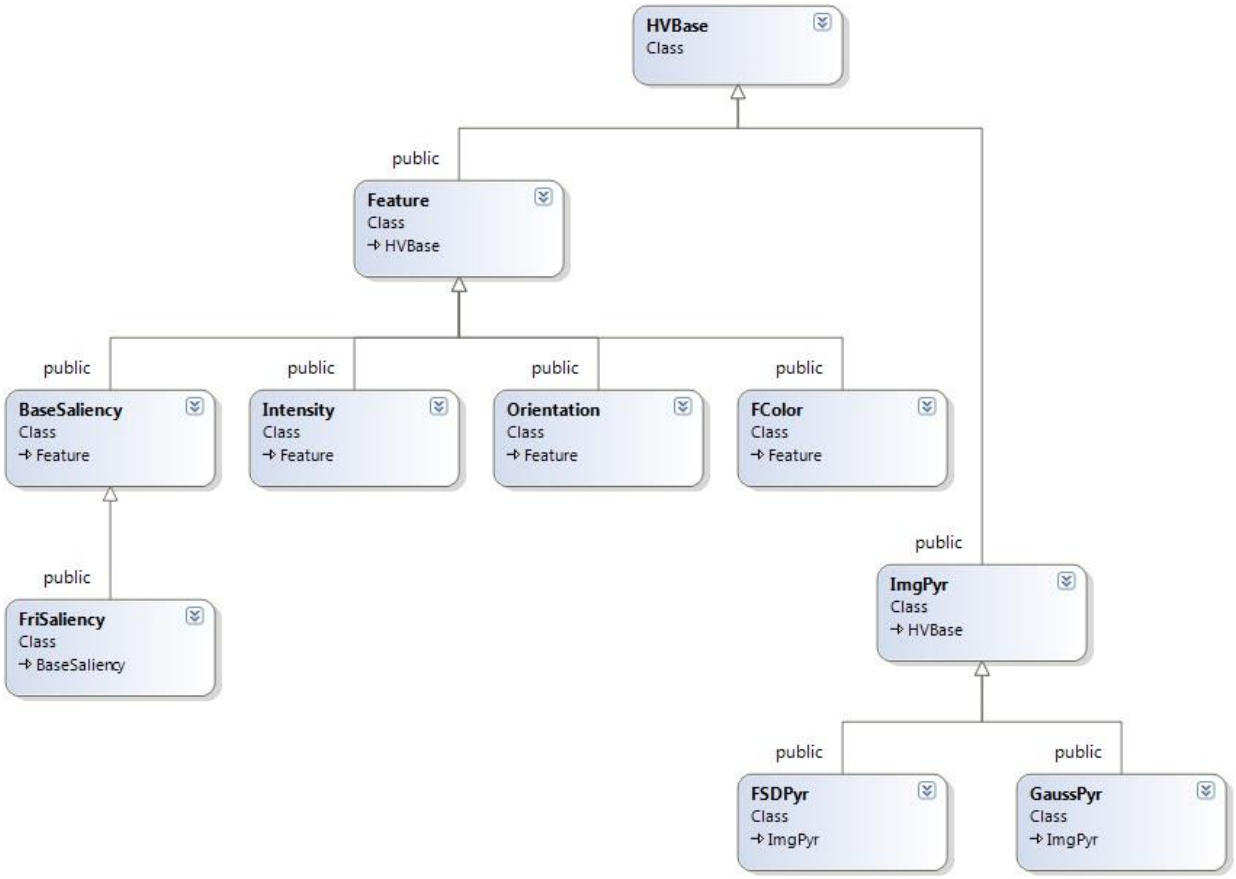


Figure 4-1: UML class diagram of a significant portion of the system.

4.2.2 Details on Key Functions

4.2.2.1 Computation of Center-Surround

The application of center-surround is ubiquitous throughout the saliency computation technique. It is applied on intensity, color and entropy maps at each layer of the pyramid (see sections 2.2 and 3.1). Computation of center-surround is expensive if achieved through the brute force method as described in the above mentioned sections. The actual computation is rather performed through the use of an integral image [57], defined in Eq. 4-1, which is actually a two-dimensional cumulative sum of an image.

$$I(x,y) = \sum_{u \le x, v \le y} f(u,v) \qquad 4\text{-}1$$

This particular representation of an image allows a much faster technique to compute the sum of values in a window of any size within the image using only two addition and two subtraction operations; whereas the total number of addition operations using a brute form method would be equal to the number of cells in the window in question. Let a window, *D,* of size *WxH* in the image have the upper-left coordinates of *(x, y)*. Then, the sum of the values in that window using the integral image, *I(x,y)*, can be found using Eq. 4-2.

$$S(D) = I(x-1, y-1) + I(x+W, y+H) - I(x+W, y-1) - I(x-1, y+H) \qquad 4\text{-}2$$

The primary operation in center-surround computation is finding the mean of values in a sliding window for all possible positions. The computation is done as follows: Given a map, first its integral image is computed using Eq. 4-1 (OpenCV provides the function *cv::integral()* for that purpose). After that, for all possible positions for the sliding window, the sum of values within that window is found using Eq. 4-2 (in constant time, O(1)) and successively the sum is divided by the number of pixels within the window.

4.2.2.2 Computation of Local Maxima

Computation of the local maxima over a map (or a function in general) is an important step in Frintrop's system, which is used in the weighting function (uniqueness function) defined in Eq. 2-13. OpenCV was not found to provide any function to compute such measurements. The computation of local maxima is performed over the 3x3 neighborhood around each pixel. If no neighbor is found with a larger value than that of the center pixel, then the center pixel is declared to be a locally maximum value; otherwise not. Equation 2-13 also requires the local maxima to lie within a certain range and this is accomplished by checking the center pixel against the range before deciding if the pixel is a local maximum value at all. In the computed local maxima map, the local maxima are replaced with 1s and non-maxima are

replaced with 0s. This helps in counting the number of local maxima, which can be achieved by taking the sum of the local maxima map.

4.2.2.3 Legendre Moments of Histogram

The use of Legendre moments is rife throughout this monograph, especially for histogram compacting. There are a few steps involved in computing Legendre moments of orders up to *k* of a histogram, *H*, of size *n*. Table 4-2 illustrates the work out for a fabricated histogram of size 5.

	Legendre Polynomial	H=>	0.30	0.10	0.25	0.15	0.20
			0	1	2	3	4
		Moments	-1.0	-0.5	0	+0.5	+1.0
$P_0(x)=$	1	1.0	+1.0	+1.0	+1.0	+1.0	+1.0
$P_1(x)=$	x	-0.075	-1.0	-0.5	0.0	+0.5	+1.0
$P_2(x)=$	$(3x^2-1)/2$	0.344	+1.0	-0.125	-0.5	-0.125	+1.0
$P_3(x)=$	$(5x^3-3x)/2$	-0.144	-1.0	+.875	0	-.875	+1.0

Table 4-2: A solid workout illustrating steps of Legendre moments computation up to orders 3 of a histogram of size 5.

The table shows histogram values on the top row and the following row depicts their usual indices in the array and is followed by transformed indices in range [-1, +1]. The four lower rows show the first four Legendre polynomials along with their values evaluated at those transformed indices. These values are computed using Eq. 2-12 and saved in a 2D array for future use. This pre-computed table is valid as long as the histogram size is kept identical. When the size of the histogram is changed, the table has to be recomputed for a new set of values. If higher order moments are necessary then more rows have to be added at the end for higher order polynomials. The third column from the left shows the actual moments. These moments are computed by taking dot product between the top row (which is the histogram itself) and one of the rows containing the evaluated polynomial values. Hence a large histogram is represented using a small number of moment values (here four), thus achieving compactness for faster processing.

4.3 Graphical User Interface

A graphical user interface (GUI) enhances the usefulness of a system. For this purpose a GUI was developed which interacts with the underlying system. The Qt [58] framework was used for the purpose of GUI development and it provides a simple interface to perform experiments. An overview of Qt framework and the developed GUI are provided in next sections.

4.3.1 The Qt Framework

Qt is a cross-platform, object oriented and well-documented software development kit (SDK) having several parts such as core GUI library, networking (TCP/IP, UDP) library, and a thread management library. A comprehensive treatment on Qt can be obtained from [59]. A large collection of different GUI elements have been implemented and the framework allows developers to implement new GUI elements with ease. All Qt classes are directly or indirectly derived from a root class called *QObject* and all GUI classes are derived (directly or indirectly) from *QWidget*. The *QWidget* class provides lots of virtual functions that can be reimplemented in the derived class to achieve the desired functionalities. One of the problems in any GUI framework is to make a system that would let GUI elements communicate efficiently with other GUI elements or the code associated with them. In Qt this is achieved by the so called signal-and-slot mechanism. Any Qt class can declare its probable signals and expose it to be used by other Qt classes. Then, any other Qt class interested in receiving such signals can create one or more slots. Finally, these signals and slots can be connected by a Qt function called *QObject::connect()*. For example the button class can have a signal called “clicked”, which is broadcasted whenever the user clicks on the button. Any other class which is interested in the “clicked” signal of that button can create a slot and connect to it. This mechanism is generalized since any Qt classes can communicate with any other Qt classes. Another issue in any GUI library is to provide a mechanism to arrange a group of GUI elements on the screen. The Qt framework provides several classes that help in solving this problem. It provides several styles that can be combined with each other to create complex arrangements.

Unfortunately, there is an incompatibility between Qt and OpenCV regarding internal image format. Qt cannot work with gray scale images directly. It uses an indexed technique for this purpose, whereas handling gray scale image is trivial with OpenCV. This problem can be solved by a channel replication technique where the same gray channel is replicated to make it appear as a three-channel image. The final outcome is inevitably gray even if it has three channels since each component of a pixel has identical value. On the other hand, Qt uses red-green-blue channel order by default for color images

whereas OpenCV by default uses blue-green-red. In order to display any OpenCV image within the context of Qt user interface elements, one must alter the default channel order of OpenCV image by swapping the blue and red channels. This can be done by using the cv::cvtColor() function provided by OpenCV with the parameter cv::CV_BGR2RGB.

4.3.2 The User Interface

The developed user interface has a main application window with different tabs for different options. A tabbed structure enables inclusion of several user interface elements within a limited space by showing only the relevant GUI elements for the current context. Using the developed GUI it is possible to locate one or more image files within the computer secondary storage media by clicking the related buttons. The use of picture widgets especially designed to work with OpenCV's image data structure is ubiquitous. Labels are commonly used to indicate the purpose of a particular UI element. Other UI elements such as scroll bar or list are used to enhance the usefulness of the software. A few snapshots of the GUI are provided below. The tab that deals with the bottom-up saliency is shown in Figure 4-2. The button labeled "Browse…" is used to select the input image. The "Compute" button serves to start the saliency computation. Other buttons are also provided to accomplish tasks such as viewing conspicuity and feature maps or saving maps. The image on the upper left is the input image in this case and its saliency map is displayed directly below it. The bottom right image is showing the top three focus of attention regions. Red, green and blue represent the first, second and third focus of attention regions respectively. The slider bar on the right of the "Compute" button can be used to choose the threshold involved in deciding the size of the focus of attention area. Similarly the scroll bars (both vertical and horizontal) can be used to scroll the entire viewing area.

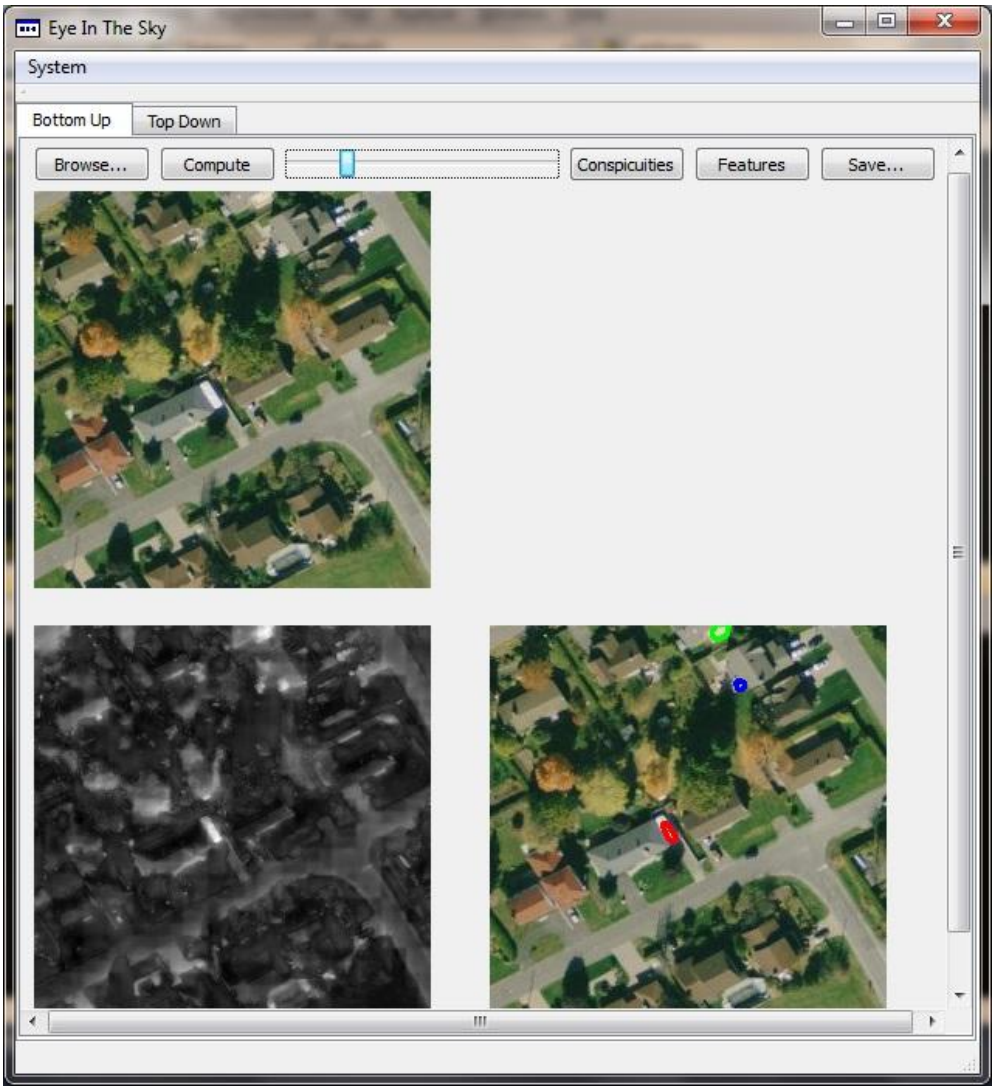


Figure 4-2: A snapshot of bottom-up saliency GUI.

In Figure 4-3 another snapshot of the tab that deals with top-down saliency is provided. The list is showing the selected test images. The picture on the right corresponds to the selected file on the left, therefore making the navigation through a database of satellite images straightforward. Buttons for both learning and top-down (see section 3.4) saliency computation are provided as well. Other buttons to display learned weights and save a computed maps are provided to make the system comfortable to use.

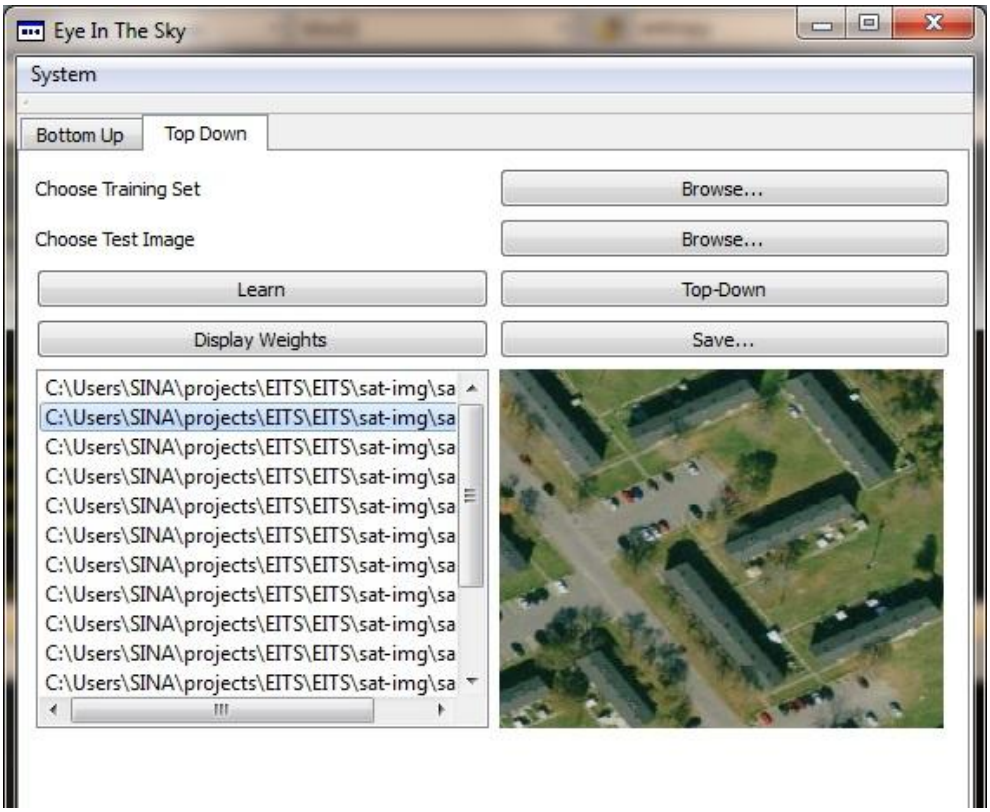


Figure 4-3: A snapshot of top-down saliency GUI.

5 Conclusion

5.1 Summary

In this monograph, the potential of biologically-inspired computational image processing techniques, especially visual attention, was explored for advancing satellite image understanding capabilities. One of the latest such systems (VOCUS) was studied in details and improvements to that system were proposed. Several other techniques such as moments of functions and local binary patterns are taken advantage of in the proposed improvements. A series of experimental results illustrate the potential of the proposed methods for effective satellite image understanding.

It was shown that the original VOCUS learning mechanism, which assists in top-down saliency map computation, is background dependent and the reasons behind such dependence were unearthed. As an alternative, an expanded top-down saliency computational approach was proposed. The incorporation of the entropy feature on top of the intensity, orientation and color features was studied. It was shown that changes in image texture usually result in changes in entropy as well. It revealed that such incorporation permits the computation of more detailed bottom-up saliency maps. A novel application of bottom-up saliency for automatic image segmentation was also devised. It was demonstrated that such a technique produces satisfactory image segmentation based on the visual appearance of image regions. Furthermore, it was shown that in order to search for a target object by means of bottom-up saliency using various image moments produce poor results, since the saliency of an object depends on the presence of other objects. Therefore moments learned in a given context will be different for the same object in a different context. With the objective to overcome this limitation, a novel learning mechanism for the purpose of top-down saliency map generation was devised. The proposed mechanism was rendered to be effective in identifying important features from a set of images and subsequently locate objects with similar features in test images. Finally, an object recognition system was devised that uses the proposed top-down saliency map as a means to localize target objects. The novel approach combines both texture and shape information by means of Legendre moments of LBP and Hu moments respectively in order to recognize those located objects in a computationally efficient manner. To assist the recognition system, an object signature learning technique was described based on mask images. The combination of both texture and shape of objects as their signatures was found to be effective.

Overall, the thesis proposes and experimentally validates a number of biologically-inspired computational methods and visual attention techniques to advance image understanding capabilities in the field of satellite imaging. The proposed approaches overcome some shortcomings identified in Frintrop's VOCUS system. Among these enhanced capabilities, the solution developed in this work allows to automatically retrieve multiple instances of identical objects in an image, a factor that Frintrop's system does not consider, but which proves essential when processing satellite images that cover very large portions of the earth surface. Another improvement is brought to the fact that Frintrop's approach relies on the saliency values only and does not attempt to actually recognize an object based on its physical appearance. The proposed system inherently attempts to first localize and then actively recognize all objects present in the input image and that are compatible with a model that is learned via an innovative representation of top-down saliency. Finally, this thesis illustrates that emulation of the human vision system can be advantageous in advancing general image analysis and understanding and expanding its application to complex satellite images.

5.2 Contributions

The main contributions made in this work towards advancing satellite image understanding by means of biologically-inspired computational techniques and visual attention are summarized below.

This research work contributes:

a) A experimental study on the incorporation of the entropy feature on top of the classical intensity, orientation and color features that are widely used in the literature and implemented in Frintrop's system;
b) A novel approach for automatic satellite image segmentation based on bottom-up saliency map and Legendre moments;
c) An experimental study on the use of various image moments computed on bottom-up saliency for the purpose of object search in satellite images;
d) An original technique for top-down saliency map computation that automatically identifies important regions of interest based on the energy content of feature maps and without assistance from an operator;
e) An novel object descriptor based on Legendre moments of LBP histogram for texture and on Hu moment invariants for shape;

f) An original object recognition technique combining the proposed top-down saliency map computation and joint texture-shape object descriptors;

g) A modular software implementation of classical and innovative biologically-inspired computational methods to support further development of satellite image analysis and understanding.

5.3 Future Work

The field of biologically-inspired computational methods and visual attention is relatively new and hence offers a broad range of opportunities to further develop the capabilities of computer vision, image processing and remote sensing systems. This research has focused on evolving the classical set of features (intensity, orientation, and color) that are currently considered in the literature as being the predominant factors to human visual attention. But other features are definitely worth consideration and examination. For example, motion features can be incorporated in both bottom-up and top-down saliency computation. The proper mechanism to achieve fusion of different features into conspicuity maps or final saliency map is still an open problem. Other fusion strategies besides existing ones can be considered, including various weighting schemes. Probably the most riveting problem in this field is to devise a reliable and efficient technique for top-down saliency map computation. The integration of a representation of knowledge or operator's psychological state, which was demonstrated to have a major impact on how attention and meticulous analysis is directed toward specific areas of a scene, remains an important challenge, especially in a context where the mechanisms of human perception and cognition are not yet fully understood. This will call for truly multidisciplinary research undertakings that will involve computer science, computer vision, human psychology, neuro-computing, and even medical science. As a starting point, testing different ways of combining bottom-up features to compute a final top-down saliency map will definitely be interesting. Experiments can also be conducted on different object descriptors and expands on a wider range of satellite images which would also include multispectral components. This thesis was meant to make a seed contribution in this direction.

REFERENCES

[1] A. M. Treisman and G. Gelade, "A feature integration theory of attention," *Cognitive Psychology,* vol. 12, pp. 97-136, 1980.

[2] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiology,* vol. 4, no. 4, pp. 219-227, 1985.

[3] L. Itti, C. Koch and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. PAMI,* vol. 20, no. 11, pp. 1254-1259, 1998.

[4] S. Frintrop, "A visual attention system for object detection and goal-directed search," PhD Thesis, University of Bonn, Germany, 2006.

[5] M. I. Sina, A. -M. Cretu and P. Payeur, "Biological visual attention guided automatic image segmentation with application in satellite imaging," in *IS&T / SPIE Electronic Imaging*, Burlingame, CA, January 2012.

[6] M. I. Sina, P. Payeur and A.-M. Cretu, "Object recognition on satellite images with biologically-inspired computational approaches," in *IEEE International Symposium on Applied Computational Intelligence and Informatics*, Timisoara, Romania, 2012.

[7] S. Xu, T. Fang, H. Huo and D. Li, "A novel method of aerial image classification based on attention-based local descriptors," in *Intl. Conf. Mining Science & Technology*, 2009.

[8] R. Milanese, "Detecting salient regions in an image: from biological evidence to computer implementation," PhD Thesis, University of Geneva, Switzerland, 1993.

[9] T. Kadir and M. Brady, "Saliency, scale and image description," *International Journal of Computer Vision,* vol. 45, no. 2, pp. 83-105, 2001.

[10] U. Neisser, Cognitive Psychology, New York: Appleton-Century-Crofts, 1967.

[11] R. Desimone and J. Duncan, "Neural mechanisms of selective visual attention," *Annual Reviews of Neuroscience,* vol. 18, pp. 193-222, 1995.

[12] M. Corbetta and G. L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature Reviews,* vol. 3, no. 3, pp. 201-215, 2002.

[13] J. Theeuwes, "Top-down search strategies cannot override attentional capture," *Psychonomic Bulletin & Review,* vol. 11, pp. 65-70, 2004.

[14] J. Flusser, T. Suk and B. Zitova, Moments and moment invariants in pattern recognition, West Sussex, U.K., PO19 8SQ: John Weily & Sons Ltd., 2009.

[15] M. -K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Information Theory,* vol. 8, no. 2, pp. 179-187, 1962.

[16] H. Zhang, H. Shu, G. Z. J. Coatrieux, Q. M. J. Wu, Y. Z. H. Zhang and L. Luo, "Affine Legendre moment invariants for image watermarking robust to geometric distortion," *IEEE Trans. Image Processing,* vol. 20, no. 8, pp. 2189-2199, 2011.

[17] T. Ojala and M. Pietikainen, "Unsupervised texture segmentation using feature distributions," *Pattern Recognition,* vol. 32, no. 1999, pp. 477-486, 1998.

[18] D. Heinke and G. W. Humphreys, "Computational models of visual selective attention. A review," *Connectionist models in psychology,* pp. 273-312, 2004.

[19] C. Siagian and L. Itti, "Biologically inspired mobile robot vision localization," *IEEE Trans. Robotics,*

vol. 25, no. 4, pp. 861-873, 2009.

[20] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Trans. PAMI,* vol. 29, no. 2, pp. 300-312, 2007.

[21] V. Navalpakkam and L. Itti, "Modeling the influence of task of attention," *Vision Research,* vol. 45, pp. 205-231, 2005.

[22] D. Parkhurst, K. Law and E. Niebur, "Modeling the role of saliency in the allocation of overt visual attention," *Vision Research,* vol. 42, no. 1, pp. 107-123, 2002.

[23] S. Frintrop and P. Jensfelt, "Attentional landmarks and active gaze control for visual SLAM," *IEEE Trans. Robotics,* vol. 24, no. 5, pp. 1054-1065, 2008.

[24] B. Rasolzadeh, M. Bjorkman, K. Huebner and D. Kragic, "An active vision system for detecting, fixating and manipulating objects in the real world," *Intl. Journal of Robotics Research,* vol. 29, no. 2-3, pp. 133-154, 2010.

[25] A. M. Rotenstein, A. Andreopoulos, E. Fazl, D. Jacob, M. Robinson, K. Shubina, Y. Zhu and J. K. Tsotsos, "Towards the dream of an intelligent, visually-guided wheelchair," in *Intl. Conf. Technology and Aging*, Toronto, Canada, 2007.

[26] C. H. Anderson, "A filter-subtract-decimate hierarchical pyramid signal analyzing and synthesizing technique". USA Patent 4,718,104, 1987.

[27] A. M. Treisman, "The perception of features and objects," in *Attention: Selection, Awareness, and Control*, A. Baddeley and L. Weiskrantz, Eds., Oxford, Clarendon Press, 1993, pp. 5-35.

[28] J. M. Wolfe, "Guided search 2.0: A revised model of visual search," *Psychonomic Bulletin and Review,* vol. 1, no. 2, pp. 202-238, 1994.

[29] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat," *Journal of Neurophysics,* vol. 28, pp. 229-289, 1965.

[30] "MapQuest," [Online]. Available: http://www.mapquest.ca. [Accessed 25 April 2012].

[31] J. G. Daugman, "Uncertainty relations for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters," *Journal of Optical Society of America,* vol. 2, pp. 1160-1169, 1985.

[32] P. Kruizinga and N. Petkov, "Non-linear operator for orientation texture," *IEEE Trans. Image Processing,* vol. 8, no. 10, pp. 1395-1407, 1999.

[33] H. Greenspan, S. Belongie, R. Goodman, P. Perona and C. H. Anderson, "Overcomplete steerable pyramid filters and rotation invariance," in *IEEE Computer Vision and Pattern Recognition*, 1994.

[34] P. J. Burt and E. A. Adelson, "The laplacian pyramid as a compact image code," *IEEE Trans. Communications,* vol. 31, pp. 532-540, 1983.

[35] D. A. Forsyth and J. Ponce, Computer Vision: A Modern Approach, Berkeley: Prentice Hall, 2003.

[36] W. S. Hungate, R. Watkins and M. Borengasser, Hyperspectral Remote Sensing: Principles and Applications, Florida: CRC Press, 2007.

[37] R. A. Schowengerdt, Remote Sensing: Models and Methods for Image Processing, 3 ed., California: Elsevier Inc., 2007.

[38] R. Szeliski, Computer Vision: Algorithms and Applications, Microsoft Research, 2010.

[39] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3 ed., New Jersey: Prentice-Hall Inc., 2006.

[40] L. Alparone, L. Wald, J. Chanussot, C. Thomas, P. Gamba and L. M. Bruce, "Comparison of pansharpening algorithms: Outcome of the 2006 GRS-S data-fusion contest," *IEEE Trans. on Geoscience and Remote Sensing,* vol. 45, no. 10, pp. 3012-3021, 2007.

[41] I. Saha, U. Maulik, S. Bandyopadhyay and D. Plewczynski, "SVMeFC: SVM ensemble fuzzy clustering for satellite image segmentation," *IEEE Geoscience and Remote Sensing Letters,* vol. 9, no. 1, pp. 52-55, 2012.

[42] F. D. Acqua and P. Gamba, "Texture-based characterization of urban environment on satellite SAR images," *IEEE Trans. on Geoscience and Remote Sensing,* vol. 41, no. 1, pp. 153-159, 2003.

[43] T. Celik, "Unsupervised change detection in satellite images using principal component analysis and k-means clustering," *IEEE Geoscience and Remote Sensing Letters,* vol. 6, no. 4, pp. 772-776, 2009.

[44] D. Fernàndez-Prieto and M. Marconcini, "A novel partially supervised approach to targeted change detection," *IEEE Trans. on Geoscience and Remote Sensing,* vol. 49, no. 12, pp. 5016-5038, 2011.

[45] A. H. S. Solberg and T. Taxt, "A Markov random field model for classification multisource satellite imagery," *IEEE Trans. on Geosciene and Remote Sensing,* vol. 34, no. 1, pp. 100-113, 1996.

[46] Y. Zhao, L. Zhang, P. Li and B. Huang, "Classification of high spatial resolution imagery using improved Gaussian Markov random-field-based texture features," *IEEE Trans. on Geoscience and Remote Sensing,* vol. 45, no. 5, pp. 1458-1468, 2007.

[47] F. D. Frate, F. Pacifici, G. Schiavon and C. Solimini, "Use of neural networks for automatic classification of high resolution images," *IEEE Trans. on Geoscience and Remote Sensing,* vol. 45, no. 4, pp. 800-809, 2007.

[48] L. Shen and S. Jia, "Three-dimensional Gabor wavelets for pixel-based hyperspectral imagery classification," *IEEE Trans. on Geoscience and Remote Sensing,* vol. 49, no. 12, pp. 5039-5046, 2011.

[49] G. Heidemann, R. Rae, H. Bekel, I. Bax and H. Ritter, "Integrating context-free and context-dependent attentional mechanisms for gestural object reference," *Machine Vision and Applications,* vol. 16, no. 1, pp. 64-73, 2004.

[50] "Google Maps," Google Inc., [Online]. Available: http://maps.google.ca/. [Accessed 25 April 2012].

[51] From communications with industrial partner*,* PCI Geometics, 2011.

[52] N. Shorter and T. Kasparis, "Automatic Vegetation Identification and Building Detection from a Single Nadir Aerial Image," *Remote Sens.,* vol. 1, no. 4, pp. 731-757, 2009.

[53] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools,* 2000.

[54] O. A. R. Board, "OpenMP," [Online]. Available: http://openmp.org/. [Accessed 25 April 2012].

[55] B. Chapman, G. Jost and R. V. D. Pas, Using OpenMP, Massachusetts: The MIT Press, 2007.

[56] B. Stroustrup, Programming Principles and Practice Using C++, Boston: Addison-Wesley, 2008.

[57] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, HI, 2001.

[58] "Qt," Nokia, [Online]. Available: http://qt.nokia.com/. [Accessed 25 April 2012].

[59] M. Summerfield, Advanced Qt programming : Creating Great Software with C++ and Qt 4, Boston: Pearson Education Inc., 2010.

Appendix A

Image Pyramid

In image processing the use of an image pyramid for the purpose of achieving scale-invariance is ubiquitous. It is widely applicable in scale-invariant feature detections such as edges and corners. A pyramid consists of more than one layer of different versions of the same input image. The first layer (layer zero) is the input image. Each successive layer can be computed by applying a smoothing filter on its respective previous layer and down-sampling the resulting image by a factor of two in each spatial direction. Hence the image at layer *k* is smoothed *k* times in total. For example, the image at layer 2 is smoothed twice since it is computed from the layer 1 image which is already smoothed once when it was computed from layer zero. At higher layers, image details gradually disappear due to repeated smoothing and their respective sizes become smaller (thereby it looks like a pyramid). Use of pyramid is space efficient since the extra memory required is only one-third (at worst case) of the size of the original image. On the other hand, faster execution is also possible through the use of pyramid since higher layers are smaller in size and require smaller filters in the convolution operations. Pyramids can be of either low-pass or band-pass type. When a pyramid is generated by using a smoothing kernel (e.g., Gaussian), low-pass pyramid is obtained. On the other hand, a band-pass pyramid can be obtained by computing the intensity difference between successive layers of a pyramid. Figure A-1 shows an example of a Gaussian pyramid. The smoothing filter used to generate it is a 3x3 Gaussian kernel and hence the name.

Figure A-1: A Gaussian pyramid.

Appendix B

Flood Fill Technique

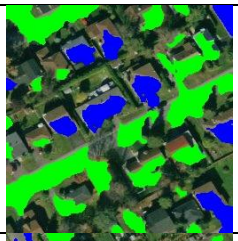
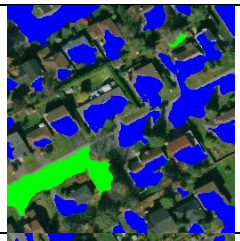
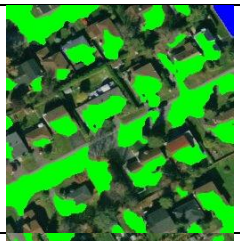
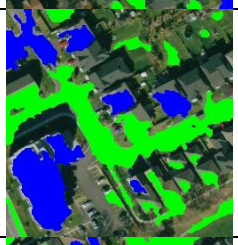
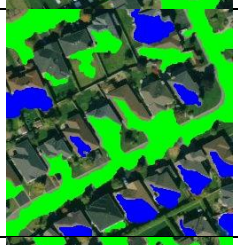
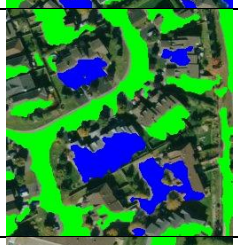
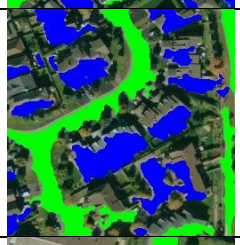
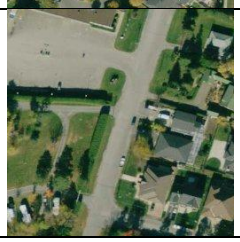
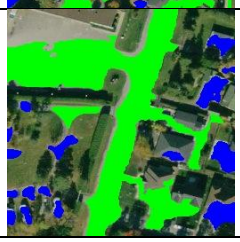
Flood-fill is a well known technique originally used in painting algorithms to fill up the interior of an arbitrary object with a specified color. The key idea behind this technique is to find entirely connected components. The algorithm starts from a seed point and checks if any other point within its neighborhood satisfies the condition to be connected to the seed point. An example of a condition would be the following: If the value of the neighbor is similar to that of the seed within a threshold then they are connected; otherwise not. The algorithm progresses recursively assuming that each of the neighbors satiating the condition becomes a new seed at the next iteration. Hence second level neighbors will be evaluated, and so on. Eventually a cluster of connected points (possibly the entire space) are retrieved. Within this cluster any two neighbors would satiate the given condition. There is a variation to this algorithm. The condition that decides on the connectivity between two points can be defined in two ways. One of them is to evaluate the condition for the current seed and one of its neighbors, and the other way is to evaluate the condition for the root seed (the first seed) with all other points. Depending on the context, both of them are useful. The flood-fill technique is simple but extremely useful in solving related problems. It can be implemented by using either a queue or a stack data structure.

Appendix C

Object Recognition Evaluation

The figures below present the training and testing image sets used in sections 3.5.6.1 and 3.5.7, respectively to experimentally evaluate the performance of the proposed object recognition technique for satellite images of residential areas from which houses and streets are to be classified. Figure C-1 represents the training set along with the corresponding mask images for houses and streets. Figure C-2 presents the 10 test images along with the classification results obtained, respectively, when both the shape and texture descriptors are considered (2nd column), when only the Hu moment invariants shape descriptor is used (3rd column), and when only the LBP with Legendre moments texture descriptor is used (4th column). This figure supports the quantitative recognition rates presented in Table 3-3, Table 3-6 and Table 3-7, respectively.

Figure C-1: Training set images (1st and 4th rows) and manually segmented mask images (houses: 2nd and 5th rows; streets: 3rd and 6th rows) corresponding to experimental results reported in Table 3-3, Table 3-6 and Table 3-7.

Test images	Recognition results with both shape and textures descriptors (Table 3-3)	Recognition results with shape descriptor only (Table 3-6)	Recognition results with texture descriptor only (Table 3-7)
1) 			
2) 			
3) 			
4) 			
5) 			
6) 			

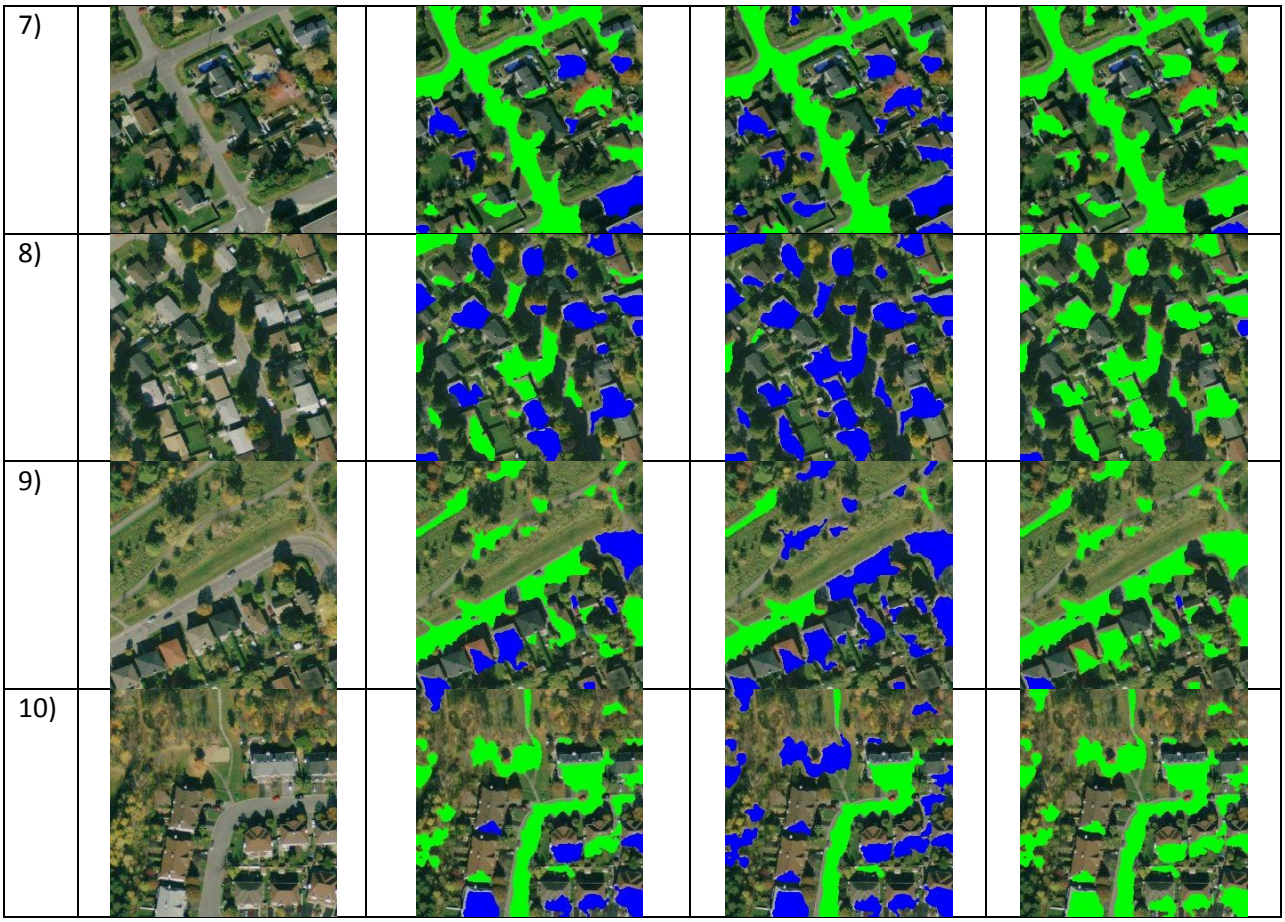


Figure C-2: Recognition results (blue for houses, green for streets) on 10 test images when both the proposed shape and texture descriptors are jointly used, and when only either shape or texture descriptors are considered.