A VISION-BASED ENVIRONMENT ANALYSIS ALGORITHM FOR OPTICAL THREAT DETECTION:

# OPTRO 2010 – 4TH INTERNATIONAL SYMPOSIUM OPTRONICS IN DEFENCE AND SECURITY

OECD CONFERENCE CENTER, PARIS, FRANCE / 3 – 5 FEBRUARY 2010

Vincent Paquin [1], André Morin [1], Jean Fortin [2]

[1] Lyrtech RD, 2800 Louis-Lumière St., Québec (Qc) Canada G1P 0A4, andre.morin@lyrtech.com; vincent.paquin@lyrtech.com

[2] DRDC-Valcartier, 2459 Pie XI North Blvd, Val-Bélair (Qc) G3J 1X5, Canada, jean.fortin@drdc-rddc.gc.ca

## ABSTRACT

This paper describes an image analysis algorithm that is part of an advanced optical threat detection system installed on patrol vehicles.

This algorithm processes wide field of view (WFOV) image sequences captured by a color camera to select regions of the environment that are most likely to contain threats and that will be later inspected by a slower narrow field of view (NFOV) active sensor. A set of features on which the region selection will rely is extracted. Given the limited availability of operation context sequences, a heuristic approach was preferred over learning methods for feature selection and weighting.

The algorithm has primary application in scenes where little *a priori* knowledge is available. The proposed approach was validated on real-world images. Preliminary test results showed that the suggested method can effectively reduce the number of scan areas and that using several redundant cues enables the system to gracefully degrade.

## 1. INTRODUCTION

Equipping military patrol vehicles with systems able to detect optical threat would represent a great tactical advantage, since it will allow convoys to avoid ambushes. The UGLARES project is an advanced optical threat detection system that is mainly based on active sensing. It is composed of a "slow scanning" narrow field of view sensor dedicated to the detection of optical threats, a laser range finder and a wide field of view camera. Image sequences captured by the WFOV camera are used to define a region of interest (ROI) to be inspected by the NFOV sensor. The goal of the algorithm is to optimize the ROI area given the scanning limitations and the likelihood to find threats. This paper presents the image analysis algorithm involved in ROI selection.

The main objective of algorithm is to limit the scene regions to the worthwhile part (less than 30%). Image analysis seeks to identify regions where optical threats are most likely to be found. This is true for regions containing structures like windows, doors and roofs; vehicles and bushes; towers and hilltops, etc. Among the challenges faced towards the objective, let's mention: (1) unconstrained environment; (2) a moving camera; (3) real-time execution; (4) few or inexistent training data, which excludes learning methods; (5) little *a priori* available. The last point is explained by the following facts: the scene is unknown; there are little specific color and shape cues (i.e. as opposed to road panel detection); there is neither specific scale nor motion cue.

Another challenge is that nothing comparable could be found in the literature. Various researchers have reported relative success in the area of building localization from satellite and aerial images [1] [2] [3] and in the area of general building and man-made structures detection [4] [5] [6] [7]. Unfortunately, none are intended for bushes and hill tops detection and most of them are based on learning methods. The work of Krishnamachari and Chellappa [1] which is based on straight lines offers a good starting point.

The approach described in this paper relies on several redundant features extracted from image sequences including lines. The features are extracted at multiple scales and are combined in a probabilistic way. The whole solution is highly heuristic based. An overview of the algorithm architecture is illustrated in Fig. 1. A WFOV RGB image sequence is captured at a rate of 20 fps and digitized by a high dynamic range Bayer color camera at a resolution in the order of 1 megapixel. For optimization reasons, the frames are first sized down to power-of-two dimensions. Intensity statistics are also extracted to allow camera iris aperture control and frame scaling filters the noise and useless high frequencies. The frames are then

partitioned in cells by an optimized quad tree algorithm for further processing. At the next stage, four families of features are extracted from each cell. The first one involves intensity analysis, the second relies on color, the third is about edges and

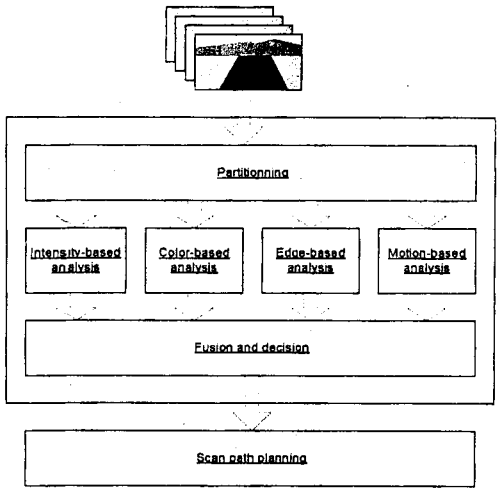


Figure 1. General architecture

the fourth is motion-based. As a consequence of their independence, each family can be processed in a parallel way, which is interesting for optimization on recent multi core platforms. Each feature extraction family will be detailed in section 2. Once all cues are extracted, they are merged to decide which scene region must be inspected by the narrow NFOV system. A region can be composed of one or more cells and a score is attributed to every region for further inclusion in a scan path planning algorithm which is outside the scope of the current paper. The fusion and decision process are explained in section 3.

## 2. FEATURE EXTRACTION

### 2.1. Intensity-based analysis

The first family of features focuses on the intensity of pixel groups instead of their boundaries. This includes skyline detection, trinarization, texture analysis and dark rectangle detection. These steps will be further detailed below.

#### 2.1.1. Skyline detection

The first extracted cue within this family is a rough estimate of the skyline location to be used later for region classification. The estimate is the y-coordinate of a horizontal line that splits the scene in two parts. Cells that are above this line will be treated differently than those below.

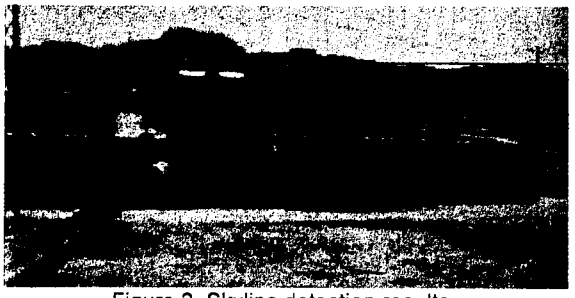

Figure 2. Skyline detection results.

To determine this position, a set of operations is performed. First, the image blue channel is binarized. The result is then projected on the y-axis by summing the "on" pixels on each row and the derivative of the projection is computed to localize transitions. All the derivative values are weighted according to their row position with larger weights attributed to top rows since it is expected that the sky will be found at the top of the scene. The strongest weighted derivative y-coordinate is considered as the skyline position for the current image. An example of skyline detection is illustrated in Fig. 2. Finally, temporal filtering is applied across the sequence of images to smooth out the process.

#### 2.1.2. Trinarization

Another cue comes from classifying the scene pixels according to their tone. Pixels are labeled as light, dark and in-between. A modified version of the Otsu's method [8] is then applied to produce an intermediate label image from which a tone distribution is built for each cell. Cells are then described by their ratios of dark, light and in-between content. Fig. 3 shows a typical scene trinarization example where the day sky is labeled as light, the sand is labeled as in-between and the buildings and hills are labeled as dark. This cue is weak alone but it will be shown that it proves itself useful when combined with others.

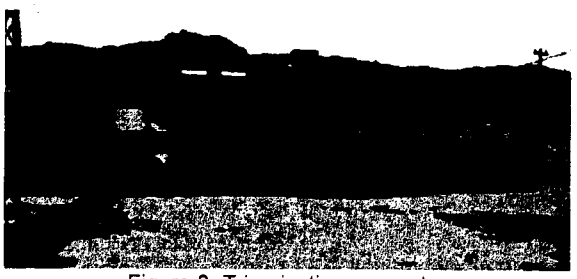

Figure 3. Trinarization example.

#### 2.1.3. Texture analysis

Partition cells are also described by their texture content. Texture features are computed by a signal processing-based algorithm using texture filters applied to the image intensity channel to create

25 filtered grey scale images. Modified Laws energy descriptors [9] are used to produce the intermediate images. Laws introduced 5 one-dimensional kernels that generate 25 two-dimensional convolution masks once combined with each others by means of an outer product. Instead of combining the 2-D features to achieve rotation invariance, it is assumed that the scene will not rotate in the image plane. The input image is then resized to fit the convolution masks dimensions to select the proper range of texture frequencies.

The intermediate images are thresholded and then used to compute two 25-bin histograms for each cell. One distribution represents the ratio of pixels that are above the threshold value and the other increments a bin only when one of the 25 descriptors is above the threshold and all the other descriptors for a pixel. This represents a 50-dimension texture feature vector. Since the scene analysis method presented here is heuristic-based instead of learning-based, there are no concerns about the curse of dimensionality [10].

#### 2.1.4. Dark rectangle detection

Fig. 4 shows a good example of the usefulness of dark rectangle detection. Such an analysis can identify structures as openings in concrete or plaster walls. Experiments made on readily available image sequences showed that the detection of dark rectangles was a lot more reliable and produced far less false detections than light rectangles.

Figure 4. Dark rectangle detection results.

Similarly to Haar-based wavelets described in [11], dark rectangles are defined as rectangular regions in the image that are darker than a proportional border that shares the same center. This implies that the mean intensity value must be computed on two areas centered on each image pixels. Detection is signaled when the difference between the two mean values excess a given threshold. This process must be repeated for all width and height combinations and is very demanding especially for large areas. To improve computation performances, an intermediate integral image [11] is created. This allows computing the sum of intensities in constant time independently from the regions size. Research is performed only for a specific range of sizes and ratios which explains the results for the building in the center of Fig. 4. A partition cell containing a dark rectangle will see its score increased.

### 2.2. Color-based Analysis

The color analysis assigns to each pixel one color from a palette of seven. The color palette is composed of grey, yellow, green, cyan, blue, magenta and red. First, the input image is converted from RGB to HSV color space which has the advantage of expressing the color information independently from the intensity. Each pixel is then classified according to its hue value when its saturation and intensities are considered as sufficient. Pixels are classified as grey when they do not meet the saturation or intensity criteria. A histogram is built for partition cells from the color classification results and the histograms are normalized by cells area. Later cue fusion process will use the color distribution-based heuristics to adjust the cells scores.

### 2.3. Edge-based Analysis

Since man-made structures represent a large part of what is considered worthy of interest and must absolutely be scanned by the NFOV system and since man-made structures generally have regular shapes with sharp borders, this feature family focuses on scene region boundaries. Everything is computed from the outputs of an improved Canny's algorithm applied on the input image intensity channel. The results then obtained are gradient buffers, edge buffer and edge chains.

#### 2.3.1. Segment Analysis

Sharp building edges are likely to produce straight segments in the input image according to classic pinhole camera model. One could expect to find vertical segments and almost horizontal segments in image regions occupied by buildings. Straight road sides also project segments for which the orientation could be estimated since the perspective model is known. Those segments are first found by using the Hough line algorithm on the edge buffer from where too short chains were removed. The Hough table is then inspected according to specified orientations to fill a list with a given number of strongest line responses. A Bresenham algorithm [12] is also used to scan the edge buffer to segment each line. The lines are broken in parts where there are no overlapped pixels. A morphological dilate operation was previously applied to the edge buffer to

compensate for imprecision. A bounding box is built using the highest and the lowest horizontal segments and the leftmost and rightmost vertical segments. A better score is attributed to partition cells that are located inside or intersecting this box. Vertical and horizontal segments are also inspected to find intersections. Because intersections are very often found at the corners of buildings, they contribute to improve the containing cells score. Fig. 5 illustrates an example of segments and intersections found in a typical scene.

To determine the segments polarity, measures of image intensity transitions taken perpendicularly to oblique segments are used. Negative polarity is

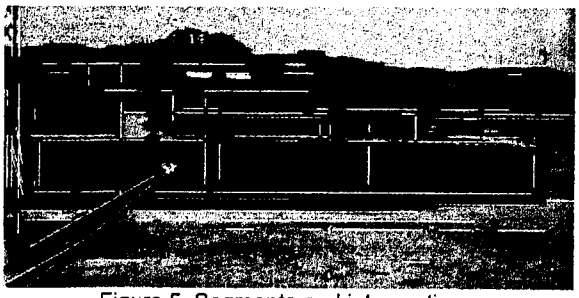

Figure 5. Segments and intersections.

assigned to segments showing light to dark transitions from left to right and positive polarity in the opposite case.

The segments orientation and position in the image as well as their polarities are used to localize potential road sides (see Fig. 6). For example, two oblique segments: segment A is

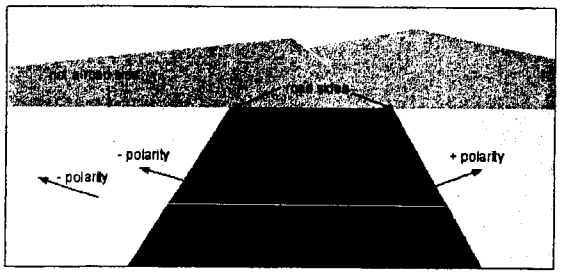


Figure 6. Using segment polarities to find road sides.

oriented within a given angle range and negatively polarized and segment B is oriented within another specified angle range and positively polarized. If A is located to the left of B, then A will be considered very likely to be a left road side. The same logic applies to segment B.

### 2.3.2. Edge Chain Analysis

Like segments, edge chains are a higher level representation than the edge buffer since they are objects with a set of properties as opposed to a simple binary bitmap. This representation not only allows filtering edges by their length as previously mentioned in segment analysis but it also enables the detection of specific shape contours. Edges chains are inspected to find three kinds of shapes: closed rectangles (and ellipses), u's and n's and mountain edges.

Closed rectangles (and ellipses) are detected by computing the chain closure, the chain elongation, the aspect ratio and the area of the chain bounding box. The first property simply tells whether the edge is closed or not. The edge elongation is defined as the ratio of the principal values of the edge's inertial matrix, which corresponds to the principal directions of the edge shape. This can be approximately defined as the ratio between the edge's minimum and maximum moment. Too small values are rejected. The aspect ratio allows to limit the detection range and the area is used to compute a closed rectangle score for the containing cells.

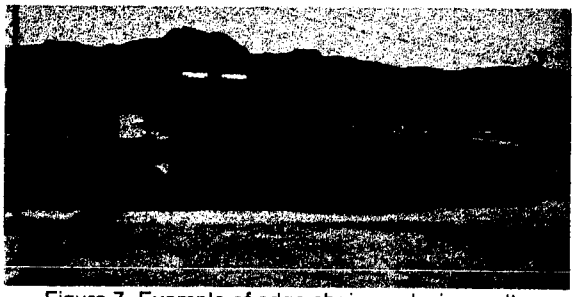

Figure 7. Example of edge chain analysis results

Noise and improved Canny algorithm automatic settings will almost always prevent edge extraction to generate all potential closed shapes. Consequently, the analysis focuses also on nearly closed shapes: the u's and n's. This time, tortuousness score, elongation, aspect ratio and chain extremities contribute to the identification. The tortuousness score is equal to the diagonal length of the chain's bounding box, divided by its length. A straight line will have a tortuousness of 1.0 while a tortuous edge will have a score decreasing towards zero. U's and n's are considered as slightly tortuous. Again too small elongation values are excluded and it is mandatory that chain extremities be localized close to bounding boxes. U's and n's contribution to cell scores is lower than closed shapes contribution.

Finally, chains are inspected to find hills and mountain edges. This category is characterized by high tortuousness values (weakly tortuous edges). An upper bound is set to exclude too straight lines and the bounding box aspect ratio and dimensional criteria are also applied. Moreover, mountain and hill edges must cross the skyline at least one time (section 2.1.1). Cells containing those edges are labeled as interesting. Fig. 7 illustrates mountain and closed edges example.

### 2.4. Motion analysis

Since the system is installed onboard a moving vehicle, knowing which parts of the scene are most likely to disappear from the field of view is useful to prioritize region scanning. In any case, regions that are qualified as interesting based on the previous cues will be privileged by the scan path planning algorithm. To estimate the region displacement within the image, a motion vector is associated to each partition cell.

The first step of the motion vector computation is based on the Shi and Tomasi [8] method. For each partition cell, the algorithm finds (if possible) one reliable feature to track. This method consists in evaluating and comparing the eigenvalues ($\lambda_1$, $\lambda_2$) of a 2X2 matrix computed from the image gradients taken from a square window centered on each pixel against a threshold. The matrix computation is presented by Eq. 1 where $g_x$ and $g_y$ are the gradients components and where the sums are computed over the window ($w$).

A window contains a reliable tracking feature when the smallest eigenvalue ($\lambda_2$) is larger than a fixed threshold value. For each cell, only the feature that produces the highest $\lambda_2$ is considered. When a cell's $\lambda_2$ is below the threshold, scene region is considered as still (or as not really important).

$$\begin{bmatrix} \sum_w g_x^2 & \sum_w g_x g_y \\ \sum_w g_x g_y & \sum_w g_y^2 \end{bmatrix} \quad (1)$$

All selected features are backtracked in the previous sequence image using a Modified Normalized Cross Correlation [14] and the current and previous feature positions are used to extrapolate their future position. When the predicted position of a cell is located outside the image, its score gets increased by a large factor to force rapid inspection. The other cells get a score proportional to their speed. Fig. 8 shows an example of the feature tracking and position prediction where the camera is panning to the left. The vector centers represent the current position, while left and right extremities represent past and future positions.

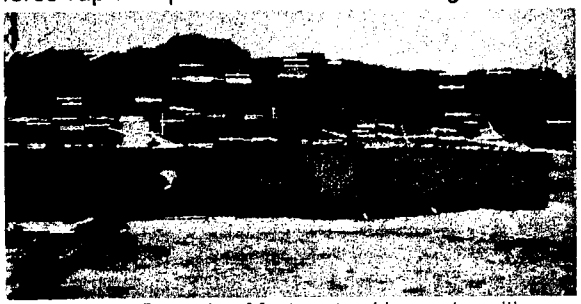

Figure 8. Example of feature tracking and position prediction.

## 3. FUSION AND DECISION

All partition cells are analyzed regarding the extracted cues. The first step is to identify the cells that belong to a road (if there is any). When segments are identified as possible road sides, they are used to compute a road potential region. A left side segment inhibits cells located to its left and activates cells located to its right, while a right side segment activates cells located to its left and inhibits cells located to its right. A road probability is also computed using the following cue combination. A cell is considered very likely to belong to a road region when its color distribution contains high proportions of grey and is weakly textured or highly granular. A large proportion of dark tone reinforces this probability. Cell image position also influences the road probability as bottom center positions are favored. A total cell road score is computed as the average between potential and associated probabilities.

The second step emphasizes on labeling cells that probably belong to the sky (if any). A cell sky probability is favorably influenced by a large proportion of blue contained in its color distribution. Low content of high frequencies in a cell texture distribution and a large part of bright tone in tone distribution also have a positive impact. Cells located above the skyline or intersecting with it will see their probability increased.

The third step relates to the texture score. Global texture statistics are computed from all partition cells. An average 50-dimension vector and a variance vector are calculated while temporal filtering is applied to both vectors. From the problem definition, it is assumed that the system searches for rare events. Consequently, cells for which texture vector components differ from filtered average vector components more than six times the associated standard deviation will get their texture score increased proportionally to the number of deviant components.

The fourth step focuses on the tone score computation. A dark cell with many bright or grey spots is considered worthy of interest. This implies that a mainly dark cell's score is proportional to its distribution bright and in-between tone contents (equally weighted). Following the same logic, the score of a mainly bright cell is proportional to its distribution dark and in-between tone contents (also equally weighted). Experiments showed that cells for which dominant tone is neither bright nor dark are less interesting. Therefore, an in-between cell tone score is penalized proportionally to its

distribution in-between tone content.

The edge score is computed at the fifth step and every cells containing closed edges will receive a score twice as important as those containing u's and n's. A high score is also attributed to cells intersecting with a mountain edge.

Step six is dedicated to computing scores related to segments. Cells intersecting with vertical segments and/or horizontal segments will receive a score proportional to segments length. The presence of segments intersections influences the score according to the following rule: a first intersection counts for 50%, a second for 25%, a third for 12.5%, and so on. Global segments bounding box also contributes to the cells score. Cells intersecting with or located inside a bounding box get bonus scores while others get penalties.

The penultimate step assigns a binary score to dark rectangles. Cells containing dark rectangles receive a 100% score.

Finally, for each cell, a level of interest is calculated from all the aforementioned scores. The scores are normalized to behave as probabilities and classified in two categories: activators and inhibitors. The former regroups segments scores, dark rectangle score, texture score, edge score, intersection score and tone score. The latter consists in the sky score and the road score. The largest activator value is used as a base score and the other activators are used as multipliers. For example: assuming the largest activating score is the texture-related score at 0.6, the other values account for 0.05. The total activation score will be equal to 0.6x1.05x1.05x1.05x1.05x1.05. The sum of the inhibiting scores is then subtracted from that result and the levels of interest are spatially and temporally filtered. Cells for which the level of interest is above a threshold value are considered worthy and will be fed to a scan path planning algorithm.

## 4. EXPERIMENTS AND DISCUSSION

The proposed algorithm has been implemented in C++ and tested on operational context sequences. In order to achieve real-time performances, a commercial image processing library has been used for most low level image processing operations. Rates of 20 fps were reached on a dual core platform. The algorithm will be field tested with the complete system onboard a patrol vehicle in a near future.

Given the nature of this work and the lack of ground truth, it is difficult to produce results other than qualitative. Nevertheless, as the algorithm's main objective is to reduce the size of a ROI to be scanned, special care was given to scene coverage. Available test sequences were used to compute mean coverage ratio over time. The algorithm succeeded to reduce scene coverage by two thirds for typical sequences. A low coverage ratio is necessary but not sufficient to insure good performances. The algorithm must also select appropriate regions. This aspect was qualitatively tested with satisfying results. A weakness was identified: the algorithm does not handle well structures like tree tops. When present in the imagery, useless scene coverage can increase significantly. The highlighted cells in Fig. 9 demonstrate the algorithm result and the area to be scanned. A human operator would certainly label a lot less cells. However, this is compensated by the fact that the algorithm is much faster.

Figure 9. Scene coverage ratio.

Another test performed was to turn off cues one by one. It was found that the system allows graceful degradation. This is explained by the method used to calculate the level of interest associated with cells, by the large number of cues involved in the process and by the fact that the cues are highly redundant.

Among the method's interesting properties: (1) it is said probabilistic because at some point in the region classification process each cell is associated to a class probability (classes being: sky, road and interesting region); (2) it is multi scale due to quad tree partitioning and multi scale texture analysis; (3) since it uses low resolution images, the method is less sensitive to acquisition noise; (4) since the method is heuristic-based it does not requires the tedious work of collecting data for a training database; (5) it is also qualified as "reactive" (as opposed to cognitive) because it does not try to formally label all the objects in the scene (like trees, buildings, ground, vehicle, etc.) and instead puts on a collection of evidences to describe a region as worthy of interest. This last aspect should be an advantage while working in a context where little scene *a priori* is available or the

environment is changing because there is no need to add new object classes when changes happen.

The proposed method could be further improved. Using a laser range finder in conjunction with the system would reduce coverage rate by excluding too far and too close regions. A common database could be used for all patrol vehicles to store positive threat detections. Data mining tools (such as classification and regression trees) could be used to learn appearance patterns. With more processing power, an optimal partitioning algorithm could be used instead of a quad tree. Finally, with a lot more processing power, local operations could be used on each pixel instead of working on partition cells.

## ACKNOWLEDGEMENTS

This work has been conducted in the framework of Defense Research and Development Canada — Valcartier Contract No. W7701-075358 under financial support from the technology demonstration program and Lyrtech Inc. The authors would like to thank cpt. PUT NAME HERE for operational context image sequences.

## REFERENCES

1. Krishnamachari, S. & Chellappa, R. (1996) Delineating Buildings by Grouping Lines with MRFs. *IEEE Trans. on Pat. Anal. Mach. Intell.*, **5**(1), 164–168.
2. Lin, C. & Nevatia, R. (1998) Building Detection and Description from a Single Intensity Image. *Computer Vision and Image Understanding*, **72**, 101–121.
3. Mayer, H. (1999) Automatic Object Extraction from Aerial Imagery a Survey Focusing on Buildings. *Computer Vision and Image Understanding*, **74**(2), 138–149.
4. Kumar, S. & Hebert, M. (2003) Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field. In Proc. IEEE Int. Conf. on CVPR, 1, 119-126.
5. Todorovic, S. & Nechyba, M.C. (2004) Detection of Artificial Structures in Natural-Scene Images Using Dynamic Trees. In Proc. 17th Int'l Conf. Pattern Recognition (ICPR), vol. 1, Cambridge, U.K., 35–39
6. Bradshaw, B., Scholkopf, B. & Platt, J. C. (2001) Kernel Methods for Extracting Local Image Semantics. Tech. Report MSRTR-2001-99, Microsoft Research.
7. Iqbal, Q. & Aggarwal, J. K. (1999) Applying Perceptual Grouping to Content-Based Image Retrieval: Building images. In Proc. IEEE Int. Conf. on CVPR, 1:42–48.
8. Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Sys., Man., Cyber.* **9**: 62–66.
9. Habib, H. A., Yousaf, M. H. & Mohibullah, M. (2004) Modified Laws Energy Descriptor for Inspection of Ceramic Tiles. In Proc NCET 2004.
10. Duda, R. O., Hart, P. E. & Stork, D. G. (2001) *Pattern Classification*, Second Edition, p. 170.
11. Viola, P. & Jones, M. (2001) Rapid Object Detection Using Boosted Cascade of Simple Features. In Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition. 1:511-518
12. Bresenham, J.E. (1965) Algorithm for Computer Control of a Digital Plotter. *IBM Systems Journal*, Vol. 4, No.1, pp. 25–30
13. Shi, J. & Tomasi, C. (1994) Good Features to Track. In Proc. IEEE Conference on CVPR.
14. Mulligan, J., Isler, V. & Daniilidis, K. (2001) Trinocular Stereo: a Real-Time Algorithm and its Evaluation. *International Journal of Computer Vision*. 47:51-61.