Video-Based Obstacle Detection for Coordinated Machines

Shinko Y. Cheng, Mike Daily, Yuri Owechko, Swarup Medasani, Zach Bonefas*

{sycheng,mdaily,yowechko,smedasani}@hrl.com, bonefaszacharyt@johndeere.com
HRL Laboratories, Malibu, CA
**John Deere Technology Center, Moline, IL*

ABSTRACT

Robotics holds great promise for improving the productivity of farm machines. The next jump in productivity for the farmer will likely involve the coordination of multiple smaller, lower power, autonomous farm machines by a single operator. One of the challenges in designing such vehicles is detecting obstacles automatically, in order to alert the operators to avoid them. We present a study of a computer vision approach for detecting obstacles in a farm environment, which to our knowledge has not been previously done before. The approach is a patch-based object detection algorithm called SwarmVision™ trained to detect humans, all-terrain vehicles and trucks. We describe the algorithm and the detection performance and results, and conclude with a discussion of our findings.

Keywords: Object Detection, Computer Vision, Obstacle Avoidance, Coordinated Machine

1 INTRODUCTION

Development of larger tractors and combines has traditionally achieved major productivity gains, but this paradigm is reaching its limitations. At the same time, the availability of farm labor has become scarcer. Many farmers who are enlarging their operations need new ways to increase the productivity of their labor resources. One potential solution is to use a number of smaller automated vehicles, each with greater flexibility but lower power. Highly automated vehicles have the potential to provide a jump in productivity for the farmer.

In order to enable automated vehicles, new and promising capabilities from the field of robotics are needed. Full autonomy of farm equipment is likely not feasible for the foreseeable future due to such variables as required safety, reliability and associated liability concerns, ultimately requiring a human-in-the-loop. With a human-in-the-loop, a central question is how to enable a single operator to monitor two or more machines. One of the most challenging aspects of controlling multiple machines and the focus of this paper is detecting obstacles. This capability would then be used to aid the operator and avoid the obstacles. This is also known as safeguarding.

Our objective in this research is to determine what obstacles in the farm environment can be automatically and reliably detect using video cameras and computer vision methods. The problem is challenging due to the harsh operating conditions, such as rain, dust, wind, operation at dusk, and widely varying crop texture patterns and hues. We present quantitative results from applying our implementation of HRL Laboratory's image-based object detection called SwarmVision™ to detecting multiple classes of objects in the farm environment under these various conditions.

Image patch-based object detection techniques have received much attention in computer vision research in recent years (Dalal & Triggs, 2005)(Luo, 2005)(Medasani & Owechko, 2007)(Owechko & Medasani, 2005)(Owechko & Medasani, 2004)(Saisan et al., 2005)(Zhang & Viola, 2007)(Viola & Jones, 2001)(Zhu et al., 2006). This approach involves taking all fixed aspect-ratio patches of an image and analyzing each patch with a two-class pattern classifier to determine whether the object of interest is contained within. The patches that generated a classification response that exceed an optimized threshold are returned as "detections." Some of the challenges with this approach relate to the generally large number of patches that needs to be processed and the computationally intense feature extraction step prior to classification in the processing. Some aspects of these challenges were overcome in Viola and Jones' seminal paper on the approach by introducing the boosted classifier cascade structure and an efficient feature extraction algorithm called the integral image that has made the approach practical for a wide range of applications. The cascade of boosted classifiers allowed the system to quickly reject obvious image patches that contains object, and focusing the computation on only patches that are not obvious.

Figure 1 Obstacle detection for the farm environment

In (Owechko & Medasani, 2005), we introduced the use of particle swarm optimization in place of the traditional exhaustive search where the classifier is applied on every

patch in the image. The result is several orders of magnitude reduction in the number of image patches that needs to be analyzed, with negligible loss in object detection performance. There are several implications of this approach: First, more computational resources can be used for the generation of the image features and classification because the number of window analyses has been vastly reduced. Second, the particle positions which are solutions to the search may include *(x,y)* position, window height *h*, *as well as* other pose parameters such as object orientation, tilt and so forth. Finally, an elegant method of incorporating prior knowledge exists where the solution space (object space) may be biased to favor regions of the image where objects are more likely to be found.

Using SwarmVision, we present object detection performance results for humans, all-terrain-vehicles (ATVs), and trucks within difficult crop-textured backgrounds under a variety of lighting conditions. For each object type, we present the classification performance in terms of receiver operating characteristic (ROC) curves, which is a plot of true-positive and false-positive rates of a classifier, while we vary the structure of classifier cascade in terms of the number of stages, configuration of features and target correct classification rates.

The structure of this paper is as follows. In Section 2, we describe the SwarmVision object detection algorithm in greater detail. In Section 4 and 5, we present and discuss the results of our experiments with data collected on a farm during harvest season. We conclude with a summary in Section 6.

Figure 2 SwarmVision™ Block Diagram.

2 IMAGE-BASED OBJECT DETECTION IN THE FARM

2.1 *System Overview*

This section gives an overview of our vision-based object detection algorithm referred to as SwarmVision™. The algorithm block diagram is illustrated in Figure 2. A new image is captured and first processed by the Local Search block, which redetects objects that the system has retained in memory from previous images in the form of tracks. There are no tracks initially, so the first image is processed by the Global Search block which deploys the object detector in a "global" mode to search the entire image for objects. Any objects that have been found are used as observations by the multi-target tracker consisting of the Kalman Filter and Global Nearest Neighbor Data Association filter. The final output of the system is the tracked object windows in the image.

The process continues from the beginning with the next image. Assuming that an object has been tracked since the last frame, the window location of the track is used to redetect the object in the new image, by initializing the particle swarm in the neighborhood of that location. The details of particle swarm optimization are described later on. If the redetection of the object was successful, the mask of the window is laid onto an *object map*. This is a kind of *pheromone map* that discourages subsequent particle swarms from converging at those regions. This map prevents both the Local and Global Search blocks from detecting the same object in the same image, again and again. The Global Search block searches the image for additional objects in the image using the object map. Then, again any objects that have been found (detected or redetected) are provided to the tracker as observations to update the tracks.

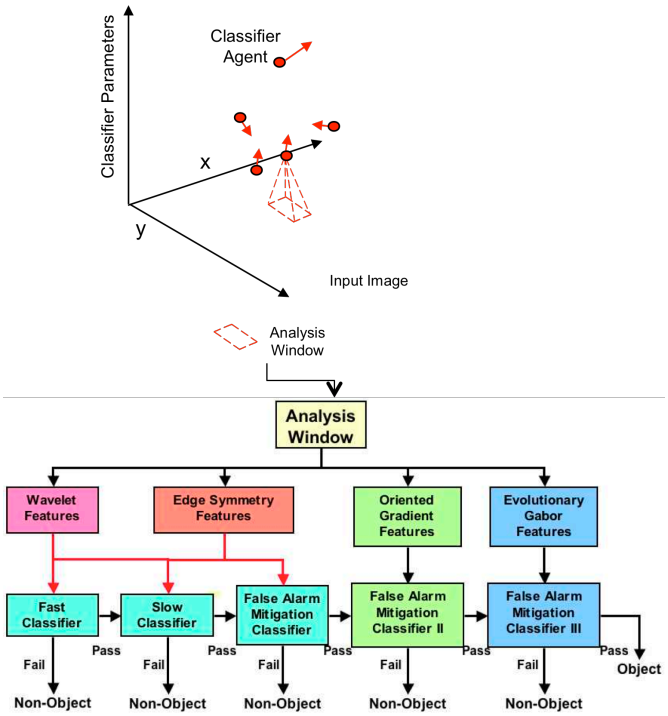

Figure 3 SwarmVision™ uses particle swarm optimization to rapidly search a scene and a classifier cascade to robustly detect objects. The above cascade structure is one of the many that can be constructed and trained using the framework.

The *height map* is an optional pheromone map that is used by the Local and Global Search blocks to confine the search space using prior knowledge about the geometry of the scene. Specifically, the height map is the same size as the image whose value at each *(x,y)* location is the expected height of the object in pixels at that location in the image. For the case of a forward looking tractor camera, one can compute the expected height in pixels that a human may appear in the image given the camera orientation, intrinsic parameters, and expected human height of 180cm.

The Local and Global Search blocks utilize of the SwarmVision™ Object Detector. Each search block consists

of several simulations of the particle swarm. One run of the particle swarm searches for one object. The Local Search block implements as many swarms as there are tracks. The Global Search block implements as many swarms as there are objects that exceed a threshold, i.e. new objects. The maximum number of objects to locate and track may be fixed.

The SwarmVision™ Object Detector consists of two primary components: the search algorithm based on particle-swarm optimization (PSO) and the classifier cascade. These two components are illustrated in Figure 3. SwarmVision™ uses PSO to rapidly search a scene to locate objects of interest. Each particle in the swarm operates as a self-contained obstacle detector, trained to return a high response if the fixed aspect-ratio image window contains an object. The collection of particles collaborates by guiding each other in locating the object of interest. Through this collaboration, the speed at which the particles converge to the "solutions" has been shown to be orders of magnitude faster than alternatives such as exhaustive search and pyramid search (Owechko & Medasani, 2005) (Owechko & Medasani, 2004) with little loss in performance.

2.2 *Particle Swarm Optimization*

PSO focuses the attention of the search for objects in areas of the image that are more likely to contain an object, eventually converging on the object—using the analogy of swarms—once its "scent" was found. PSO is part of the family of genetic optimization techniques that searches for the optimal solution to the objective function by using swarm-like dynamics to find it (Kennedy & Eberhart, 1995). In object detection, the classifier response map for a given image window is our objective function, and swarm searches for the optimal window location *(x,y)* and height *h*. Let the position of particle *i* at time *t* be $\mathbf{x}_{i,t}$ which is in the space of all possible image window positions $\mathbf{X}$, that is $\mathbf{x}_{i,t} \in \mathbf{X} \subset \Re^3$. Without any prior information, the object may be anywhere in the image. In this case, a fixed number of *N* particles are initially uniformly distributed over the space of $\mathbf{X}$, that is $\mathbf{x}_{i,0} \sim U(\mathbf{X})\ \forall i$. The cascade classifier $h : \mathbf{X} \to \Re$ analyzes each particle and produces a likelihood of object existence at the particle's position. A second function, in the language of particle swarm optimization, is the pheromone map $f : \mathbf{X} \to \Re$. Via pheromone maps, the particles can be encouraged to either seek or avoid solutions in various regions of the solution space. These maps are analogous to the prior probability in Bayesian estimation. The highest likelihoods that each particle has encountered since *t=0* is referred to as *local-best*,

$$\mathbf{l}_{i,t} = \arg\max_{t'<t} h(\mathbf{x}_{i,t}) + f(\mathbf{x}_{i,t}) \quad \forall i \tag{1}$$

The highest likelihood among all particles over all-time is referred to as the *global-best*,

$$\mathbf{g}_t = \arg\max_{i,t'<t} h(\mathbf{x}_{i,t}) + f(\mathbf{x}_{i,t}) \tag{2}$$

Each particle is then moved to the next position by recursively evaluating (1), (2) and (3) until the $\mathbf{g}$ converges to a solution or a fixed number of iterations have been met.

$$\begin{aligned} \mathbf{v}_{i,t+1} &= \omega\, \mathbf{v}_{i,t} + c_1 \mathbf{r}_1 \otimes (\mathbf{l}_{i,t} + \mathbf{x}_{i,t}) \\ &\quad + c_2 \mathbf{r}_2 \otimes (\mathbf{g}_t - \mathbf{x}_{i,t}) \quad \forall i \\ \mathbf{x}_{i,t+1} &= \mathbf{x}_{i,t} + \mathbf{v}_{i,t} \end{aligned} \tag{3}$$

The symbol $\otimes$ indicates element-wise multiplication, vectors $\mathbf{r}_1$, $\mathbf{r}_2$ are randomly distributed vectors, and constants ω, $\mathbf{c}_1$, $\mathbf{c}_2$, control the influence of the local and global best estimates in the dynamics of the swarm. In our experiments, $\mathbf{r}_1, \mathbf{r}_2 \sim U[-1,1]$ were uniformly distributed, $\omega = c_1 = c_2 = 0.7$. If the likelihood at the converged $\mathbf{g}$ exceeds a specified threshold, the swarm is considered to have detected the object.

2.3 *Classifier Cascade*

The second component of our algorithm is the classifier cascade. The cascade consists of several stages of feature generators followed by a two-class classifier. The set of features consists of

1. Evolutionary Haar Wavelet Features (EvHaar).
2. Evolutionary Gabor Features (EvGabor).
3. Edge Symmetry Features (ESF).
4. Histograms of Oriented Gradients (HOG)

Evolutionary wavelets are a grid of Haar or Gabor wavelets within an image window whose position, orientation, size and type are optimized (also using PSO) such that the filter responses (feature values) from the wavelets yield the greatest saliency, which is a measure of usefulness of the feature for distinguishing between target and non-target objects. Edge-symmetry features are a set of symmetric edge filters designed to yield a high response for objects that have symmetric horizontal or vertical lines, such as pedestrians. Histogram-of-oriented gradients consist of normalized histograms of image gradient orientations within all 8x8 pixel cells tiled over the image window, first proposed by Dalal and Triggs (Dalal & Triggs, 2005) for pedestrian detection. Each feature generator is followed by a two-class gentleBoost classifier (J. Friedman, 2000), classifying patches as true, that is containing an target object, or false, that is containing a non-target object.

The cascade structure ensures only patches that pass all stages are classified as a target-object. If an input patch is rejected at any stage, that patch is not analyzed further eliminating a large amount of computations. The trade-off is in having to throw out possibly true patches that are rejected as false in earlier stages and not allowing later stages to recover it. However, the gains in speed significantly out-weigh the loss in accuracy, and loosening thresholds in the

earlier stages can mitigate the chance of mistakenly throwing out true patches.

Figure 4 Sample images of humans, ATVs, and trucks manually collected from video.

3 DATA SETS AND METHODOLOGY

We constructed cascade classifiers for humans, ATVs, and trucks. We collected a variety of video data during the fall harvest using a Point Grey Flea2 camera (model FL2-08S2C-C) at a resolution of 1024x768 pixels. Automatic exposure adjustment fitted with a 2.5mm focal length lens. Example views are shown in Figure 1. We extracted samples of all the objects with 2:1 or 1:2 aspect-ratios, rescaled to 128x64 or 64x128 pixels. A few of these samples are shown in Figure 4. In all, 2901, 3145, and 4650 samples of Humans, ATVs and Trucks were annotated to form the training, validation and test sets. We annotated each object such that it appears upright in the window with a minimum spacing around the object of 20% the length along that axis. A portion of the positive samples are reserved for the test set. Patches of the same aspect-ratio were selected at random around the positive samples, to form the negative dataset. 10,000 negative samples were reserved for the test set. 5000 negative samples were extracted for the training set and 5000 more for the validation set (cascade parameter tuning) for each stage.

Our implementation allows for a heterogeneous set of feature and classifier types to be used in the cascade according to specific design requirements of the system; different features may be concatenated to form longer more descriptive feature representations of the image window. In our experiments, we used 2 cascade structures: The first structure consists of 4 different features: EvHaar (Evolutionary Haar), ESF (Edge Symmetry Features), EvGabor (Evolutionary Gabor), and HOG (Histogram of Oriented Gradients). We previously showed the effectiveness of EvHaar and ESF features (Owechko & Medasani, 2007). We present results in this paper on the performance gained with the EvGabor and HOG feature types. The classifier is a boosted ensemble of decision-tree classifiers learned using gentleBoost. The ordering of features in the first structure is motivated by the speed with which the features can be calculated. EvHaar is the fastest to compute feature followed by ESF, EvGabor and HOG in this order.

The second structure consists of a single type of feature and classifier, the HOG and gentleBoost classifier. This second structure is motivated by the work of Dalal and Triggs, who developed the original HOG feature for representing image patches in pedestrian detection, and yielded good results. Zhu et al. refined the HOG feature such that the computation can be sped up through the use of the integral-histogram (as opposed to the Integral image) (Zhu et al., 2006) of the image at approximately the same performance.

For each object, the features and classifiers used are summarized alongside the classifier receiver-operating-characteristics (ROC) curves in Figure 6.

To quantify cascade performance, we plot the ROC curves, constructed by measuring the true-positive (TP) and false-positive (FP) per window rates for a given threshold to the classification response over the entire range of responses. TP is the proportion of positive samples classified correctly, and FP is the proportion of negative samples classified incorrectly. First, the curve of stages 0 to 1 are measured, then 0 to 2, and 0 to 3, and so on. A sample is assigned a cascade response of $r'=100n+r$, where r is the classifier response of the n^{th} stage executed. Since each classifier

Figure 5 Examples of detected humans. (Best viewed in color)

response ranges between approximately -50 to 50, although the support of the response is the entire real line, the factor of 100 helps to ensure clear ordering between samples that pass the n^{th} stage and the m^{th} stage, when $n \neq m$.

4 EXPERIMENTAL RESULTS

Figure 5 shows the results of the object detection algorithm. In nearly all cases, each additional stage of the cascade improved the classification performance for the three objects. Between two types of Human cascade structures, the HOG only cascade performed marginally better at high FP rates and marginally worse at low FP rates. In the case of the ATV and Truck cascades, the HOG-only classifier performed better uniformly across all operating points. The performance gained with HOG-only however comes at the cost of slower execution speed because HOG is relatively computationally intensive and it is computed for all samples, whereas the classifier with the EvHaarESFEvGaborHOG cascade structure, HOG is computed only for samples that survive the first several stages, which is less computationally intensive.

We set the threshold on the classifier responses to decide whether a sample proceeds to the next stage by setting the thresholds such that a certain TP rate of the validation-set is achieved. For the human cascade, the target stage TP rate is 99.9%. For the ATV cascade it is 99.5%, and the truck cascade, 99.9%. This ensures that given N stages, the lower-bound of the cascade TP rate is $TP_{cas} = TP^{N}_{target}$. Typically, the associated FP rate at each stage is sufficiently low (0.3-0.8) so that the cascade FP rate decreases much more quickly. The final cascade performance varies over the values of the target stage TP rate, with an optimal value estimated by trying several values. The target stage TP rate was swept over values 98%, 99.5% and 99.9%, and the best performing cascade is presented. An example operating point on the multi-stage ROC curve for humans is TP=90% for FP=1%.

5 DISCUSSIONS

A number of possible improvements may be made to the classifier cascade. The choice of classifier in this experiment disallowed the ability to efficiently train for an arbitrary number of weak-classifiers which comprise the classifier for each stage. The ability to tune this number allows for the training of a classifier with an arbitrarily low FP rate for a given TP rate on the training set. Currently, the number of weak-classifiers is fixed at the beginning to 300 weak classifiers per stage. The current method potentially has drawbacks in either accuracy or speed: too few weak

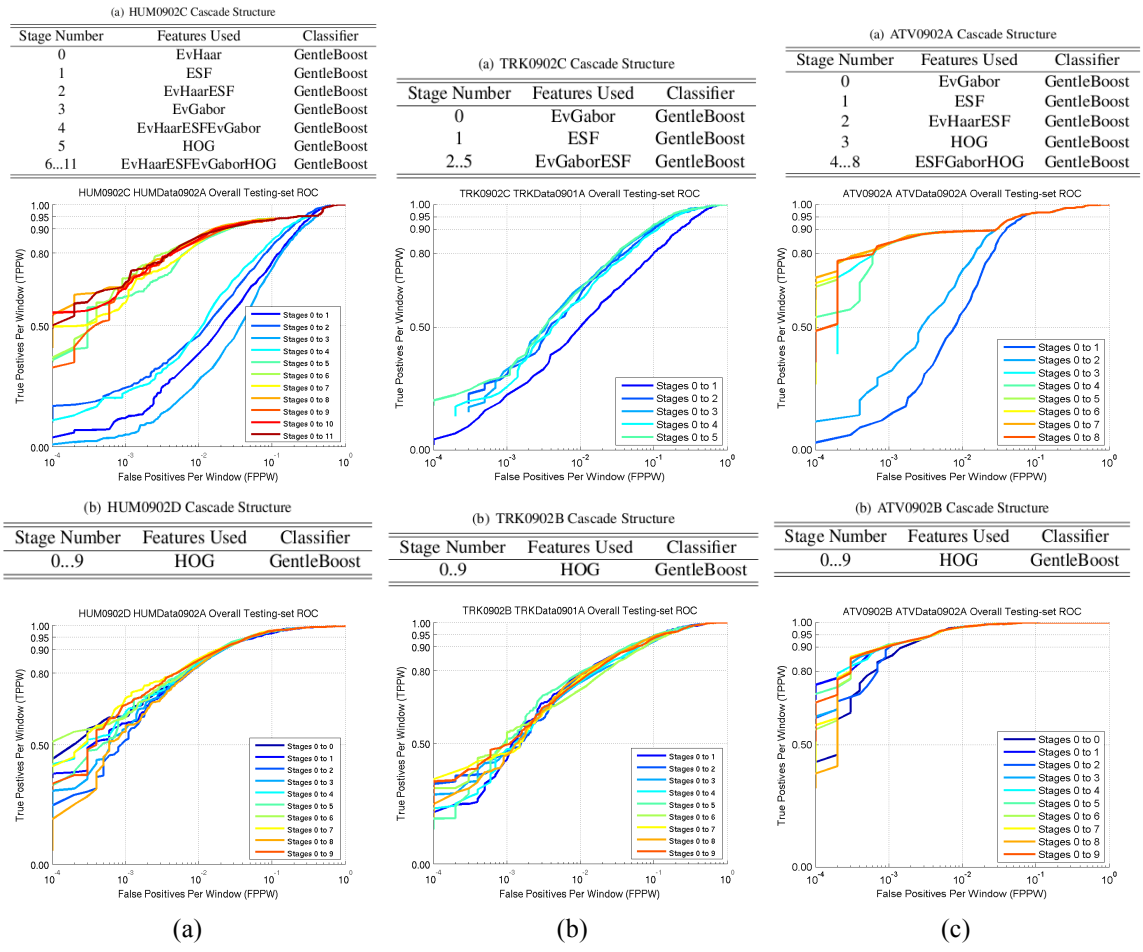

Figure 6 (a) Human, (b) ATV, and (c) Truck Cascade Structures and Classifier Performances.

classifiers would not fully exploit the discriminative power and too many weak classifiers runs the risk of needlessly computing features that could have been avoided. In fact, to optimally train a classifier, one needs to balance the number of weak-classifiers per stage by tuning the target-MD/FP rate. Multiple-instance pruning (Zhang & Viola, 2007) or optimization design of cascaded classifiers (Luo, 2005) may represent part of the solution to this problem. The next step is therefore to investigate a cascade training method that optimizes the number of weak-classifiers per stage. These modifications may address the failure conditions illustrated in Figure 7.

(a) (b)

Figure 7 Failures in detection. (a) Missed human target ducked behind crops. (b) Small section of crop mistakenly detected as humans.

6 CONCLUSIONS

Robotics holds great promise for improving the productivity of farm machines. The next jump in productivity for the farmer will likely involve the coordination of multiple smaller, lower power, autonomous farm machines by a single operator. One of the challenges in designing such vehicles is automatically detecting obstacles in order to alert the operators to avoid them. We presented a study of computer vision-based obstacle detection in the farm environment, which to our knowledge has not been studied before. We presented the classification performance of an implementation of patch-based object detection algorithm called SwarmVision™ that was trained to detect humans, all-terrain vehicles and trucks, commonly found obstacles in the farm environment.

We are looking into further improvements that combine texture segmentation with patch-based object detection given that crops have distinct repeating patterns, by using the texture segments or the texture representation to augment the features of patches. We are looking into a method to tune the correct classification rates in the classifiers as well.

7 REFERENCES

Dalal, N. & Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*., 2005.

J. Friedman, T.H.a.R.T., 2000. Additive logistic regression: a statistical view of boosting. In *The Annals of Statistics*., 2000.

Kennedy, J. & Eberhart, R., 1995. Particle Swarm optimization. In *IEEE Int'l Conference on Neural Networks*., 1995.

Luo, H., 2005. Optimization Design of Cascaded Classifiers. In *IEEE Int'l Conference on Computer Vision and Pattern Recognition*., 2005.

Malik, J., Belongie, S., Leung, T. & Shi1, J., 2001. Contour and Texture Analysis for Image Segmentation. *International Journal of Computer Vision*, 43(1), pp.1573-405.

Medasani, S. & Owechko, Y., 2007. Behavior Recognition using Cognitive Swarms and Fuzzy Graphs. In *SPIE Intelligent Computing: Theory and Applications V*., 2007.

Morimoto, E., Suguri, M. & Umeda, M., 2002. Obstacle Avoidance System for Autonomous Transportation Vehicle based on Image Processing. *Commission Internationale du Genie Rural*, IV.

Owechko, Y. & Medasani, S., 2004. Classifier Swarms for Human Detection in Infrared Imagery. In *IEEE Computer Society conference on Computer Vision and Pattern Recognition*., 2004.

Owechko, Y. & Medasani, S., 2005. A Swarm-Based Volition/Attention Framework for Object Recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*., 2005.

Owechko, Y. & Medasani, S., 2007. Swarm optimization methods for cognitive image analysis. In *SPIE Unconventional Imaging III*., 2007.

Porikli, F., 2005. *Integral Histogram: A Fast Way to Extract histograms in Cartesian Spaces*. TR2005-057. Mitsubishi Electric Research Laboratories.

Saisan, P., Medasani, S. & Owechko, Y., 2005. Multi-view Classifier Swarms for Pedestrian Detection and Tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. San Diego, 2005.

Viola, P. & Jones, M., 2001. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *IEEE Computer Society on Computer Vision and Pattern Recognition*., 2001.

Zhang, C. & Viola, P., 2007. Multiple-Instance Pruning for Learning Efficient Cascade Detectors. In *Neural information processing Systems*., 2007.

Zhu, Q., Avidan, S., Yeh, M.-C. & Cheng, K.-T., 2006. *Fast Human Detection Using a Cascade of Histograms of Oriented Gradients*. TR2006-068. Mitsubishi Electric Research Laboratories.