

# APHOG: A Framework for Fast Object Detection Using Histograms of Oriented Gradients

Chris Cowdery-Corvan  
cjc5976@rit.edu

Liangyi Fan  
lxf6441@rit.edu

Thomas Knack  
tak5633@rit.edu

**Abstract**—In this paper we show how it is possible to improve the efficiency of existing holistic forms of object detection by refining detection areas to smaller subsets. Although this method can be applied to any form of object detection, this paper will specifically focus on the topic of pedestrian detection in low-resolution non-stationary video footage.

## I. INTRODUCTION

Pedestrian detection has always been a very difficult task, primarily due to the varying nature of pedestrians. Pedestrians can be of a multitude of different sizes, shapes, colors and postures thus leading to high intra-class variability. As pedestrians are non-rigid bodies and can vary in appearance drastically, the need for a detection algorithm surfaced that is color, illumination and scale invariant. Subsequently, two paradigms emerged for the detection of pedestrians: part-based and holistic.

Part-based algorithms analyze the underlying geometry of possible pedestrians in an image using techniques such as Haar wavelets. By detecting local features of pedestrians (e.g. torso, arms, face, etc.) and the positions of the local features relative to each other, it is possible to determine whether or not the geometry of the detected portions could be a pedestrian or not.

Holistic algorithms analyze the entire frame of a video and attempt to match features on multiple scales in technique commonly referred to as pyramiding. The image is inspected repeatedly at multiple sizes to determine if the geometry of a figure exists in such a way that it resembles a person.

Although both approaches yield useful and impressive results, both have their shortcomings. Part-based algorithms are extremely computationally intensive, and correct local feature extraction of an image is an exceedingly difficult task. Holistic algorithms require vast amounts of image processing & cross-correlation to detect possible person-like figures, and they are susceptible to false positive detections due to background clutter.

As a result, APHOG aims to capture both a part-based and holistic approach to detecting pedestrians. Coarse-grained part-based detectors provide scales and coordinates of possible pedestrians, and fine-grained holistic Histogram of Oriented Gradients (HOG) detectors confirm the existence of pedestrians using the regions of interest identified by the coarse-grained part-based solvers.

## II. ARCHITECTURE

Although the APHOG detector, Statistical Model and Classifier were implemented using C++ and OpenCV 2.2, the detector can receive region of interest candidates (ROICs) from coarse-grained part-based detectors implemented in mexed MATLAB or C++ code. The primary advantage of this is that developers using the framework are not constrained to a language or technology stack, provided the algorithm can return ROICs in the following format:

$$ROIC_n = (x_1, y_1, x_2, y_2, confidence)$$

Where  $x_1, y_1, x_2, y_2$  mark the top left and bottom right points of a rectangle respectively, and confidence is a double in the range  $0 \dots 1$  quantifying the belief that the region contains an object of interest (OOI) where 0 denotes no confidence and 1 denotes full confidence. The strategy software design pattern was used to encapsulate each algorithm and make them interchangeable.

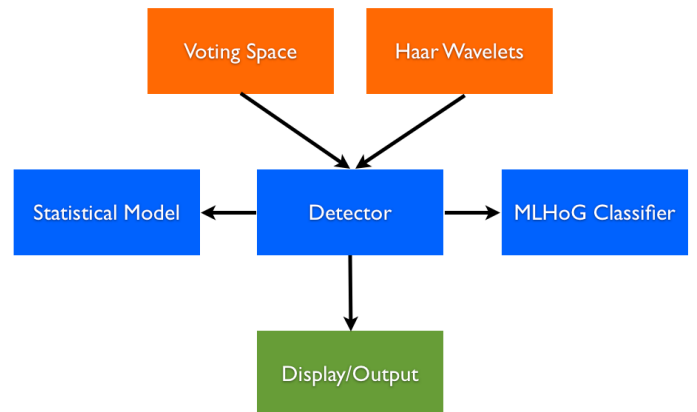


Fig. 1. APHOG Architecture

However, if all of these computations were performed sequentially no performance gains would be present, on the contrary, a significant and detrimental effect on performance would be encountered. In order to boost performance of existing algorithms the calls to the ROIC detector are massively parallelized and computed using cores available on the Graphics Processing Unit (GPU). ROIC detector computations are abstracted away into smaller computable kernels and executed using OpenCL calls. Moreover, each ROIC detector can flag

specific areas of code to be further parallelized using comment blocks as follows:

```

1  for (i = 0; i < count; i++) {
2  /*APHOG: beginParallelizable*/
3      results[i] = do_some_work(data, i);
4  /*APHOG: endParallelizable*/
5  }

```

Listing 1. C++ Example of APHOG Parallelizable Code

Calls are executed on each GPU core available, and whenever flagged the detector will create a separate kernel stack for each section of parallelizable code and join the results when available. Significant care had to be taken to mitigate race conditions, ensure high utility despite soft deadlines, and load-balance the computation to the next available core.

In the next coming sections we will focus on how the entire algorithm works while referencing its architecture.

### III. APHOG

As previously mentioned, both part-based and holistic approaches yield useful and impressive results, both have their shortcomings. Part-based algorithms are extremely computationally intensive, and correct local feature extraction of an image is an exceedingly difficult task. Holistic algorithms require vast amounts of image processing & cross-correlation to detect possible person-like figures, and they are susceptible to false positive detections due to background clutter.

The key advantage of APHOG is that it is able to detect pedestrians in both a holistic and part-based manner. Not only did this lead to lower false positives per window (FPPW), it also increased performance due to its massively parallel nature.

To begin, the coarse-grained part-based detectors return to the detector regions that have a possibility of having pedestrian in them. In addition to the area, the respective confidences are returned to the fine-grained HOG detector to confirm the existence of pedestrians in the specified area. This eliminates analyzing every possible permutation of the HOG pyramid by narrowing it to a smaller subset containing regions of interest. In this paper, Voting Space and Haar Wavelets are used as our coarse-grained detectors. More information about how these optimized detectors operate is delineated in the following section.

The advantage of this approach is that once an understanding of how a pedestrian appears and moves, it is no longer necessary to run an entire detector on a candidate match. A model is created and stored that tracks a particular person for as long as they are determined to exist in the frame, and that model is tagged with that person. This is a great boon for the detection process as it drastically reduces the number of times the entire core detector needs be run across the entire frame because as the core detector is notably large.

#### A. Coarse-Grained Part-Based Detector

To visualize the results, below is a hypothetical scenario for a given initial frame.

In Figure 2, the red boxes represent areas that contain possible pedestrian-like features using solely Haar Wavelets. The Haar Wavelet detector finds multiple patches where features associated with pedestrians are detected, and these are sent along with confidences to the detector.

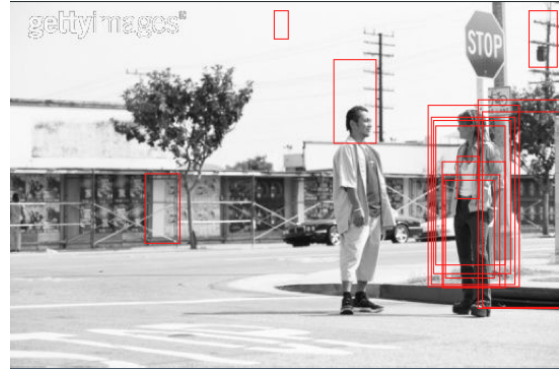


Fig. 2. An initial frame with pedestrian-like features from Haar Wavelets.

Next, in Figure 3 bounding boxes are created. Bounding boxes are designed to encompass all the coarse-grained part-based detections, and are scaled relative to the confidence of the result. A lower confidence creates a bounding box with more padding, provided the confidence meets a pre-defined threshold. These are the areas that the fine-grained detection will focus on.

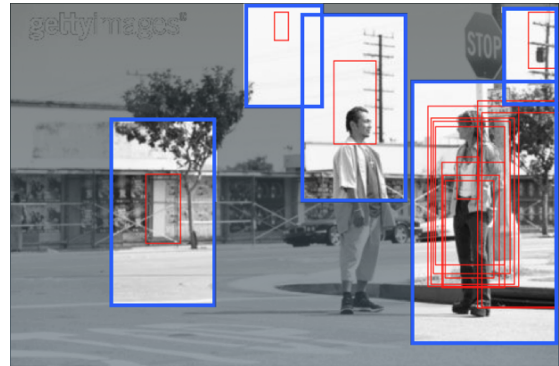


Fig. 3. Bounding boxes around grouped detections scaled with respective confidences.

Now the pyramids needed to detect against are cut down drastically, and an approximate understanding of the pedestrians has been attained. Not all results are perfect (and this was our most problematic frame throughout all of our testing), which further underscores the importance of multiple coarse-grained part-based detectors to increase possible areas and confidences.

#### B. Statistical Model & MLHOG Classifier

After the coarse-grained part-based detectors suggest ROICs and confidences to the detector, the detector will accept the areas and immediately pass them to the statistical model in another thread for processing. The statistical model is

used to calculate the new positions of pedestrians based on previous movement, namely by calculating the pedestrian's current acceleration and velocity. This is used to further refine where the pedestrians will be detected in both coarse-grained and fine-grained detectors for the next few consecutive future frames. The primary advantage of which is that it is much cheaper computationally to guess where a pedestrian will be in the future than it is to run coarse-grained part-based detections across an entire frame. The astute reader will now see that the detector uses two forms of approximation to refine the areas it analyzes. The statistical model is used to refine where the coarse-grained part-based detector should analyze, and furthermore, the coarse-grained part-based detector reduces the number of areas that the fine-grained HOG detector should analyze.

Another extremely useful and important facet to APHOG is the machine-learning SVM classifier (hereby referred to as MLHOG) used to programmatically add pedestrians to the training dataset. If a pedestrian (markerlessly tagged) follows the statistical model, has a confidence above a pre-set threshold (90% was used in our tests), and is successfully detected in at least 40 frames out of a possible sequence of 60 consecutive frames, the gradients for the pedestrian are extracted and dynamically added to the database. Once these gradients are extracted, the pedestrian is detected using solely the gradients that match the pedestrian, thereby drastically reducing the number of computations needed to detect the pedestrian. This also led to a lower FPPW count as the gradient model was already well-defined for the pedestrian it was being analyzed against.

It is important to note though that if a false positive result does match the aforementioned criteria it will be added programmatically to the database. Given a specific scenario, it is possible to have the detector without any safeguards 'collapse', or in other words, exponentially detect more false positives due to previously added false positives to the database. In order to protect against detector collapse, sanity checks are performed every ten seconds. More specifically, if the programmatically added gradients do not meet a minimum threshold (in our tests, approximately 60%) when tested using fast normalized cross correlation, the gradient is removed from the database.

#### IV. VOTING SPACE

In this project, as our framework shows, we use voting space to detect some potential person's area, and pass the matrix as the coordinates of person's potential area and its confidence value. The detail of implementation in the coming section.

The algorithm steps are shown below:

- 1) Use Harris or SIFT to find feature points in the shape of person. (Using a mask image in the shape of a person).
- 2) Find the center of the detected person.
- 3) Connect the feature points to the center point of person. Obtain several sets of patches that contain the density value of patches and the voting vector with that patch.
- 4) Using clustering algorithm to cluster the set of patches, average the density value of same cluster patch and keep

voting vector to the average patch.

- 5) Use the average patches and their corresponding voting vectors to test the unknown image to create the voting space of that image.
- 6) Find the bounding box for each person.

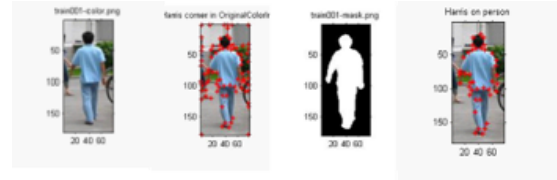


Fig. 4. Detecting the corresponding feature points on the person.

Figure 4 shows the process of detecting feature points on person using Harris corner detection methods.

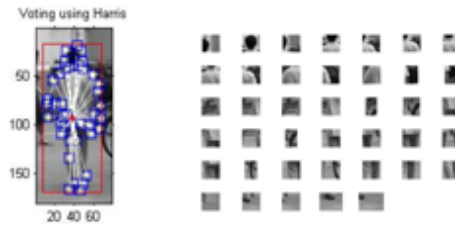


Fig. 5. Voting vectors and set of patches.

If the mask of a person is known, locating the center of a person is the next most important task. Thus, we can get the voting vector of each patch on a person in a process demonstrated in Figure 5. For each patch that was detected on a person, it is possible to get the voting vector, where the voting vector points to the center of where a pedestrian is believed to be given the position of the patch. The rightmost frame of Figure 5 demonstrates the set of patches that can be detected on one training image.

It is possible to use the aforementioned method to do the training on several image as desired. For this paper, 20 images of pedestrians with masks were selected for use. In a hypothetical execution, 741 patches on person are obtained and then clustered using cross correlation. Figure 7 shows some elements from the cluster group image. It is important to note that the clustering threshold is dependent on the images associated with the voting space detection. For our dataset, we selected 90% [ $i.f(max(g(:)) \geq 0.9)$ ] to be the cross correlation threshold to cluster the patches.

After the training patches and their corresponding voting vectors are collected, it is possible to use these patches to create the voting space in unknown image. Figure 9 shows some of the voting space result.

After the voting space is obtained, the potential area (ROIC) of the person is calculated. The process entails calculating the voting number of the space, and then compute the maximum points of voting. After this, find the corresponding patches

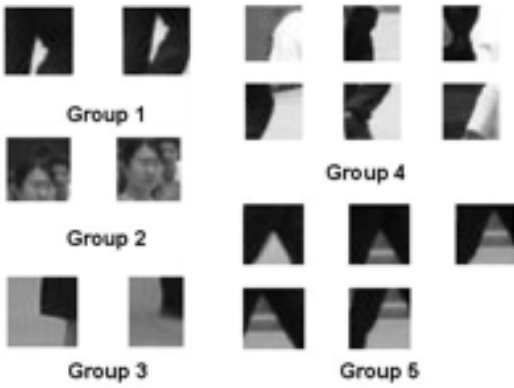


Fig. 6. A cluster group image.

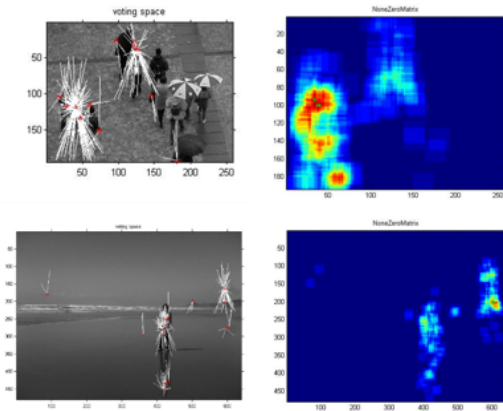


Fig. 7. Creating a Voting Space for an unknown image.

index and find the maximum and minimum index of the vector coordinates. Lastly, return (or plot) the area. The aforementioned steps are for one iteration, and upon the next iteration, the voting vectors that were not calculated before are removed. Next, the voting space is calculated again, and the process continues. Figure 10 demonstrates the detection area process.

After each iteration, we can get several potential areas containing people. Figure 11 show some of the results of the voting space detection.

#### A. Limitations

This method has two primary limitations. First, since we have little training examples which have mask images associated with them for training, our detection process can't match against as many possible patches as we would like, therefore leading to a loss in accuracy (See Figure 11).

Second, processing each frame is very computationally intensive task - so much that in many cases it became more computationally intensive to run a voting space calculation for a frame than it was to execute a full pyramid for a given frame.

Although the algorithm leads to decent results, we recommended not to use it in real-time processing as it leads

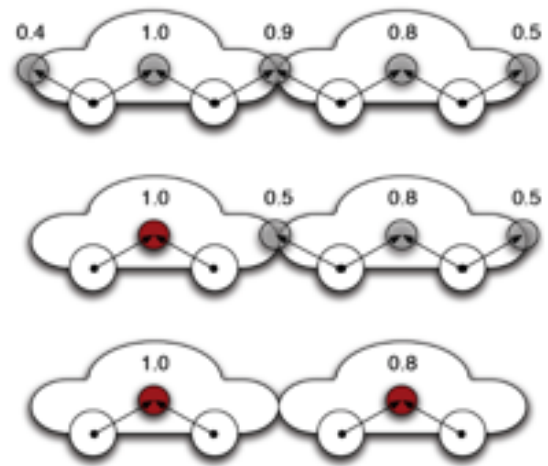


Fig. 8. A hypothetical voting space for a car.

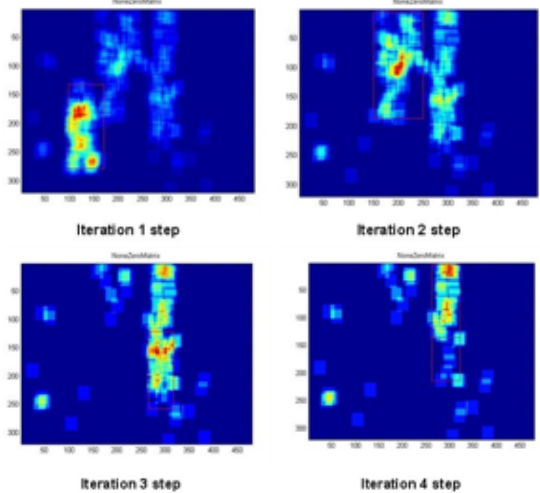


Fig. 9. The plot area process.

to a considerable number of dropped frames on modern-day hardware - even when parallelized.

As a result, there is much room for possible improvements in this detector. Potential improvements to the clustering algorithm, finding more training dataset with mask images, and discovering faster (or fewer) ways to execute the cross correlation for the patches would be an ideal area for future research in this detector.

#### V. HAAR WAVELETS

As mentioned previously, Haar wavelets were used in this research as a preliminary matching scheme. Matches defined using the templates derived by Haar wavelets were presented to the pyramid HOG algorithm for further classification. This section aims to describe the Haar wavelet process in more detail in order to develop a deeper understanding of its concepts and use within this research.

Previous research has been done concerning ratio templates. Essentially, this work sought to detect a human face through the construction and use of defined ratio templates.

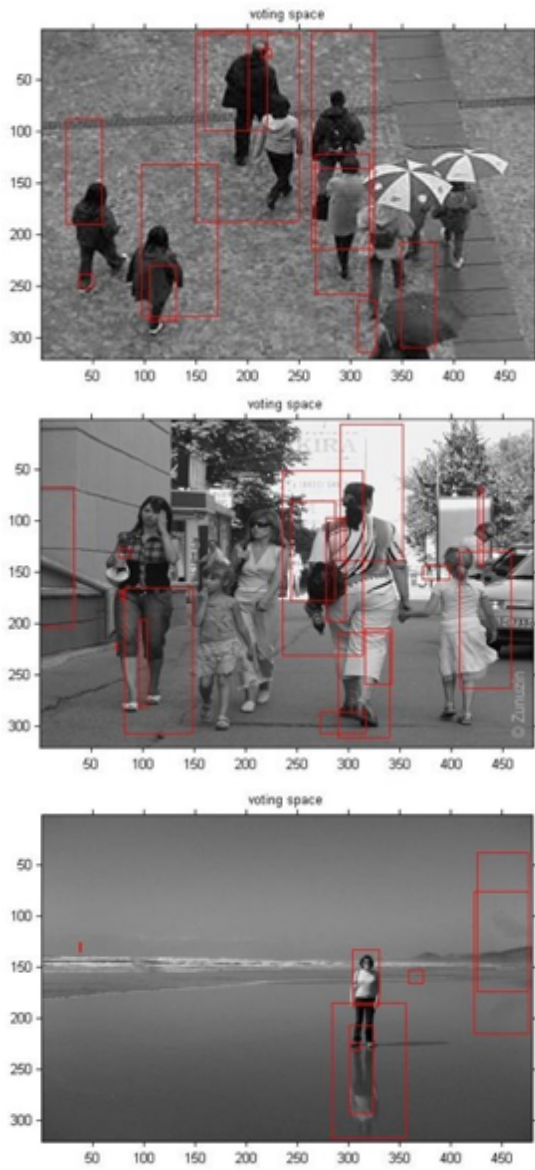


Fig. 10. The voting space detection result.

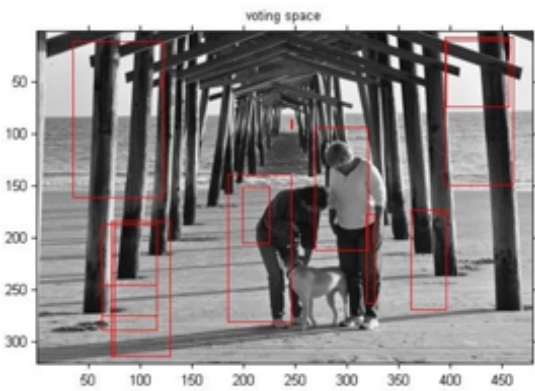


Fig. 11. An example undesirable detection result for the voting space ROIC detector.

Ratio templates attempt to encode the general structure and brightness distribution associated with a typical human face. The incentive for this work was as follows: though regions of a human faces absolute intensity values change under various illumination conditions their absolute intensity values relative to each other will always remain the same. The set of these relationships collectively define the ratio template. Logically, this sort of association can be extended to any object at various scales which makes it a perfect candidate for pedestrian tracking.

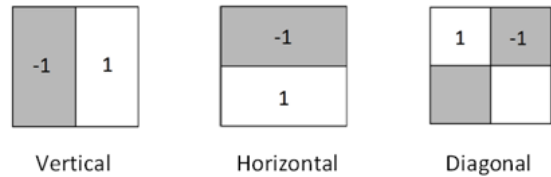


Fig. 12. 2-Dimensional vertical, horizontal and diagonal Haar wavelet basis.

The templates used to define a pedestrian were derived using Haar wavelets. Haar wavelets form an orthonormal basis function. To get a better understanding of Haar wavelets the 2-dimensional Haar wavelet basis is demonstrated in Figure 12. The elements that compose the basis describe changes in their respective directions using simple coefficients. As it will be shown later, the diagonal Haar wavelet can be generated by applying both the vertical and horizontal wavelets in conjunction with each other.

$$W = \begin{bmatrix} H \\ G \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & \text{---} & \text{---} & \text{---} \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & 0 & 0 & 1/2 & 1/2 \\ \text{---} & \text{---} & \text{---} & \text{---} & -1/2 & 1/2 & 0 & 0 \\ 0 & 0 & -1/2 & 1/2 & 0 & 0 & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & 0 & 0 & -1/2 & 1/2 \end{bmatrix}$$

Fig. 13. Haar transform matrix.

To learn the pedestrian templates the Haar wavelet basis functions were applied to each of the training data set images. These decomposition images were averaged together to define the final Haar wavelet template at various levels. To determine an images decomposition the Haar transform was used. The Haar transform is a sampling process which uses the basis functions to exam the target image at finer and finer resolutions. The Haar transform matrix is defined in Figure 13. This matrix was used within the following equation in order to determine the respective image decomposition:

$$T = W_m * I * W_n^t$$

Equation 1

In Equation 1,  $I$  is an  $m \times n$  image,  $W_m$  is an  $m \times m$  Haar wavelet transform matrix,  $W_n$  is an  $n \times n$  Haar wavelet

transform matrix and  $T$  is the Haar wavelet transform of the original image. By applying  $W_m$ , the vertical Haar matrix, and  $W_n$ , horizontal Haar Matrix, in series four 4 quadrants result in  $T$ . Starting in the top left corner and moving clockwise they are the smoothed target image, the horizontal Haar transform, the diagonal Haar transform and the vertical Haar transform. An example of this decomposition is shown in Figure 14. Multiple iterations of the decomposition procedure can be performed by decomposing the smoothed region of the decomposition produced in the previous iteration. This is also demonstrated in Figure 14 and helps to develop multiple scales if needed.

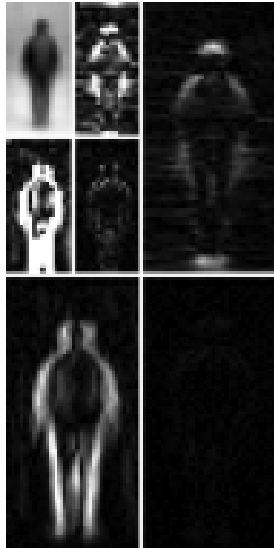


Fig. 14. Decomposition of the learned standing pedestrian pose at 2 levels.



Fig. 15. Annotation of a pedestrian in the training data set.

To learn the pedestrian templates a large training data set comprised of pedestrians in various poses was used. The images in the training data set were annotations of pedestrians

with dimensions of 128x64. That is, the images contained only the pedestrian with a significant portion of the background cropped out of the image. An example image, which was used in the training data set, is displayed in Figure 15. The learning process used a Haar transform decomposition with 2 levels. Since pedestrians exhibit various poses it made intuitive sense to define various templates for each of these poses. Using different template poses helps to increase the number of accurate matches. Four templates were defined for the purposes of this research and are shown in Figure 16. The standing and mid-stride poses are self-explanatory. The briefcase poses are meant to define a pedestrian with an object at his or her side, such as a briefcase or grocery bag.



Fig. 16. The templates learned and used within this research. From left to right the poses are classified as follows: standing, mid-stride, briefcase right and briefcase left.

Using templates to find matches within an image had various problems. First, it was very difficult to develop a generic pedestrian template regardless of any pose. A great deal of variation exists, whether color or figure or pose, between individual pedestrians. Secondly, a great deal of false positives appear in the matching results. Specifically, when templates at small scales relative to the image size are used false positives are more prevalent. This is due to the fact that, at smaller resolutions, the template itself begins to breakdown and the structure of the pedestrian becomes distorted. As such, more regions of the image appear as if they match the template due to the templates limited matching capabilities at this scale. Lastly, it was difficult to define an adaptive threshold. The strength of a match has different meanings at different scales and illuminations. Therefore, it was important to define an approach which sought to attempt to capture this variability in matching confidence across the images without sacrificing the matching scheme as a whole. However, this was a difficult process and its solution will demonstrate the pros and cons of the implementation.

To increase the amount of training when learning each of the individual templates shifts in each training image were performed. By shifting the training images by a single pixel in the up, down, left and right directions significant background noise could be removed from the learned template. This was critical for the mid-stride and briefcase templates that had a substantially smaller amount of training images than the standing template. This effectively increased each set of pose training images by four times.

Haar wavelet template matching was also restricted to larger scales. By doing this, the number of false positives was drastically decreased. As mentioned previously, smaller scale matching resulted in more false positives. Since pedestrians in

the foreground of an image or video are of greater importance than those further in the background it made sense to impose this restriction. If higher video resolutions could be obtained then extending the template matching to smaller scale would be reasonable and easily achieved.

Since template matching is not the most accurate of techniques it was determined that it should be used to obtain an initial starting point for where pedestrians might be. False positives were acceptable because they would later be filtered out by passing the regions of interest to the HOG detector which would sort the false positives from the actual matches. Since HOG takes a considerable amount of computational power, which includes numerous iterations and recursive calls, template matching can greatly reduce the amount of image regions that HOG has to search.

Initially, a single pedestrian template was used to determine matches within an image. However, though decent results were achieved, there was still significant room for improvement. Specifically, pedestrians which were clearly defined were not being detected. It was determined that these pedestrians were substantially different from the standing pedestrian template developed. As such, multiple pedestrian templates were learned and used in the template matching scheme. By incorporating more templates of various poses it was possible to decrease the amount of pedestrians overlooked by the matching algorithm. The pedestrian templates learned from the training data set are shown in Figure 16. Each of these templates helps to define a unique pose which is considerably different from any other template. Some of the templates did not have enough data to explicitly and clearly define which could achieve optimal results. This was a restriction due to the training data set.

A semi-adaptive threshold was used as a means of attempting to adjust the threshold for each template scale and image. In essence, the max absolute cross correlation coefficient was defined for each template at a given scale for a particular image. A match could then be determined based on a percentage of this max absolute cross correlation coefficient. Absolute cross correlation coefficients occurring in the top  $x$  percentage, where  $x$  is user defined, would be considered a match for the template at the given scale. In this research, only the top one percent was chosen as a match. This implementation has its advantages and disadvantages. The advantage was that it helped to roughly adapt the threshold for various conditions. The disadvantages, however, was that it was forced to define a pedestrian at each template scale, even if there wasn't which generated false positives. Nevertheless, the amount of matches it was able to detect outnumbered the amount of false positives making it a plausible solution.

Overall, using Haar wavelets to define a template used for pedestrian matching did adequately. It seemed to produce a great deal of false positives along with actual matches. There was a great deal of variability associated with the technique. However, it significantly reduced the regions containing possible pedestrians which helped to diminish the amount of computations needed downstream in the architecture. Im-

provements could be made to the templates by using more training data to help reduce the noise of each of the templates defined. More training data would also help to highlight distinct features of the templates in more detail.



Fig. 17. Haar wavelet pedestrian detection on a sample image.

## VI. DISCUSSION

In this paper we described an object detection framework for fast pedestrian detection, however it is useful and important to note that the detectors are by no means limited to detecting solely pedestrians. Given the appropriate training data, this modular detection framework can be used to detect any object.

Moreover, the detector was able to focus on particular areas that have pedestrians in them using comparatively cheaper, coarse-grained algorithms execute in parallel. This markedly cut down on the time to perform pedestrian detection, and also increased the accuracy of the results.

In addition, we crossed the boundaries between MATLAB and OpenCV/C++, thereby granting us the power to use whichever technology lent itself best to the task at hand. Also, by having all code associated with the project invocable via C++, it was possible to use GPU computing frameworks such as CUDA to massively parallelize and subsequently speed up the detection process.

It is also important to note that our research with APHOG crossed the boundaries between part-based and holistic algorithms by incorporating the strengths of each.

Lastly, we gained a better appreciation for what works well for pedestrian detection and what does not! Pedestrian detection is an extremely difficult and computationally-intensive operation, and we look forward to improving this algorithm to be even more efficient and accurate in future revisions. Future areas of work include (but are not limited to) machine-learning & validation for dynamic gradients, testing & refining new coarse-grained part-based detectors and improving the statistical models used to predict pedestrian movement.

## ACKNOWLEDGMENT

The authors would like to thank the OpenCV and NVIDIA CUDA teams for their efforts in Open Source community.

Without them this research would not be possible.

#### REFERENCES

- [1] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In Proc. CVPR, Volume 1, Pages 886-893, 2005.
- [2] S. Stalder, H. Grabner, and L. Van Gool. Cascaded Confidence Filtering for Improved Tracking-by-Detection. In Proc. ECCV, Volume 6311, Pages 369-382, 2010.
- [3] D. M. Gavrilu. The Visual Analysis of Human Movement: A survey. Journal of Computer Vision and Image Understanding (CVIU), 73(1):8298, 1999.
- [4] Q. Zhu, S. Avidan, M. Yeh, and K. Cheng. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In Proc. CVPR06 2006, Pages 491-149, 2006.
- [5] Oren, M.; Papageorgiou, C.; Sinha, P.; Osuna, E.; Poggio, T.; , "Pedestrian detection using wavelet templates," Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on , vol., no., pp.193-199, 17-19 Jun 1997
- [6] Fleet, Patrick J.; The Discrete Haar Wavelet Transform Joint Mathematical Meetings, January 2007 <http://cam.mathlab.stthomas.edu/wavelets/pdffiles/NewOrleans07/HaarTransform.pdf>