HMAX Image Processing Pipeline with Coupled Oscillator Acceleration

Brandon B. Jennings[1], Reggie Barnett[3], Chet Gnegy[3], John A. Carpenter[1], Yan Fang[3], Donald M. Chiarulli[2], Steven P. Levitan[3]

Computer Engineering Graduate Program[1]
Computer Science Department[2]
Department of Electrical and Computer Engineering[3]
The University of Pittsburgh
Pittsburgh, PA, 15260

***Abstract*—In this paper we report on the performance of a coupled oscillator based implementation of the HMAX image-processing pipeline. Within this pipeline we have used coupled oscillator arrays to replace traditional Boolean logic with a Degree-of-Match (DoM) function that measures the L2 distance squared between two vectors in an n-dimensional space. We show that this operation can be used in three stages of the pipeline: 1) as a substitute for convolution in filtering operations, 2) as a computational kernel for pattern matching, and 3) as a distance function in a nearest neighbor classification algorithm. In this study, we have modeled the performance of the latter two and report our recognition results over a test set from the NeoVision2 image database.**

Keywords: Image Processing, HMAX pipeline, Coupled Oscillators

I. Introduction

The spontaneous synchronization of coupled oscillator systems was first described by Christian Huygens in 1673 [1]. He noticed that pendulum clocks, when sitting on the same base, would synchronize to the same phase, or to exactly 180 degrees out of phase. Since then, numerous other models of interacting oscillatory systems have been reported, spanning the neural, mechanical, magnetic, and electronic oscillator domains [2] [3] [4] [5] [6].

The goal of this research is to design a computer vision architecture that can exploit the computational abilities of small arrays of weakly coupled nonlinear nano-oscillators. The target application is a low-power, high performance system that emulates the functionality of the human visual cortex. Emerging nano-devices including Spin-Torque Oscillators (STOs) [7]) and Resonant Body Oscillators (RBOs) [8], and vanadium dioxide devices (VO_2) [9] with coupling based on magnetic, substrate and direct electrical interactions are the enabling technologies for nano-scale low-power devices which we are targeting for future implementation. Other research has demonstrated computational primitives that use coupled arrays of nano-oscillators for pattern matching computations[10], and in a variety of associative memories [11][12][13].

This work was supported in part by the National Science Foundation under grants DMR-1344178, CCF-1317373 and the DARPA/MTO UPSIDE program.

This research aims to extend these results to the architectural level by modeling a fully functional Image Processing Pipeline (IPP) capable of recognizing objects extracted from a scene by a front end saliency algorithm [14]. In other words, given an image, a saliency algorithm locates the interesting objects and passes sub-image "chips" to the IPP. The IPP in turn identifies the object encapsulated by the chip as a member of one of a set of previously trained object classes.

Our study is based on a well know and widely studied IPP called HMAX [15] [16] [17] [18]. Among the significant characteristics of HMAX are early stage filtering operations that support rotation and scale invariance, followed by a pooling and template matching stage using learned templates in an object library and an N-way classification stage at the output. In other work [19] we have shown that coupled oscillator clusters can provide equivalent functionality to the convolutional filters of the early stages of the pipeline. In this paper, we show that oscillator clusters can perform equally well in the template matching and classification stages.

We begin with a discussion of the functional model of an oscillator cluster that we have adopted. Although it is not linked to any specific technology, it is designed to match real-world constraints for cluster size, area, and power taken from the literature and from communications with colleagues [20]. Next, we give a brief overview of the HMAX pipeline and the particular version that we have implemented. This is followed by results for object recognition for image chips taken from the NeoVision2[21] database. Finally we discuss our conclusions and future research.

II. Coupled Oscillator Model

The oscillator model used in this study is a simple 2-port voltage controlled oscillator with an input control port and bidirectional output/coupling port. Figure 1 shows a cluster of these oscillators together with a Degree of Match (DoM) circuit built by direct electrical coupling of the output ports of the oscillators in the array and a rectifier/integrator which detects the peak voltages from the common summing node. The input to this circuit is two vectors of analog (pixel) voltages $(v_1...v_n)$ and $(v_1'..v_n')$. Each oscillator is driven by the pairwise difference of the individual voltages *(v_i'-v_i)*. The oscillator outputs are directly coupled though a resistor network and the voltage at the common node is integrated as a measure of the relative synchronization of the oscillators and hence the degree of match of the input vectors. This circuit is designed assuming relativity small oscillator clusters with sizes on the order of 16 to 64 oscillators. This is consistent with both current technology limitations of the target nano-scale oscillators and support for a direct electrical coupling structure. One of the findings of this study is that clusters of this size are more than adequate for building an effective IPP.

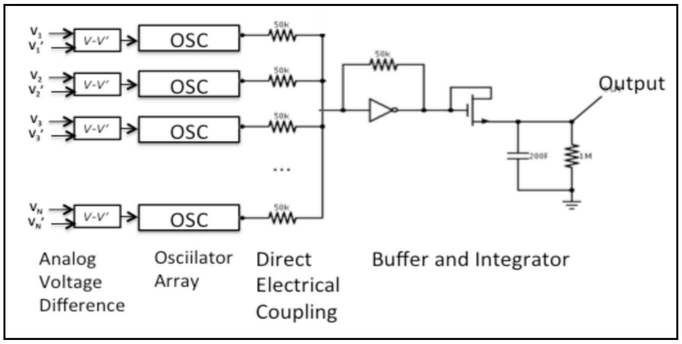

Figure 1: Coupled Oscillator Implementation of a Degree of Match (DoM) circuit

Since we do not have access to nano-oscillator clusters at present, we validated this circuit in a previous study using CMOS ring oscillators [22]. Figure 2 and Figure 3 show a portion of the data from that study. Figure 2 is the voltage output versus time for a three-oscillator cluster under each of two voltage configurations for the inputs. In the configuration labeled A, the three input voltages match at *.6V*. In the configuration labeled B, V_0 and V_1 match at .6, and V_3 is equal to *.3V*. After about 40ns (for this CMOS implementation) the integrator output stabilizes at different voltages for each case. Figure 3 generalizes this result with a plot of the integrator after 40ns versus a sweep of input voltage for V_3 while holding V_0 and V_1 = .6. Note that case A from Figure 2 is labeled at the peak and case B on the left shoulder of the plot.

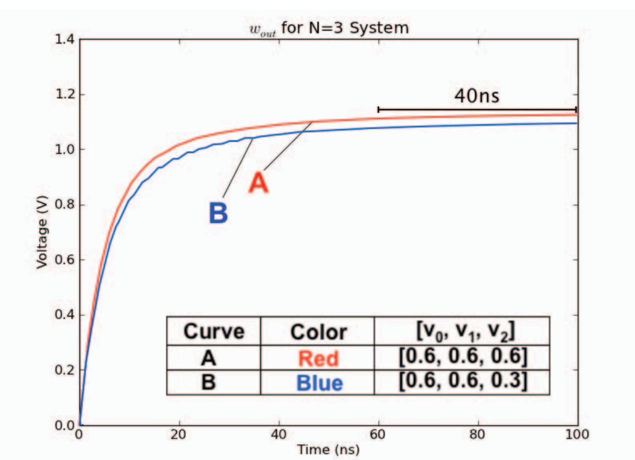

Figure 2: Integrator Output versus time for two input cases of a three oscillator cluster if CMOS ring oscillators.

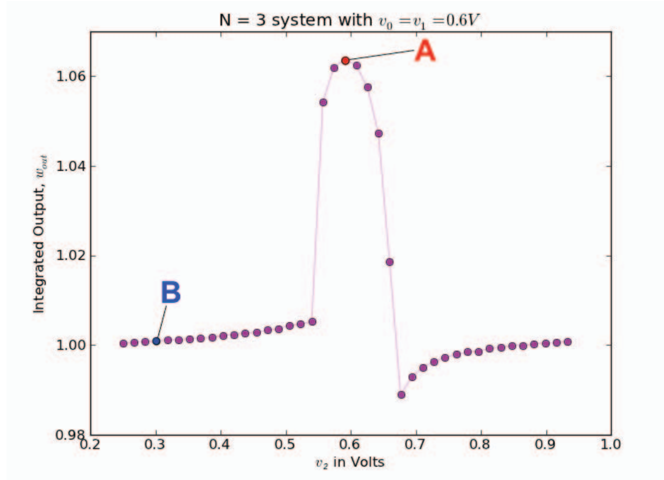

Figure 3: Integrator Output Voltage after 40ns versus a sweep of the input voltage V3. (V_0 and V1=.6)

Thus the sampled integrator output is a quantitative measure of the degree of synchronization among a cluster of coupled oscillators, which corresponds to a higher degree of match of the input vectors. A higher DoM indicates better synchronization and a lower DoM indicates less synchronization. However, if we make a simple observation that DoM is functionally the inverse of the algebraic distance between the two input vectors, this leads to a model for the (now inverted) parabolic region of interest around the match that corresponds to square of the L2 (Euclidean) distance between the vectors. Based on this observation we have adopted the L2 distance as our MATLAB model for the output of an oscillator cluster in the simulations that follow. While not exact, this is reasonable approximation for the output oscillator clusters shown above given a properly designed operating region.

III. HMAX OVERVIEW

We move now to the architecture of the IPP. We have chosen the HMAX pipeline [15] [16]. HMAX has been a widely studied and well understood in the image processing community and is particularly notable for its rotation and scale invariance. Our model is based largely on the Mutch and Lowe[23] implementation with substitutions for oscillators in the key computations kernels and substituting a nearest neighbor classifier (vs a SVM) in the final stage of the pipeline.

The input to the HMAX IPP is an "image chip" which is a salient sub-image from a scene such that the chip contains an object of interest. The output is a classification into one of *N* object classes against which the system is trained. Figure 4 shows an overview of the four pipeline stages from input to output as described below.

IM Stage- In this initial stage, a set of progressively scaled copies of the image chip are created. This set of scaled images is referred to as an image pyramid. In our model, the input chips are 120x120, and are scaled to a 4 level pyramid of 120x120, 84x84, 60x60, and 50x50.

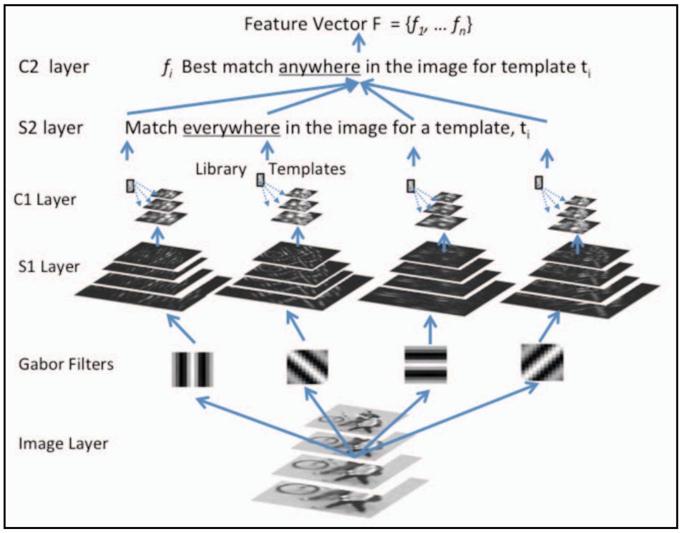

Figure 4 HMAX Inage Processing Pipeline (classifier not shown)

S1 Stage – In this step, Gabor filters in four orientations, horizontal, vertical, and both 45° diagonals are applied. This creates four pyramids that contain representations of the image at multiple orientations and scales.

C1 Stage – This is a pooling stage, where the resolution of each pyramid is reduced by a pooling filter that replaces each region where it is applied by the maximum intensity pixel in that region. In our model, depending on the size of the pooling filter and step size of the scan, this can reduce the size of the image chips dramatically. For this work, the pyramids are pooled to about 1/3 of the original resolution.

S2 Stage – This is the pattern matching stage of HMAX. The pipeline has a stored library of image templates that were extracted from the C1 layer of training images during the training phase. Template patch sizes may vary but are typically between (4x4) and (16x16). Each template contains a set of patches, one patch taken from each orientation pyramid. During this stage, each of these template patches is scanned over all positions at all scales and orientation of the pooled C1 layer representation of the test image..

C2 stage – In this stage, the scanned outputs of the template patches are pooled into a set a features. The 'best match', is selected and becomes the feature value corresponding to that library template.

Classifier Stage - The training features for this classifier come from a training set of images run through the same pipeline steps and matched against the same template library. The training images define N classes of objects and the output of the classifier is the predicted class for each of test image chips. Mutch and Lowe use an all-pairs SVM classifier; we have chosen nearest neighbor to facilitate the use of oscillator clusters in this stage.

IV. OSCILLATOR BASED HMAX MODEL

To demonstrate that HHAX can operate on a computing platform of loosely coupled non-linear oscillators, we have substituted our L2 distance based DoM model into the computation kernels of the S1, S2, and Classifier stages of the HMAX pipeline. Although they are not explicitly modeled with oscillators, the kernel of the C1 and C2 layers are MIN and MAX filters that are also compatible with oscillator based computation. The specific changes are as follows:

1. *DoM Is substituted for GRBF in the S2 layer.* The replacement of the Gaussian Radial Basis Function used by HMAX with the oscillator based DoM distance is a straightforward step. In this case, the *MAX* of a GRBF for pairwise templates in replaced by the MIN of the L2 distance.

2. *Change the classifier algorithm from all-pairs SVM to Nearest Neighbor approach with DoM as the distance metric.* On a conventional logic platform, nearest neighbor classifiers are significantly more computationally intensive that other algorithms given the number of pairwise vector comparisons. On an oscillator based platform this is less of a concern than the limits on the size individual oscillator clusters, driven by size of the feature vector (or segments thereof) that are being compared.

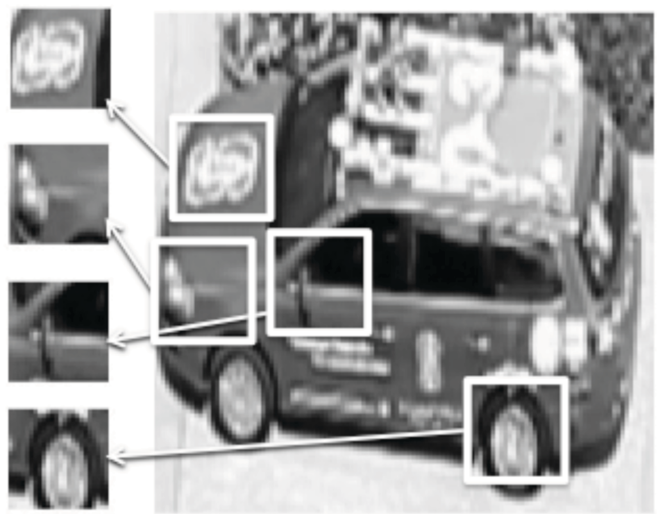

Figure 5 Template patch annotations

V. TRAINING THE HMAX PIPELINE

Training an HMAX pipeline takes place in two phases and in our model, both of the training and testing phases incorporate the oscillator based DoM function. In the first phase, small patches are extracted from representative image for each class and stored in a template library. Patches are extracted at the C1 layer and are typically small, ranging in size from 4x4 to 16x16. Once the template library is built, the second phase builds the training data for the classifier by running a set of training images through the pipeline using the new template library. In the classifier training data there is a one-to-one correspondence between elements in the feature vector and members of the template library.

Image Chip Dimensions	120x120
IM Pyramid Scales	4
IM Pyramid Dimensions	(120x120) (84x84) (60x60) (50x50)
Annotation Patch Sizes	(48x48) (24x24) (12X12)
S1 Number of Filters	4
S1 Filter Dimensions	(8x8)
S1 Pyramid Sizes	(113x113) (77x77)(53x53)(43x43)
S1 Orientations	4
S1 Orientation Angles	0° 45° 90° 135°
C1 Pooling Filter Type	MAX
C1 Pooling Filter Size	(6x6)
C1 Pooling Filter Step Size	3
C1 Pyramid Sizes	(37x37) (25x25) (17x17)
S2 Template Library Patch Sizes	(4x4) (8x8) 16x16)
C2 Feature Set Size	Size of Template Library
C2 Pooling Filter Type	MIN

Table 1: Model Parameters for HMAX Pipeline

As HMAX is a template based algorithm, the key to the accuracy and performance of an HMAX pipeline is the selection of the template patches. Our model allows for multiple template patch sizes and each patch is manually extracted from annotations of the training images. An example of these annotations is shown in Figure 5. Automated optimization of the template library is an open research question[24] that is not addressed in this paper. However, it is clear that hand optimization allowed us to significantly reduce the size of the template library which is a key requirement for an implementable oscillator based design.

During training, each annotation is scaled to fit the pooled image sizes in the C1 layer. Separate patches are extracted from all four orientation pyramids at the annotation location. Thus each template consists of four patches. However, in this implementation each annotation creates a separate template library entry for each scale in the C1 pyramid. For example, if the C1 layer has three scales in the image pyramid and four orientations, the HMAX training script creates three library templates, each with four image patches. These are in turn pooled into three features at the C2 layer.

The design of the template library and the C2 pooling layer has implications for the cluster size of the oscillators in the nearest neighbor classifier. There is a tradeoff between the number of matching operations that are pooled for each template and the feature set diversity (and thus discriminatory power) of the classifier. The cost is a larger feature set size which increases the dimensionality of the DoM function in the classifier.

VI. HMAX Models and Simulation Parameters

We conducted three studies using our HMAX model implemented in MATLAB. The first study is intended to validate our results by comparing the performance of the oscillator based HMAX pipeline, using the L2 distance DoM function for template matching in the S2 layer and nearest neighbor classification, to an implementation of the Mutch and Lowe pipeline design using GRBF template matching and all-pairs SVM classification.

In the second and third studies we focus on design parameters that directly impact the oscillator cluster size requirements of an oscillator based implementation. The second study deals with the patch size of the library templates. Template patch size determines the cluster size in the in S2 layer. The third study explores the design space of library template size and training set size. HMAX training is done in two phases, one phase to create the template library and a second to build the classifier training data. In a nearest neighbor classifier, the dimensionality of the L2 distance computation is determined by the size of the feature vectors and in HMAX there is a one-to-on correspondence between features and library templates. Thus, it is desirable in a oscillator based implementation to minimize the template library size built in phase one at the expense of the size of the training set built in phase two.

Figure 6: Slide tray of representative image chips for each object class in NeoVision2 Database[21]

Table 1 lists the various parameters selected at each level of the HMAX pipeline in our models. Our studies use test and training images sets that were randomly and independently selected from the NeoVision2 Database[21]. This database contains image chips for five classes of objects, Cars, Trucks, Buses, People and Cyclists. Figure 6 is a slide tray of representative images chips from each class. All of our test datasets contain equal numbers of randomly selected image chips from each class.

All results are plotted in terms of the F_1 score[25] of the classifier. The F_1 score of a classifier is the harmonic mean of the ***precision*** (a.k.a. Positive Predictive Value) and ***recall*** (a.k.a. Sensitivity) of a classifier such that

$$F_1score = \frac{2 * Precsion * Recall}{Precision + Recall}$$

where

$$Precision = \frac{True\ Positives}{True\ Posotive + False\ Positives}$$

and

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

VII. Results

Figure 7 shows the results from our first study that compares the oscillator based HMAX model to the Mutch and Lowe model. Both models were run over identical test and training image chips with 60 templates per class in the HMAX library, 8x8 patches, and 100 test images. F_1 scores are plotted for training set sizes of 20 to 200 images. Comparison of these results shows that the oscillator implementation provides comparable accuracy to the convential GRBF/all-pairs SWM design.

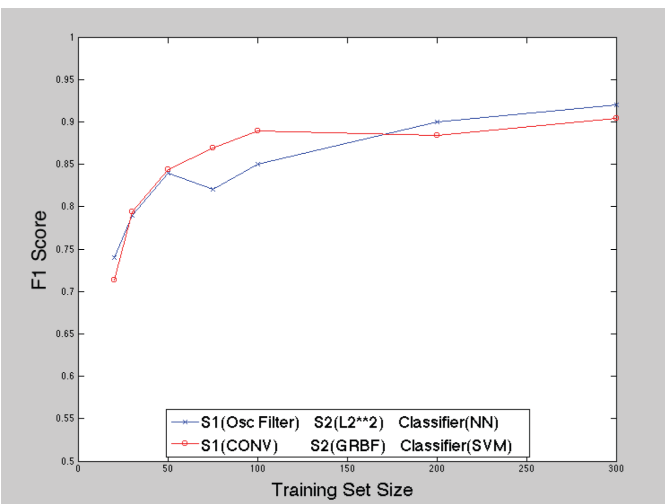

Figure 7: Conventional HMAX versus oscillator based HMAX performance

Figure 8 shows results for the oscillator based model run under the same parameters with separate results for each object class and overlay plots for each patch size. This data suggest that the impact of patch size is minimal and specific to the object class. Three of the five object classes in our database cars, trucks, and buses, seen to perform equally well for small (4x4) versus large (16x16) patch sizes while cyclist and people tend to do better with larger patch sizes. This is consistent with the conjecture by Mutch and Lowe that large patch sizes tend to work better on more highly textured image chips. In any case, this data suggests that feature set optimization can favor small patch sizes with minimal cost in performance.

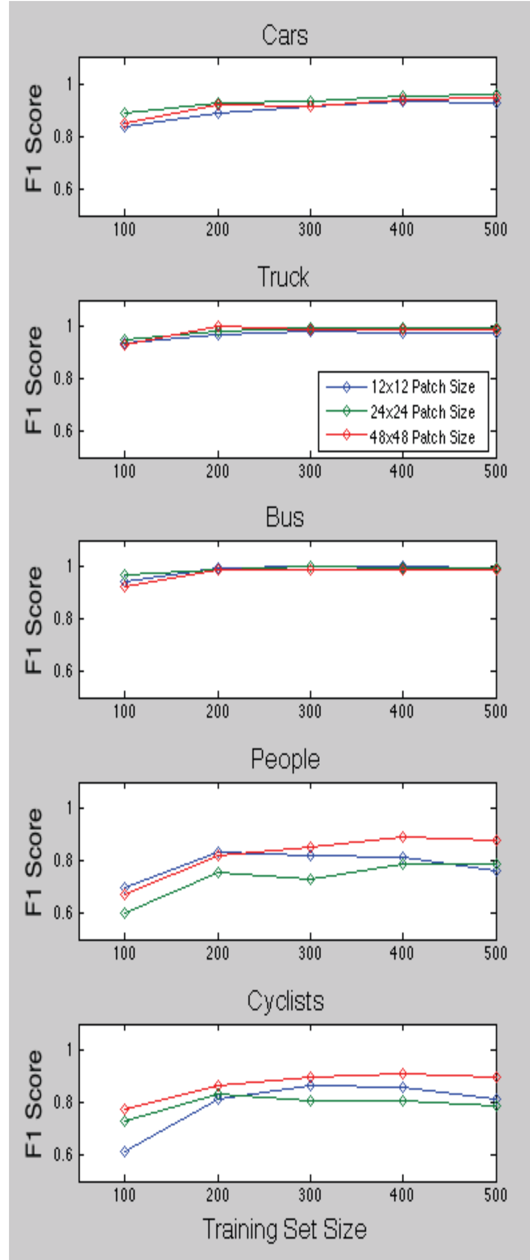

Figure 8: F_1 Score versus training set size plotted separately for each object class with multiple plots for models with patch sizes of (4x4), (8x8) and (16x16).

Figures 9 and 10 examine the tradeoffs relating to cluster size in the nearest neighbor classifier. Figure 9 is a plot of F_1 versus training set size with multiple plots for template library sizes from 75 to 450 templates. Figure 10 replots the same data, this time versus template library size with multiple plots overlaid for training set sizes of 20 to 300 training images.

Figure 9 shows as expected that classifier performance increases with the size of the training set and levels-off above 200 training images. Conversely the plots in Figure 10 are relatively flat across all template library sizes and are spaced on the y axis consistently with the data in Figure 9. This suggest that performance is significantly more dependent on training set size than template library size. This means that an effective classifier can be built with a relatively small number of features per class and thus small cluster sizes for the oscillator based nearest neighbor classification. Increasing the training set size increases the potential number of oscillator clusters required but not the size of each oscillator cluster. Thus a high performance design that is consistent with current technology constraints can be implemented.

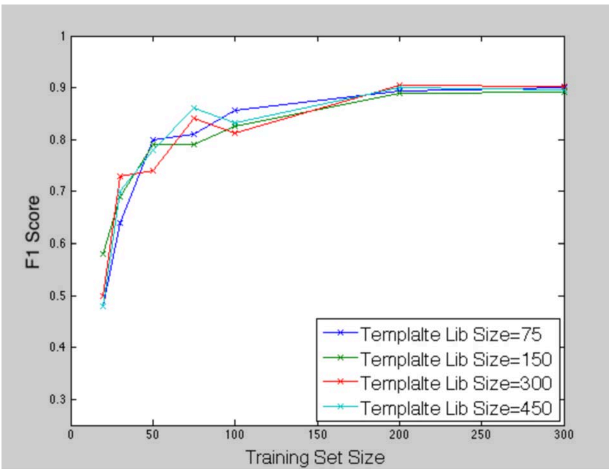

Figure 9: F_1 score versus training set size for plotted for multiple Template Library sizes (75-750 templates).

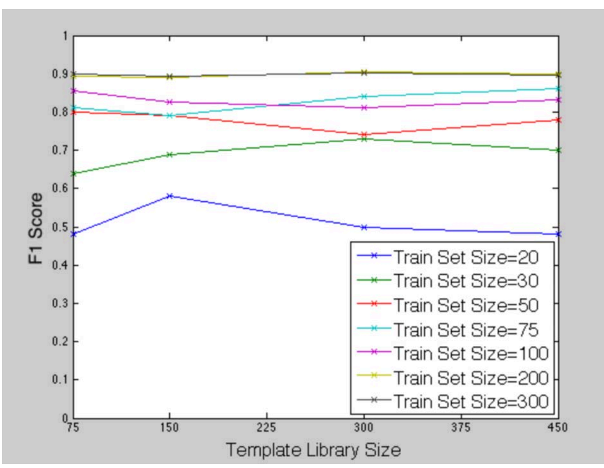

Figure 10: F_1 Score versus Template Library size plotted for multiple training set sizes (20-300 images).

VIII. Conclusions and Future Research

We have demonstrated a model of an HMAX image processing pipeline with oscillator based DoM functions substituted for the key computation kernels. Our simulations have analyzed the key parameters related to the design constraints associated with oscillator technology and DoM circuit design. We have shown comparable performance to conventional architectures can be achieved within these technological constraints. However, optimization of template selection in training is a key requirement and must be the focus of further research.

References

[1] Horologium oscillatorium (1673)

[2] Han, S. K., Kurrer, C., and Kuramoto, Y., Dephasing and bursting in coupled neural oscillators. *Physical Review Letters, 75* (17), 3190, 1995.

[3] Horvath, A., Synchronization in cellular spin torque oscillator arrays. *Cellular Nanoscale Networks and their applications (CNNA), 13th International Workshop on. IEEE*, 2012.

[4] Shibata, T., *et al*., (2012). CMOS supporting circuitries for nano-oscillator-based associative memories. *13th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)* , 1-5. Turin, Italy, August 29-31, 2012.

[5] Hoppensteadt F.C. and Izhikevich E.M. (2001) Synchronization of MEMS Resonators and Mechanical Neurocomputing. *IEEE Transactions On Circuits and Systems I,* 48:133-138

[6] Van Der Pol, B. (1927). Forced oscillations in a circuit with non-linear resistance. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 3* (13), 65-80.

[7] Kaka, S., Mutual phase-locking of microwave spin torque nano-oscillators. *Nature , 437* (15), 389-392, 2005.

[8] Weinstein, D., and Bhave, S. A. A resonant body transistor, Nano Lett. 2010, 10, 1234–1237 DOI: 10.1021/nl9037517.

[9] Nikhil Shukla, et al., , Synchronized charge oscillations in correlated electron systems, Scientific Reports 4, Article number: 4964 doi:10.1038/srep04964 14 May 2014.

[10] Levitan S. P., et al., Non-Boolean associative architectures based on nano-oscillators," 13th IEEE Int'l Workshop on Cellular Nanoscale Networks & Their Applications (CNNA 2012), pp. 1-6, Turin, Italy, August 29-31, 2012.

[11] Csaba, G., Spin torque oscillator (STO) models for applications in associative memories. *Cellular Nanoscale Networks and Their Applications (CNNA)*, 2011.

[12] Nikonov, D. D., Coupled-oscillator associative memory array operation. *arXiv preprint arXiv , 1304* (6125), 2013.

[13] Csaba, G., and Porod, W., Computational study of spin-torque oscillator interactions for non-Boolean computing applications. *Magnetics, IEEE Transactions on , 49* (7), 4447-4451, 2013.

[14] Acharya, T., & Ray, A. K. (2005). *Image Processing: Principles and Applications.* John Wiley & Sons.

[15] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In CVPR, San Diego, June 2005.

[16] Al Mashri, A, A hardware architecture for accelerating neuromorphic vision algorithms. *Signal Processing Systems (SiPS), 2011 IEEE Workshop on. IEEE*, 2011 .

[17] Kestur, S., et al., Emulating Mammalian Vision on Reconfigurable Hardware, *Symposium on Field-Programmable Custom Computing Machines (FCCM), 2012 IEEE 20th Annual International*

[18] Karimimehr, S., A novel face recognition system inspired by computational neuroscience, *EUROCON, 2013 IEEE.*

[19] Carpenter, John, Image processing pipeline based on coupled oscillator Models. Master's Thesis, University of Pittsburgh, 2014

[20] Matthew Cotter, et al., Computational architectures based on coupled oscillators, *IEEE Computer Society Annual Symposium on VLSI*, Tampa, FL, July 9-11, 2014.

[21] *Neovision2 dataset - iLab - University of Southern California*., Retrieved November 12, 2013, from http://ilab.usc.edu/neo2/dataset/, 2013

[22] Levitan, S. P. et al., Associative processing with coupled oscillators, *International Symposium on Low Power Electronics and Design* (ISLPED'13), Beijing, China, September 4-6, 2013.

[23] Mutch, J., & Lowe, D., Multiclass object recognition with sparse, localized features. *Computer Vision and Pattern Recognition , 11* (18), 17-22, 2006.

[24] Sungho Park, et al., Saliency-driven dynamic configuration of HMAX for energy-efficient multi-object recognition, *IEEE Computer Society Annual Symposium on VLSI (ISVLSI),* 2013

[25] http://en.wikipedia.org/wiki/F1_score