Fast Extraction of Feature Salience Maps for Rapid Video Data Analysis

Nikos Pitsianis and Xiaobai Sun
Department of Computer Science, Duke University
Durham NC 27708, USA
{nikos, xiaobai}@cs.duke.edu

Introduction

Many-core processors such as the graphics processing units (GPUs) are among the first of choice for generating feature maps in rapid analysis of video data [3]. To cope with the volume and rate of video data, one extracts, frame by frame, salient information in multiple feature dimensions such as in color, orientation(edge), shape, texture and motion, among others, with or without selective tuning [5, 9] or feedback, depending on the application contexts as well as attentional or intentional guidance. The feature salience maps are used, in separation or integration, to assist human in visual search tasks or facilitate automatic visual search tasks, such as target indication, object recognition, tracking of moving objects [6]. A feature salience map is a topographic representation of the feature-specific salience at every location in the visual scene. Multiple feature maps may be combined together, based on the feature integration theory [8], to render a comprehensive and compressed saliency map [2], see for example, Figure 1 from [4]. Extracting multiple feature maps and integrating them into a saliency map involve many filtering steps at multiple spatial scales, require more memory space, and demand low latency. Nonetheless, the extraction and integration process is rich in array operations. A systematic approach is presented here for fast extraction of many feature maps in intensity, color and orientation from each image frame at video rate. It includes algorithmic variants for filtering directly in the image domain or via the Fourier domain, parallel computation within each filtering process and across different ones, and a configuration schema for algorithm selection and scheduling based on the image size, filter selection, and the performance of the primitive operations on GPUs. We comment briefly on the remaining challenges in extracting salient motion features, feature map integration and visual search tasks above and beyond the saliency analysis.

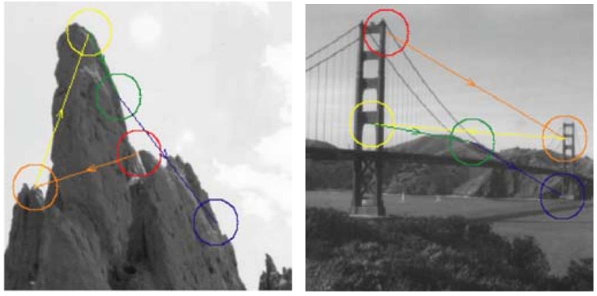

Figure 1: The five most salient locations are shown in the red, orange, yellow, green and blue circles, with the arrows from one salient point to the next less salient. Notice the prevalence of representation by junctions and end-stops. The images and analysis are from [4].

Feature Maps: generation and integration

In feature analysis, an image frame is decomposed or analyzed in multiple feature dimensions. A typical procedure for generating a feature salience map may be described in three stages. We describe the procedure with the orientation feature in particular. In this case we may assume that an image frame at input is an $m \times n$ data array $I(x_i, y_j)$ with grayscale values. At stage 1, the orientation feature is extracted from the image and represented by a Gabor pyramid, which contains the orientation contrast components $O(\sigma, \theta)$ obtained from I with a bank of 2D Gabor filters [1], for instance, at multiple spatial scales $\sigma = 0, 1, \cdots, 7$ and multiple local orientations $\theta = 0^o, 45^o, 90^o, 135^o$. The image I is at the scale level 0. Every orientation component $O(\sigma, \theta)$ can be obtained by a convolution of I with the corresponding Gabor filter $G(\sigma, \theta)$, see Figure 2, followed by local normalization. For simplicity, we have omitted the frequency selection and phase selection, which are important for certain other features. The localized normalization is necessary and can be described also in terms of convolutions with binary filters of local support.

At stage 2, the orientation pyramid undergoes a local-integration process. Specifically, the feature contrast components $O(\sigma, \theta)$ are transformed by what is referred to as the *center-surround* operations, based on a model for visual receptive fields,

$$\mathfrak{O}(c, s, \theta) = |O(c, \theta) \ominus O(s, \theta)|, \tag{1}$$

where $c = 2, 3, 4$ and $s = c+3, 4$ indicate the center and surround scales, respectively, and $\ominus$ denotes the cross-scale difference associated with an interpolation scheme. The 32 contrast components on the multi-scale pyramid are transformed into 24 center-surround orientation components. This local integration process has the denoising effect, the transformed components are sensitive to local orientation contrast instead of variation in raw feature magnitude.

At stage 3, the feature-specific components are normalized to a common dynamic range, in order to equalize the weights in integration or summation within feature-specific components, as well as across multiple features. The normalization procedure for each and every component array $\mathfrak{O}(c, s, \theta)$ is a fixed point iteration process. Each iteration step involves a convolution with a 2D difference-of-Gaussian (DoG) filter, which yields strong local excitation, counteracted by broad inhibition from neighboring locations, and a truncation of the off-range value.

Additionally, there is a Gaussian pyramid in intensity, among other features, at stage 1. Associated with an RGB image are also four Gaussian pyramids for four different color channels, as combination of the RGB values at scale level 0. We omit the detail. Thus, there are 72 filtering processes at stage 1 (each has a convolution with a feature filter

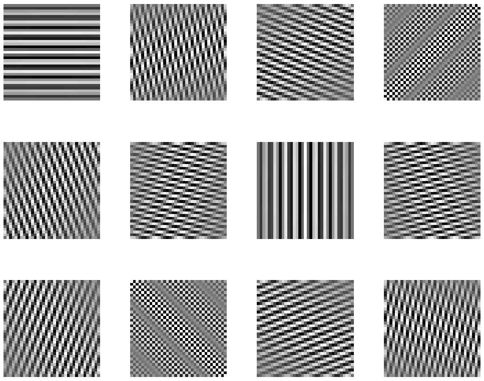

Figure 2: Pictorial illustration of Gabor filters for extracting orientation contrast

followed by a normalization) and 42 feature maps at stage 2 to render one integrated saliency map, not including the other feature maps.

Parallel generation and integration of feature maps

The extraction and integration of multiple feature maps can be carried out in various parallel fashions. The basic operations in stages 2 and 3 are the convolution of a data array with a filter of local support, for example, from 7×7 to 37×37 at different spatial scales. The DoG filter for each feature map normalization is relatively larger. There are two basic convolution approaches, one is the direct convolution in the image domain, the other is via the Fourier domain, based on the convolution theorem. In the direct approach, there are two algorithmic variants. One is frame centic and the other is filter-pattern centric, corresponding to the row-wise and column-wise versions for the matrix-vector multiplication, respectively. Here, the convolution matrix is in its natually compact representation as each row is a shift of the filter pattern. Their parallel implmentations on GPUs give rise to different thread assignments, spatio-temporal partition in data and tasks for parallel computation.

In the approach via the Fourier domain, the same image transformed in the Fourier domain can be used for multiple convolutions with different filters at the same scale. When there is sufficient on-board memory, such as in the recent GPUs (4GB) of the Tesla and Fermi families by NVIDIA, the filters can be saved in their Fourier representations (Gaussian and Gabor filters remain in their respective function families in the Fourier domain). One can further use the special convolution theorem for real-valued data [7]. The cross-over point in arithmetic complexity between the image-domain approach and the Fourier-domain approach with 2D filters is approximately 9×9 for image size about 512×512. The observed cross-over point on a GPU of Tesla/Fermi is 13×13 with our present implementation of the two algorithm variants, using the CUDA CUFFT library on the NVIDIA GPUs.

Exploiting the available on-board memory, one can have multiple convolutions carried out in parallel at non-overlapping memory buffers. At present, the parallelism at this level is restricted to those of the same filter size and the same image frame size at the same scale (this restriction has been lifted in the new-generation GPUs). We have developed a configuration scheme for filling the available memory space to maximize the number of concurrent convolutions, selecting the convolution algorithms, and scheduling the array operations, based on the parameters for image size, filter size and type, and the profiling of the basic array operations on a particular GPU. With the recent improvements in GPUs, the CUDA programming environment and our approach as described above, the generation and integration of feature maps and can match the video rate, 30 frames per second, with 72 filters and 42 feature maps per second, using two NVIDIA C1060 GPUs.

Discussion

The extraction and use of salient information from static or dynamic images are recent and active research topics. The computation based on an extraction model serves two purposes. One is to test and validate the underlying neurobiological model for certain visual function in the visual system of the primate brain. The other is to exploit the new understanding and model(s) for developing and improving artificial vision systems. GPUs have been used in saliency analysis for both purposes. Remaining challenges include the generation of motion features, which are much more computation intensive, and the visual tasks at the higher levels, such as segmentation, object recognition, tracking of moving targets. At higher levels, the representation of data tend to be sparse, irregular, although still structured in certain ways. It remains to be seen whether or not the high-level processing steps can be carried out efficiently on GPUs.

This research is supported in part by the Defense Sciences Office of the Defense Advanced Research Projects Agency (DARPA).

References

[1] DAUGMAN, J. G. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Trans. Acous. Speech. Sig. Proc. 36*, 7 (1988), 1169–1179.

[2] ITTI, L., AND KOCH, C. Computational modeling of visual attention. *Nature Reviews Neuroscience 2*, 3 (Mar 2001), 194–203.

[3] ITTI, L., KOCH, C., AND NIEBUR, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (Nov 1998), 1254–1259.

[4] MUNDHENK, T. N., AND ITTI, L. Computational modeling and exploration of contour integration for visual saliency. *Biological Cyberntics* (2005).

[5] PARKHURST, D., AND NIEBUR, E. Variable resolution displays: a theoretical, practical and behavioral evaluation. *Human Factors 44*, 4 (2002), 611–29.

[6] SERRE, T., WOLF, L., BILESCHI, S., RIESENHUBER, M., AND POGGIO, T. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29* (2007), 411–426.

[7] SUN, X., AND PITSIANIS, N. A special convolution theorem for real data. Tech. Rep. CS TR-2010-03, Duke University, Apr. 2010.

[8] TREISMAN, A. M., AND GELADE, G. A feature-integration theory of attention. *Cognitive Psychology 12*, 1 (1980), 97–136.

[9] TSOTSOS, J., CULHANE, S., WAI, W., LAI, Y., DAVIS, N., AND NUFLO, F. Modeling visual attention via selective tuning. *Artificial Intelligence 78*, 1-2 (1995), 507 – 547.