Spatially and Temporally Dense Object Recognition in Real-time

project by

Ioannis Panousis

Department of Computer Science

University of Bath

supervised by

Dr Matthew Brown

November 12, 2012

Signature of student ....

Signature of supervisor ....

1 Problem Description

This project is an attempt to overcome obstacles in a crucial area of computer vision by exploiting the massively parallel architecture of today's graphics cards. The programming model that will be used to implement this work is the Open Computing Language (OpenCL) [6], introduced in 2008 by the Khronos Group, a very powerful cross-platform and cross-device language standard.

Object recognition within a video sequence has long been at the center of computer vision. Existing algorithms in this area often consist of two components; 1) a tracker and 2) a recognition component. As such, the, initially unknown, moving object is tracked through a video sequence. Second, it will be compared to an existing database of models or templates of objects in order to be recognised as a particular object. Our work proposes an approach that ignores the tracking stage and a full-frame search is performed for a particular known object to be detected. This is aimed to be done in a dense spatial and temporal fashion, in essence for every pixel in a template to be localised in a frame, for every frame in a video sequence.

Previous work in the field of object recognition utilises only a very small subset of the pixels in a template image. The object is recognised within a new image by use of these *sparse interest points*. Ways to describe these points and their local regions vary from Harris corners or SIFT [9], to other effective improvisations within this area such as Rotation-Invariant Fast Feature [15] or Color Distance Transform [13]. There are some vital problems with interest points however. Firstly, it needs to be the case that the same ones need to be reliably detected in more than one appearance of an object, even when it changes slightly. This necessary property for interest points to be useful is known in vision as *repeatability*. Secondly, existing systems show issues when using sparse points with the presence of partial occlusion. When this is the case, either no points can be matched in the given search image or they are incorrectly matched, as highlighted in [7] [8] [14].

Methods to solve online appearance changes include online boosting [4] [12], and online multiple instance learning [1], but admit failure cases. Partial occlusion is dealt with weightings placed on the right interest points, but this is not always reliable [7]. In contrast, we propose to perform the operation of spatially dense template recognition in every frame of a video sequence. This is in attempt to solve the above problems.

In addition, the omitted component of object tracking will be replaced, initially, by a full-frame scan to detect the template object. Previous work in efficient search techniques include pyramid-search in the Gaussian scale-space before SIFT [14]. A similar technique shall be used to perform the full-frame search at low computational cost. This will be done by taking advantage of the structure of feature point descriptor DAISY [16]. The highly real-time dense computation of DAISY has been previously implemented in OpenCL [5].

High level applications that aim to be enabled directly and indirectly by this work include: image retrieval, keyframe retrieval, visual localisation and mapping, and more. Moreover, a range of applications that can be made possible through the dense spatial matching: fine-grain deformable object modelling, 3D space object occupation and occlusion awareness, well-informed motion subspace segmentation through rich number of features and frames, to achieve 3D motion modelling, and so on.

1.1 Objectives

Base objectives:

- Use a template object image provided in advance
- Compute DAISY descriptor images of template and target image
- Match every DAISY feature in the template onto a feature in the target image
- Implement this system in OpenCL
- Perform the operations fast enough for high resolution (>VGA or 640x480) regular frame rate video sequences

Extensions that will be considered depending on the achievement of the base requirements:

- multi-object recognition capability
- motion prediction by consultation of [2]
- motion segmentation with formed feature trajectories by method of subspace segmentation [10]

1.2 System

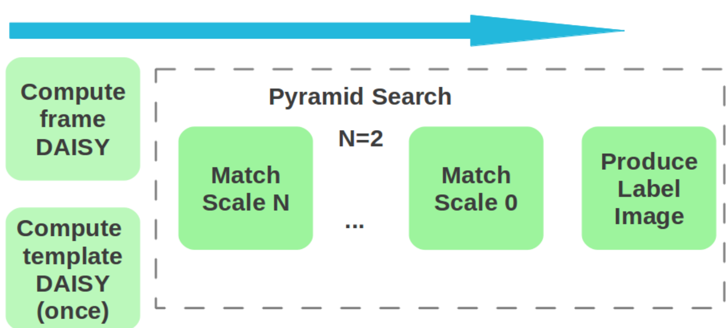


Figure 1: *Functionality that the system will consist of. The boxed section contains the components that need to be implemented.*

The system will follow the logical flow outlined in Figure 1. The existing implementation of dense DAISY computation will be used. What is to be implemented consists of the pyramid search of a set of template descriptors in a target frame. The output has to be a set of assigned labels to the target frame indicating particular template points.

1.3 Work Plan

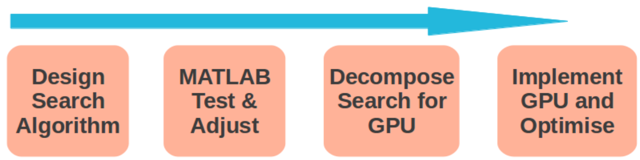


Figure 2: *High-level progress tasks to reach base project goal.*

The work on the proposed system will consist of these high level tasks, in Figure 2. Some milestones need to be set for those as well as for the documentation throughout the project. The first few milestones are listed below. The full set can be seen in more detail within the Gannt chart appended in this document.

Planned milestones	
Task	End date
Literature Review write-up	17/12/2012
Implementation and Optimisation of search on GPU	17/12/2012
Assess Project Goal Extension	31/01/2013
Testing & Benchmarking	28/02/2013
Write-up	31/03/2013

1.4 Challenges to overcome

The, predicted, challenges to overcome during this project include the following.

A - To identify critical parameters of pyramid-search during MATLAB testing and adjust those in pursuit of robustness with respect to:

- object appearance change
- partial occlusion

B - To render the search decomposition optimal for the GPU in terms of:

- the structure, as is, of DAISY descriptors in GPU memory
- efficiently make use of GPU computing resources, i.e. ensure *resource occupancy*
- efficient computation instructions of DAISY descriptor comparison
- efficient memory transactions during comparison of template descriptors within a search neighbourhood

2 Resources

The resources we will take advantage of in the duration of this project will include the following.

2.1 Hardware

- A desktop computer with an NVIDIA GTX660 graphics card. An NVIDIA GTS250 graphics card may be used for additional testing.
- A laptop computer with an ATI Mobility Radeon HD4650 graphics card.

2.2 Software

- The OpenCL SDK implementation for the respective graphics cards
- NVIDIA's Compute Profiler for performance analysis
- KernelAnalyzer for ATI graphics cards
- MATLAB
- Open-source code implementing relevant paper publication algorithms

2.3 Literature

- Books for OpenCL background and programming practices will include; "The OpenCL Programming Guide" [11] and "Heterogeneous Computing with OpenCL" [3]
- Online support community forums may be used for technical troubleshooting or OpenCL-related enquiries, such as the AMD Developer Forum, the Khronos Group forum or other
- NVIDIA and AMD code samples and best practice guides for OpenCL programming

2.4 People

People that could be of assistance during this project are fellow Maths students (reg. algorithms to analyse), friend and colleague Mr Kostas Papaharisiou studying GPU-relevant subjects, and supervisor Dr Matthew Brown regarding the project as a whole.

References

[1] B. Babenko, M.H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632, 2011.

[2] M. Garrigues and A. Manzanera. Real time semi-dense point tracking. *Image Analysis and Recognition*, pages 245–252, 2012.

[3] B. Gaster, L. Howes, D.R. Kaeli, P. Mistry, and D. Schaa. *Heterogeneous Computing with OpenCL: Revised OpenCL 1.2 Edition*. Elsevier Science, 2012.

[4] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting.

[5] Panousis Ioannis. Github - ipanousis (ioannis panousis), January 2012.

[6] Group Khronos. Opencl - the open standard for parallel programming of heterogeneous systems, January 2012.

[7] W. Kloihofer and M. Kampel. Interest point based tracking. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, pages 3549–3552. IEEE Computer Society, 2010.

[8] T. Lee and S. Soatto. Learning and matching multiscale template descriptors for real-time detection, localization and tracking. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1457–1464. IEEE, 2011.

[9] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[10] C. Lu, H. Min, Z. Zhao, D. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. *Computer Vision - ECCD 2012*, pages 347–360, 2012.

[11] A. Munshi, B.R. Gaster, T.G. Mattson, J. Fung, and D. Ginsburg. *OpenCL Programming Guide*. OpenGL Series. Prentice Hall, 2011.

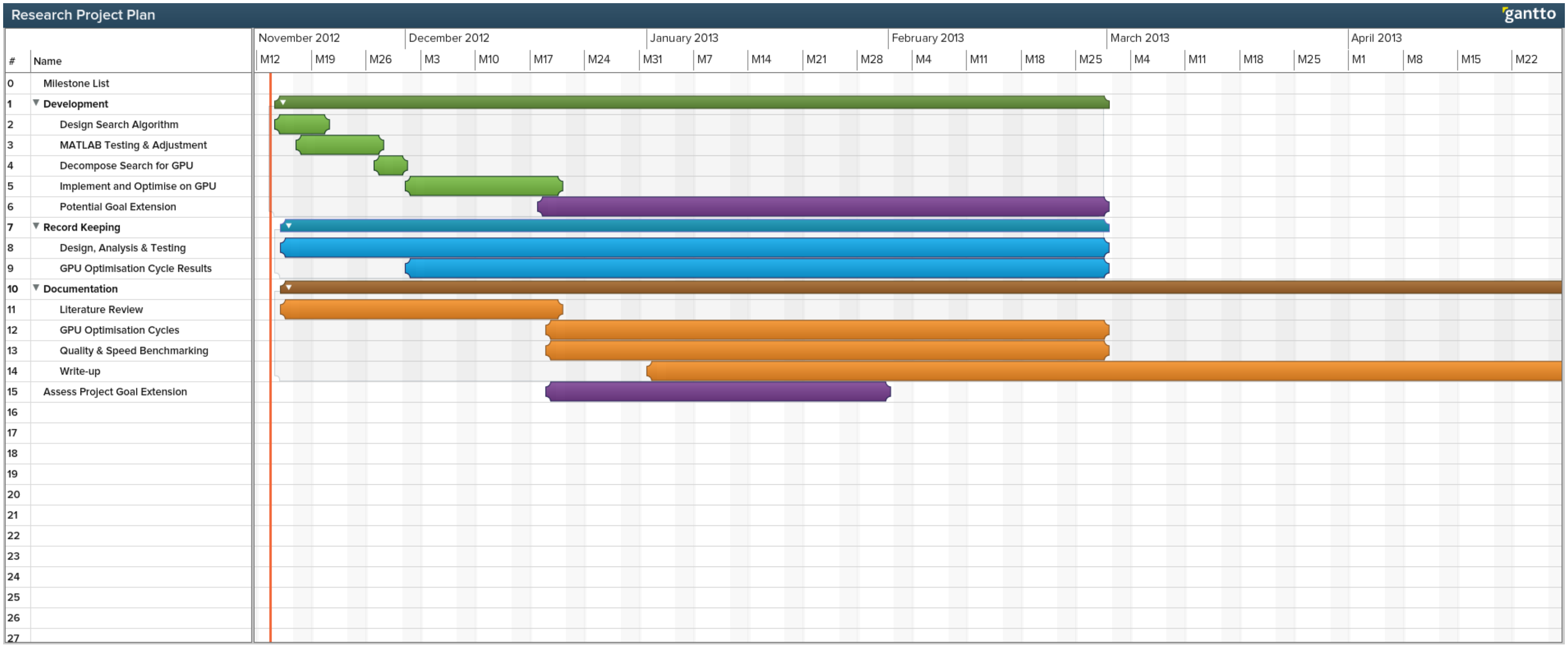
[12] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(3):416–431, 2006.

[13] A. Ruta, Y. Li, and X. Liu. Real-time traffic sign recognition from video by class-specific discriminative features. *Pattern Recognition*, 43(1):416–430, 2010.

[14] D.N. Ta, W.C. Chen, N. Gelfand, and K. Pulli. Surftrac: Efficient tracking and continuous object recognition using local feature descriptors. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2937–2944. IEEE, 2009.

[15] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod. Unified real-time tracking and recognition with rotation-invariant fast features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 934–941. IEEE, 2010.

[16] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):815–830, 2010.


Research Project Plan
gantto
November 2012
December 2012
January 2013
February 2013
March 2013
April 2013
M12
M19
M26
M3
M10
M17
M24
M31
M7
M14
M21
M28
M4
M11
M18
M25
M4
M11
M18
M25
M1
M8
M15
M22
#
Name
0 Milestone List
1 Development
2 Design Search Algorithm
3 MATLAB Testing & Adjustment
4 Decompose Search for GPU
5 Implement and Optimise on GPU
6 Potential Goal Extension
7 Record Keeping
8 Design, Analysis & Testing
9 GPU Optimisation Cycle Results
10 Documentation
11 Literature Review
12 GPU Optimisation Cycles
13 Quality & Speed Benchmarking
14 Write-up
15 Assess Project Goal Extension
16
17
18
19
20
21
22
23
24
25
26
27