A General Toolkit for “GPUtilisation” in SME Applications
FP7-SME-2011-1 - 286545 – GPSME

Final Report: Publishable Summary

I. PROJECT CONTEXT AND OBJECTIVES

1. Project Context

The purpose of this project is to overcome the identified barriers for the SMEs in utilising the GPU technologies by using an outsourced development team (the RTD performers) to create a generic toolkit that SMEs can use to easily and cost-effectively harness the processing power of GPUs and substantially increase their competitiveness in the marketplace.

SMEs (small and medium enterprises) face continued pressure to remain competitive in an age of rapid technological change. This is particularly the case for SMEs with high-tech innovative products whose direct competitors are often large multinational companies that have vastly more resources at their disposal in product development.

The major challenges faced by the SMEs in these high-tech fields relate to a huge growth in data processing requirements through increases in quantity, in resolution, in variety etc. demanded by their applications. This leads to a continual upward pressure on computational resources and processing speed, with major bottlenecks occurring that dramatically limit the applicability of many SME applications. All of the SMEs in this project have found themselves subject to these pressures.

For a SME to maintain a strong foothold in R&D, it must have access to facilities that are powerful, that provide here-and-now availability and that are under its direct control. Local computing power facilitates research exploration by allowing the developers to frequently experiment with computing models and parameters. It also helps to address security and legal problems as the SME's data and software can remain in-house if no use is made of external computing resources.

The initiative of this project came from the demands of the 4 SME participants. While providing services in different areas, they all face a common problem: the quality of their products has been inhibited by a lack of computing power - the product developed is constrained by the computational resources of the likely user base. Hence, the availability and affordability of the equipment necessary to use the SME's products can affect their marketing and become a major obstacle to their competitiveness in the future.

Parallel computing offers fast computing by splitting tasks into small components and distributing them among multiple processors/threads. Many computing tasks exhibit a parallel nature and are hence suitable for parallel computing. Conventional parallel computing takes place using multi-core CPUs or via distributed, grid, high performance computers. The remarkably increased power of Graphics Processing Unit (GPU) in recent years offers a very attractive alternative, which can handle many demanding tasks by only harnessing local computing resource in low-cost computer platforms.

The most important development in GPUs in recent years has been the marked increase in their versatility. Their capabilities are now much more widely applicable and they have become used in many computational areas - this is known as General Purpose GPU programming (GPGPU). If the capacities of the GPU are harnessed properly, it is now the most powerful processor in a desktop computer and its development continues to outpace developments in CPUs. The GPU does, however, demand very specific skills to ensure that its potential is fully realised. These skills are still in relatively short supply and, for the moment, it certainly makes sense for an SME to outsource such work. This is the premise on which the current project is based.

The GPSME project has developed a toolkit which enables the SMEs to take advantage of current GPU capability without having the need for an in-depth understanding of GPUs and committing resources for manual CPU2GPU conversion, which has been the major stumbling

block in adopting a GPU approach. This allows the SMEs to gain great speed performance and help the advance of each application domain by allowing for advanced computing models with high complexity.

1.1 IME: Image Forgery Detection

IME is a technology research-backed company focusing on digital imaging forensics analysis. It is a leading innovator and technology provider for this field. The mission of the company is to support the widespread usage of digital photos and videos by increasing their credibility through deployment of the latest research results with exceptional reliability into the business world. Based on over 7 years of research, it provides a unique and patent-pending technology for verifying the trustworthiness of images, scanned documents and videos, detecting if digital manipulation has taken place, detecting and verifying the source of the image/video, and hence enhancing various industries by bringing a new revenue stream, fraud prevention method, competitive advantage and increasing trust between users. This technology has practical application in several fields, especially in insurance (i.e., during the processing of insurance claims and new contracts), in crime investigation and the legal processing of forensic evidence, in image search engines and in media production (e.g., newspapers and journals). The technology does not use any watermarks or signatures, which is regarded as a significant technical advantage.

IME provides forgery detection solutions via novel mathematical and computational algorithms, which detect the traces of tampering in a blind way and bring considerable improvements in the never–ending game between image forgery creators and image forgery detectors. Detecting suspicious and altered parts of the image or video is a time consuming task and the algorithms typically need high computational power. For example, when detecting cloning-based forgeries, where part of the image is copied and pasted into another part of the image, typically, with the aim to hide an important object or region of the image, we need to split the image into millions of small overlapping blocks and process and represent each block separately (e.g., using a Fourier-Transform-based representation). After that, a complex block-similarity examination is carried out. Given the typical size of an image, 35-45min of computational time is needed, on average, to process a single image. Since most newspapers and photo agencies receive thousands of photos per day, this computational time is not acceptable in practice.

Currently, the novelty and complexity of the detection methods employed puts IME in a very distinctive position compared to its direct local or global competitors, for example, the US based company Digimarc, which is the leading provider of invisible watermarking technology. However, this is a new area with fast evolution. The company is facing increasing threats from the growth of new technology and new competitors. The rapid advances in the field strongly suggest the importance of updating their existing product with latest techniques. Currently, due to speed limitations, the application of some advanced detection techniques developed by IME has been postponed to the market.

The GPU techniques support major performance improvement, and allow the company to use advanced detection techniques, and hence significantly increase its competitiveness. The GPSME toolkit can rapidly decrease the cost of converting the CPU programs into its GPU version. This main advantage results in a noticeably lower execution time for end-users, allowing IME to bring more cutting-edge image forensics technologies to real-life markets such as media, crime investigation and insurance, etc. This can lead to significant positive influence on the market position of the company

1.2 ROTA: Augmented Reality (AR) book

ROTA is a software entrepreneur company specialising in the field of Augmented Reality, supported by the latest technology of real time image processing and 3D computer graphics. ROTA has gained national and international awards through developing a number of AR

products using its unique technology. Its current product, Live Book**,** is the very first commercialized AR book in the world. The Canlı Kitap products, which are patented by Turkish National Patent Institute, have been released in Turkey before worldwide distribution. The project has been supported by TUBITAK within industry research and development assistance.

ROTA provides two sets of AR books for pre-school and secondary school children. Given the targeted market and customers, these products require real time performance at over 24 frames per second on **modest** platforms. Image processing speed is a bottleneck in the current products. ROTA needs the toolkit to enhance the current product – Live Book by performing real-time image processing. This can increase the effectiveness of the AR book. There are various AR books produced in Germany, France, South Korea, U.S.A. etc. Improving the quality of the products while maintaining the interaction speed constitutes one of the keys to winning the market.

1.3 SCS: Multimod Application Framework for VPH

Super Computing Solutions (SCS), which develops software solutions and services for computer aided medicine, is the primary designer and developer of the MAF(a software framework for the rapid development of computer-aided medicine application). Its mission is to translate research results in ICT and bioengineering into technology and services for computer-aided medicine and to promote their adoption in the clinical practice. MAF is widely used as the underlying foundation software in many VPH projects.

Local computing power based on GPUs facilitates research exploration by allowing the scientists to frequently experiment with computing models and parameters. It also helps to address security and legal problems in clinical practice since personal data from patients does not need to be accessed by external computing resources. However, these performance gains have not been widely noticed within VPH, and VPH is currently poorly placed to take advantage of GPU developments. Current VPH projects make little use of the GPU, employing them only for tasks that have traditionally been associated with computer graphics – visualization, collision detection, image processing, etc.

SCS plans to use the GPSME toolkit to convert many parallelisable applications within the current MAF into GPU execution to speed up the performance. Since MAF has been the foundational software for many VPH projects, this movement will create profound benefit to the VPH community. While many VPH applications are computationally demanding, some initial experiments have shown a speed improvement of several orders of magnitude due to the use of GPUs. GPUs now represent the most economical and effective route to high performance computing available to the average VPH practitioner.

1.4 ANS: Analysis of Eye Images for Clinical Applications

ANS is a research driven company specialising computer intelligent technology, service provision and software development. By using cutting-edge technology to develop innovative software applications, the company provides complete end-to-end intelligent software solutions to allow easy information access, data management, data analysis and visualization across a variety of platforms. The software applications are highly customizable to modern hand-held devices, such as mobile phones, tablets, etc. Medical image analysis is one of the key development interests of the company. Its current work on eye image processing, analysis, tracking and detection aims at convenient tools that facilitate the diagnosis of eye diseases.

The company has a number of competitors who are also working on eye image processing, including NewVision Fundus, Lickenbrock Technologies, faceLabTobii. The company endeavours to enhance the performance of its products by making use of advanced computer vision and image processing techniques.

The eye image analysis techniques adapted in the company have great potential in achieving parallelization. Therefore, harnessing the parallelised resources of the GPU provides a great

boost to the performance. By making use of the power of GPU, many of the current techniques can be scaled to higher complexities or dimensions to enhance the accuracy.

In summary, the GPSME toolkit allows the SME users to improve the speed performance of their products quickly and economically by automatically identifying the parallelizable sections of their programs and converting them into GPU implementations. Speed performance is the bottleneck of many SME related applications. It is expected that the SME participants to receive direct and immediate benefit from using the toolkit. This can not only significantly reduce the computational time of their current products, but also allow them to involve more advanced computational models to enhance the quality.

2. Main Objectives

The GPSME project has developed a toolkit which enables the SMEs to take advantage of current GPU capability without having the need for an in-depth understanding of GPUs and committing resources for manual CPU2GPU conversion, which has been the major stumbling block in adopting a GPU approach.

The toolkit is capable of carrying out highly automatic CPU2GPU source-to-source translation on moderately priced standard GPU cards and off-the-shelf GPU clusters. This allows the SMEs to gain great speed performance and help the advance of each application domain by allowing for advanced computing models with high complexity.

Technically, GPSME features adequate adaptation of automatic parallelization to modern GPU structures.

The specific objectives of GPSME and measurable outcomes are given below

I. Techniques allowing adequate adaptation of automatic parallelization to GPUs by taking full consideration of GPU compute architectures with a range of compute capabilities.

II. A toolkit that performs GPUification suitable to the SME application areas.

III. Scientific validation of the GPSME toolkit in the application areas.

IV. Enhanced speed and computing precision in each of the SME applications, including image forgery detection, AR book and MAF, eye image analysis, owing to the contribution of the toolkit.

V. Dissemination and exploitation of the results to a wider community.

II. FOREGROUND

1. Summary of Results

We identify three types of GPSME outputs.

- Products. These are tangible outputs in the form of new software technologies/systems/models that are exploitable into the future.

  Within the context of GPSME, this mainly refers to the C2GPU toolkit (which has been named as C2GPU toolkit), as well as the SME applications that are enhanced by the toolkit

- Publications. These can be scholarly articles by academic partners published in peer-reviewed journals or presented at conferences, or printed marketing materials created by SME partners to draw attention to new approaches or technologies.

  Some project results have been published in conferences. These have been reported in the Final Report: Impact, Dissemination and Exploitation, as well as in the dissemination plan D6.3.

- Experiences. These are less tangible but can still play an important part in the exploitation of project outcomes. They include: the experience and expertise gained by project partners and individuals in the management and undertaking of (trans-national) partnerships in a certain field, co-operation processes and methodologies; managerial lessons and know-how; and exchange of ideas and good practice through the establishment of networks, as well as events such as conferences, public awareness campaigns, seminars, debates and symposia.

These formed the basis on which the collection of exploitable foregrounds. The rest of this report will focus on the tangible output of the project, namely the C2GPU toolkit together with the SME applications that have been enhanced by the toolkit.

2. C2GPU Toolkit

2.1 Requirement Analysis of SMEs

We have analysed the general expectations of the SMEs on the C2GPU toolkit. The products of the SMEs involve a wide range of applicable areas including medical image processing, image forgery detection and augmented reality book. The user requirement analysis activities were led by a SME (IME) and these activities actively involved all the SMEs. Examples of the common requirements are (not limited to):

- Use the toolkit without need to have an in-depth understanding of GPUs
- Fully or semi-automatic CPU-to-GPU source translation
- Support C/C++ programming language
- Support either CUDA or OpenCL as output source.
- Good performance gain without accuracy loss
- Source code protection for security reasons
- Support error diagnosis

Our research shows that none of the existing CPU-to-GPU source translators is capable of fully satisfying the SMEs needs. The SMEs expect a toolkit that enables them to take

advantage of the latest GPU capability to effectively and economically speed up their products. This is evidenced by the analysis of the user requirements in the following 4 categories:

- **Acceleration**: The SMEs expect their applications to be accelerated significantly on moderate hardware platforms. They are interested in real time saving in their applications instead of a high speed up ratio of GPU over CPU. However, the existing CPU-to-GPU translators focus more on the speedup ratio of GPU over CPU rather than considering their practical performance. Their acceleration results are mostly achieved by running simple C code samples through advanced GPU hardware. Applying them to the SMEs applications cannot achieve desired performance gain. Therefore, we face the challenge from real-world applications instead of from simple sample code.
- **Applicability**: The SMEs expect the toolkit to have a wide applicability, allowing them to solve time-consuming tasks in various types of products. Among the existing CPU-to-GPU translators, algorithm skeleton based tools like Bones or Polyhedral have limited classes therefore they cannot support applications with complex or diverse loop types. The directive-based tools like OpenMP, HiCuda and PGI have a wide applicability owing to the flexible usage of the pragmas. However, learning these pragmas is a challenge to non-expert users. Thus, we need to achieve a balance between the applicability and directive complexity - the directives of the toolkit should be simple but support all types of algorithms skeletons and loop patterns.
- **Usability**: The SMEs have strong demands on the usability of the toolkits. Firstly, the input and output language are C/C++ and CUDA/OpenCL, respectively; Secondly, the SMEs suggested to use a web-service to achieve cross-platform (Windows and Linux) usage of the toolkit; Finally, a user management system with source code protection scheme is required for the web-service. Most of the existing tools use C-to-CUDA based command lines under Linux. The SMEs need a toolkit with better usability for non-expert GPU users.
- **Adaptability**: The feature of easy-to-learn is paramount to the SMEs. GPU technology and programming skills are hard to grasp. The existing CPU-to-GPU source translators still need users to study the usage of the directives. In fact, the simplicity of the directives is crucial to the adaptability of the toolkit. In this project, we have aimed to design simple, flexible and efficient directives.

2.2 Progress beyond the State of Art

In recent years, GPU computing is very effective in dealing with computationally intensive tasks. OpenCL and NVIDIA's CUDA are two most popular GPU parallel programming languages. The automatic CPU-to-GPU source translation techniques make GPU technology more accessible to users.

Currently, numerous CPU-to-GPU source parallelization translation tools have been developed for both academic and commercial purposes, which can be categorised into either fully-automatic or semi-automatic tools.

- For the fully-automatic tools, the targeted GPU codes are translated by a specific algorithm template mapping to CPU code. The algorithm class can be algorithm skeleton or polyhedral model, such as SkePU, Bones, PLUTO, R-stream and Par4All. Fully-automatic source translators have a good usability since they do not require users to have a basic GPU knowledge to identify parallel regions. However, their applicability is not good, since they are highly sensitive to the characteristics and data structure of the CPU algorithm template. This drawback significantly limits their wide acceptances by general users.
- The semi-automatic tools are also named as directive-based CPU-to-GPU translators, which produce target GPU code by adding annotations in the CPU source code. With the semi-automatic tools, users have to understand basic GPU knowledge and the usage of

the directives. Well-known semi-automatic tools include hiCUDA, MINT, PGI Accelerator. In comparison to the fully-automatic source translators, the semi-automatic tools have a better applicability in dealing with complex CPU algorithms. However, its usability is not good since users have to identify parallelizable regions and manage complex memory hierarchy by themselves. Also, the directives and unreadable output code in some tools bring difficulties to the users. Consequently, there is no existing CPU-to-GPU source translation tool reported in literature that provides an outstanding solution with good acceleration ability, wide applicability, good usability and adaptability.

Within the GPSME project, a directive-based online CPU-to-GPU toolkit is developed for the SME users to accelerate their applications. The main characteristics of the toolkit include:

- Several advanced pragmas have been defined to generate GPU kernel and to manage differentiated GPU memory hierarchy. These pragmas improve the applicability of the toolkit for the SMEs applications with complex algorithm skeletons.
- An annotation based programming model GPUSWO is presented to enable non-expert GPU users to flexibly and effectively implement CPU-to-GPU code translation for image filters applications.

  A SWO (Sliding Window Operation) logical based kernel generation pragmas scheme is introduced in the GPUSWO model. This scheme has enhanced usability and simplicity for SWOs applications than the existing CPU-to-GPU translators.
- Also, the C2GPU toolkit supports triangular loops and the optimization of multi-dimensional arrays. They significantly improve the acceleration ability of the toolkit.
- An easy-to-use web-based user interface is designed and integrated with the toolkit. Compared to the typical command line based CPU-to-GPU translators, this interface offers a much more friendly user interface.

These enhancements make the C2GPU toolkit more attractive to the SME users. A detailed performance evaluation of the GPUSWO model is given in Section 3. This evaluation shows that the GPUSWO model has an enhanced performance over other leading CPU-to-GPU translators on many applications.

2.3 C2GPU Toolkit Overview

The main system architecture of the C2GPU toolkit includes two components: GPSME web-interface and GPSME core library. The GPSME web-interface is a JavaScript command line.

2.3.1 Web Interface

Driven by the potential difficulties of having a local GPSME installation, and also to increase the visibility of the toolkit, we include access to the toolkit through a remote web server. The web server can be accessed at http://gp-sme.co.uk/web_face/.

This service facilitates the translation of the user source code, not needing anything related to the toolkit locally installed on the users' machines. A screenshot from the online translator is given in the image below.

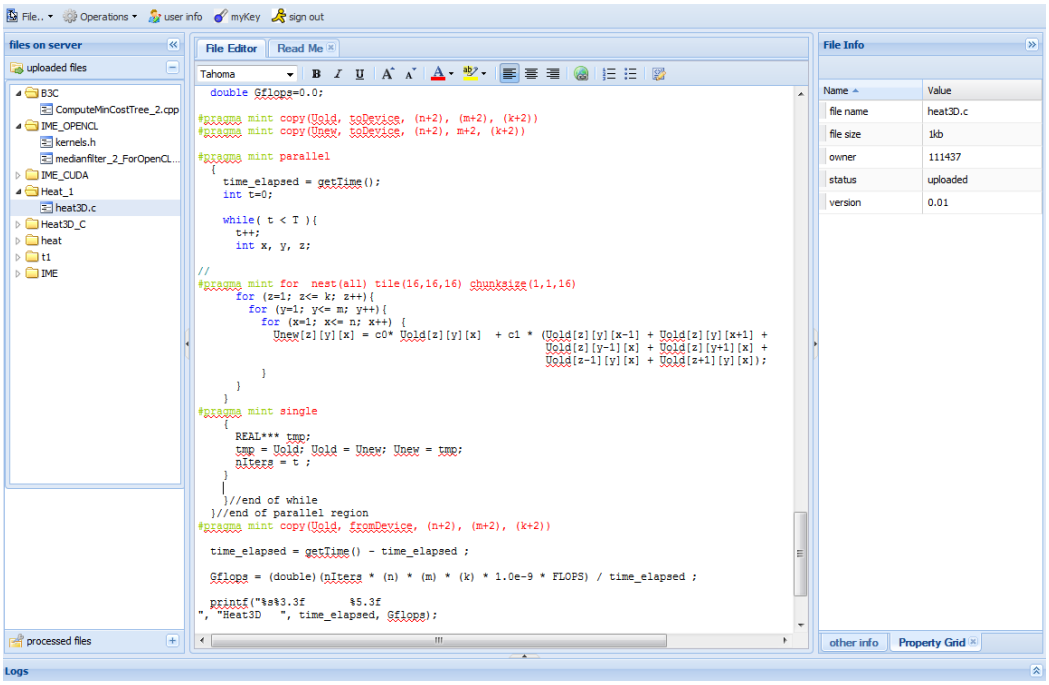


Figure 1: Web-based Interface of the C2GPU toolkit

The typical usage of the GPSME web application is as follows:

- The user creates and account with their details.
- The user uploads a C/C++ file
- The users upload the necessary header files.
- The user selects the desired output type.
- The user initiates the code translation process.
- It takes a few seconds for the C2GPU remote server to process the user files.
- The results can be retrieved under the 'Processed files' tab.

Many users will not be running the toolkit locally on a Linux machine but will instead be making use this remote web server. In this case you should keep the following points in mind in addition to those raised previously:

- If you have an external dependency then it must be installed on the remote webserver in addition to your local machine. You will need to contact you server administrator to get this set up.
- When uploading a C++ file for parallelisation you must also upload any of your own headers on which the C++ file is dependent. It is assumed that these belong in the same directory as the C++ source file, so avoid paths in your #include statements.

The webserver can also be installed locally on SME servers and act as a demo for the eventual customers.

2.3.2 System Architecture

The core structure and translation flow of the C2GPU toolkit is illustrated in Figure 2.

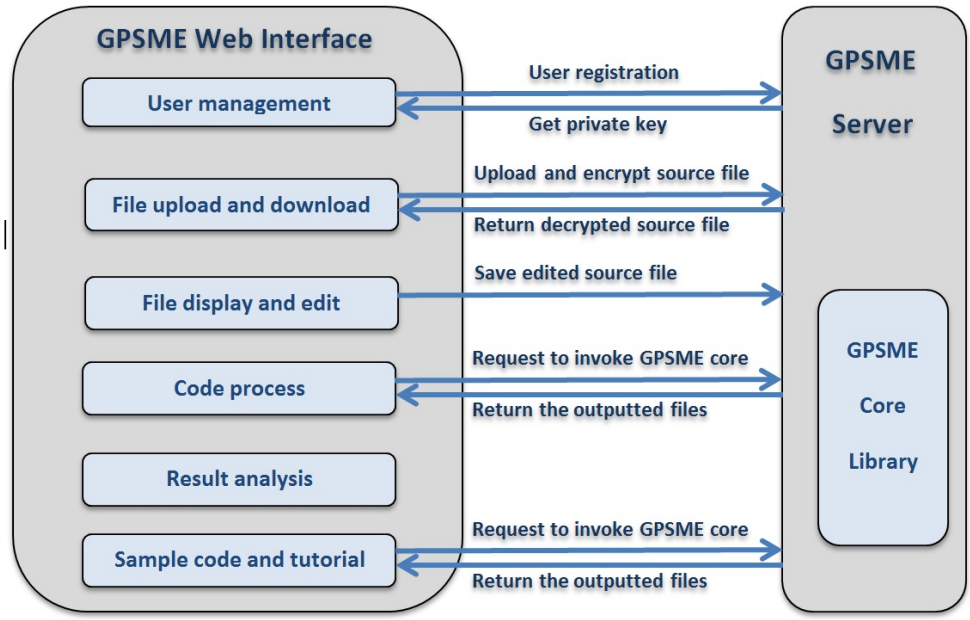


Figure 2. C2GPU System Structure

The toolkit is developed under Linux based on ROSE. Its structure is quite similar to MINT but with extended components. In each component, there are also some sub-components to manage different tasks. The system structure and translation flow is illustrated in Figure 2.

Table 1. Listing of C2GPU directives

	Directives	*Descriptions*
Basic pragma	**Parallel**	To identify a region generating kernel function
	Parallel region	To identify a parallel region containing parallel work
	For	To mark the succeeding for loop for GPU acceleration
	Single	To indicate serial regions in GPUSWO model
Memory Management	**CopyByTexture**	To create a CUDA texture on device, and bind or unbind with a 2D data
	CopyMalloc1DArray	To create a CUDA array on device, associating with CUDA texture on device.
	CopyMemcopy2D	To create a CUDA *cudamemcpy2D* function to copy a matrix between CPU and GPU memory
	CopyMemcopy2DToArray	To create a CUDA function *cudaMemcpy2DToArray* to copy data between CPU and GPU memory.
	CopyBindTexture	To bind the created texture memory to CUDA global array.
	Copy2DArrayTo1DArray	To covert the array with different dimensions on CPU memory buffer.
Kernel Generation	**Initialisation**	To define a one dimensional array for storing the data in a sliding window
	Transfer	To transform the code of putting the data in a sliding window into a local variable within "**For**" loop

	Remain	To transform the operations on sliding window from CPU algorithm to GPU kernel.
	Assign	To assign the new data to the relevant GPU buffer with correct index.

List 1 Translation of C2GPU Memory Creation

```
1:  #pragma parallel region { // Loop start for whole image
2:  ……..
3:  #pragma copy2DArrayTo1DArray(W, toHost, I, J, W_1D,
    2DTo1D) // covert data
4:  int   i_1; int   j_1;
6:  for (i_1 = 0; i_1 < J; ++i_1)
7:      for (j_1 = 0; j_1 < I; ++j_1) {
8:          W_1D[i_1*I+j_1] = W[i_1][j_1]; }

13: #pragma copyMemcopy2D(W, HosttoDevice, I,
    J, pitch1, W_1D)// Transferring Data CPU-GPU
14: cudaMemcpy2D(d_dev_5_W,sizeof(float ) * I,
    W_1D,pitch123,sizeof(float ) * I, J,
    cudaMemcpyHostToDevice);

15: #pragma copyMemcopy2DToArray(W, DevicetoDevice,
    ROI_w, ROI_h, pitch1)
16: cudaMemcpy2DToArray(array_dev_4_W,0,0,
    d_dev_5_W,pitch123,sizeof(float ) * ROI_w, ROI_h,
    cudaMemcpyDeviceToDevice);

17: #pragma copyBindTexture(W, DevicetoDevice,
    W, float, Bind)// Bind CUDA Array to Texture
18: cudaChannelFormatDesc desc_3;
19: desc_3 = cudaCreateChannelDesc<float>();
20: cudaBindTextureToArray(&tex_dev_4_W,
    array_dev_4_W,&desc_3);

17: #pragma parallel region {
    // Loop start for sliding window ,  calling kernel
18: ……..
19: … MedTex_2( float *d_out,  unsigned int Pitch, int w, int
                h)……..
20: }
21: #pragma copyMemcopy2D(imDenoised_GPU, DevicetoHost,
I,
    J, pitch1, W_1D)// Transferring Data CPU-GPU
```

```
22:  cudaMemcpy2D(W_1D,sizeof(float ) * I,
     d_dev_6_imDenoised_GPU,pitch123,sizeof(float ) * I, J,
     cudaMemcpyDeviceToHost);

23: #pragma copy2DArrayTo1DArray(W_1D, toHost,
    I, J, W, 1DTo2D) // covert data
24:      int i_2; int j_2;
25:     for (i_2 = 0; i_2 < J; ++i_2)
26:         for (j_2 = 0; j_2 < I; ++j_2) {
27:             imDenoised_GPU[i_2][j_2] =W_1D[i_2*I+j_2]; }
```

List 2 Translation of C2GPU Data Transfer

```
1:  // Initialisation and set GPU memory.
2:  #pragma copyByTexture(D, toDevice, N,  M, Bind, char)
    // set CUDA Texture memory for whole image.
3:  cudaChannelFormatDesc desc_1;
4:  desc_1 = cudaCreateChannelDesc<unsigned char>();
5:  tex_dev_3_D.normalized = false;
6:  tex_dev_3_D.addressMode[0] = cudaAddressModeClamp;
7:  tex_dev_3_D.addressMode[1] = cudaAddressModeClamp;
8:  tex_dev_3_D.filterMode = cudaFilterModePoint;
9:  cudaMallocArray(&array_dev_3_D,&desc_1,N,M);
10: cudaMemcpyToArray(array_dev_3_D,0,0,((char *)D),sizeof(char
) * N * M,
    cudaMemcpyHostToDevice);
11: cudaBindTextureToArray(tex_dev_3_D, array_dev_3_D);
13:  #pragma copyByTexture(W, toDevice, I, J, NoBind, float)
     // set CUDA Texture memory for sliding window region
14:  cudaChannelFormatDesc desc_2;
15:  desc_2 = cudaCreateChannelDesc<float>();
16:  tex_dev_4_W.normalized = false;
17:  tex_dev_4_W.addressMode[0] = cudaAddressModeClamp;
18:  tex_dev_4_W.addressMode[1] = cudaAddressModeClamp;
19:  tex_dev_4_W.filterMode = cudaFilterModePoint;
20:  cudaMallocArray(&array_dev_4_W,&desc_2,I,J);

22: #pragma copyMalloc1DArray(W, toDevice, I, J,  pitch1)
23: cudaMallocPitch(((void **)(&d_dev_5_W)),&::pitch123,I *
sizeof(float ), J);

25:  #pragma copyMalloc1DArray(imDenoised_GPU, toDevice, I,
J, pitch1, InKernel)
26: cudaMallocPitch(((void
```

```
**)(&d_dev_6_imDenoised_GPU)),&::pitch123,I *
    sizeof(float ), J);


27:  #pragma parallel region { // Loop start for whole image
28:  ……..
30:  }
```

List 3 Translation of C2GPU Kernel Generation

CPU Code	GPU Kernel
1:	***__global__ void kernel(int Pitch, float *d_out, int w, int h)***{
2: ***#pragma parallel {***	// index caculation
3: ……	int x = blockIdx.x * blockDim.x + threadIdx.x;
4: ……	int y = blockIdx.y * blockDim.y + threadIdx.y;
3: ***#pragma single initialisation{***	int i = 0;
4: float v[9] = {0,0,0,0,0,0,0,0,0}; }	float v[9] = {0,0,0,0,0,0,0,0,0};
5: ***#pragma for nest(2) tile(16,16)***	// data transfering
6: for (i = 1; i <= height ; i++)	for (int xx = x - 1; xx <= x + 1; xx++)
7: for(j = 1; j <= width ; j++){	for (int yy = y - 1; yy <= y + 1; yy++) {
8: ***#pragma single transfer***{	if (0 <= xx && xx < w && 0 <= yy && yy < h) // boundaries
9: v[0] = Image [i-1][j-1] ;	v[i++] = tex2D(tex_CFA_2, 0.5f+(float) x, 0.5f+(float) y);}
10: v[1] = Image [i-1][j] ;	// directly copy from CPU code
11: ……..	for (m = 0 ; m < 9 ; m++)
12: v[8] = Image [i+1][j+1]; }	for (t = m+1; t < 9; t++) {
13: ***#pragma single remain***{	if(v[m] > v[t]) {
14: for (m = 0 ; m < 9 ; m++)	tmp = v[m];
15: for (t = m+1; t < 9; t++) {	v[m] = v[t];
16: if(v[m] > v[t]) {	v[t] = tmp ; }}
17: tmp = v[m];	// pick the middle one
18: v[m] = v[t];	float* row = (float*)((char*)d_out + y * Pitch);
19: v[t] = tmp ; }} }	row[x] = v[4];
20: ***#pragma single assign*** {	}
21: Image[i][j] = v[4] ; }	
}	

2.3.3 GPSME Directives and Translation

The details of the C2GPU pragmas are presented here, as seen in Table 1. The basic pragma is similar to the pragma in MINT. The only difference is that ***Parallel Region*** indicates the start of a parallel region containing parallel work. These regions within the block of this pragma will be accelerated.

Parallel indicates the start of a region generating kernel function, which normally contains some "**For**" loops. We differentiate these two pragmas by using a memory management pragma "***CopyByTexture***". The parallel node behind "***CopyByTexture***" is recognized as a parallel region without generating a kernel function.

For memory management pragmas, we extend "**Copy**" pragmas regarding the memory creation procedure, data covert and data transfer. Memory creation includes two copy based pragmas :"***CopyByTexture***" and "***CopyMalloc1DArray***". The pragma "***CopyByTexture***" creates a CUDA texture on device, and binds or unbinds with a 2D data. This pragma is used to allocate device memory for the input high-resolution image. This normally occurs in the step of initiation.

The pragma "***CopyMalloc1DArray***" creates a CUDA array on device, associating with a CUDA texture on device. Data transfer is responsible for transferring data between CPU and GPU. It includes two copy based pragmas: "***CopyMemcopy2D***" and "***CopyMemcopy2DToArray***". The first one creates a CUDA function *cudamemcpy2D* to copy a matrix between CPU and GPU memory. The second one creates another CUDA function *cudaMemcpy2DToArray* to copy data between CPU and GPU memory.

Also, a pragma "***CopyBindTexture"*** is defined to bind the created texture memory to CUDA global array. The clauses of these pragmas include the name, the size of variable and the means of data transferring. Data conversion is used to convert the array with different dimensions on CPU memory buffer. Image processing applications normally use OpenCV libraries so the dimension of data has to be ordinarily converted from 2D array to 1D array for GPU use. One copy based pragmas are defined as "***Copy2DArrayTo1DArray***". The clause of this pragma includes the type of data conversion, such as 1DTo2D, 2DTo1D.

List 1-3 show the translation of GPSME memory creation directives, the translation of GPSME data transferring and converting directives and the translation of GPSME kernel generation directives, respectively.

3. Evaluation

3.1 Acceleration

Acceleration is the most important performance indicator of the C2GPU toolkit. In order to evaluate this, we have compared the performance between original CPU code, revised CPU code, machine-generated GPU code and manually-generated GPU code.

The original CPU codes were selected from the applications of the SME partners. They were revised by the users in order to be processed by the C2GPU toolkit. The toolkit generated the machine-generated GPU codes. For comparison purpose, we have also performed CPU to GPU code conversion manually.

3.1.1 SCS

The application from SCS is to extract a centerline from a given 3D model. The codes from SCS are based on C++, which calls some VTK functions for centerline extraction. The performance gain by the use of the toolkit is shown in Table 2.

Table 2 Acceleration performance of B3C centerline extraction

	Small 3D model			Big 3D model		
	CPU	**GPU**	**Speedup**	**CPU**	**GPU**	**Speedup**
Manual	921ms	202ms	4.56	28k ms	1.1k ms	24.38
Toolkit	165ms	78ms	2.11	1408k ms	448 ms	3.14

Table 2 shows that on average the C2GPU toolkit can accelerate the application from SCS up to 2-3 times.

3.1.2 IME

Table 3 Acceleration performance of IME camera fingerprint measurement

	CPUs	GPUs	Times	Accuracy	Details
GTX 690	28.24	8.283	3.40	0.0	Whole Application (1 image)
GT 540	84.45	19.66	4.29	0.0	Whole Application (1 image)
GTX 690	16.56	3.912	4.23	0.0	Kernel Region (1 image)
GT 540	51.37	8.321	6.17	0.0	Kernel Region (1 image)
GTX 690	1118	288.7	3.78	0.0	Whole Application (39 image)
GT 540	3047	707	4.31	0.0	Whole Application (39 image)
GTX 690	663.7	153.6	4.35	0.0	Kernel Region (39 image)
GT 540	2023	325.8	6.21	0.0	Kernel Region (39 image)

The application from IME is to produce a camera fingerprint by applying de-noising methods to a set of images that are known to come from a given camera. The sample codes from IME are based on C++, which aim at implementing a 3×3 median filter de-noising to process their image data. The image resolution is 3648 × 2736. The number of images is 39. The algorithm split the images into a number Region of Interest (ROI), which can be processed in parallel. Table 3 shows the performance gain by the use of the C2GPU toolkit.

Firstly, it shows that the accuracy of the IME application delivered by the GPU implementation and the CPU implementation is exactly same, which means that the machine generated GPU implementation does not affect the accuracy of the camera fingerprint in the IME application. Secondly, it appears that on average the C2GPU toolkit accelerates the performance up to 3-4 times. If only considering the acceleration of the kernel region, the speedup performance can achieve up to 6 times faster than the original CPU application. This proves that the GPU kernel implementation is certainly capable of speeding up the CPU code in the parallel regions.

3.1.3 ROTA

The application from ROTA uses ASIFT algorithm for feature extraction in augmented reality applications. ROTA has successfully evaluated the ASIFT implementations on their own dataset, as shown in Figure 3. The matching accuracy of the GPU implementation is almost the same as the original CPU implementation. The performance evaluation results are shown in Table 4.

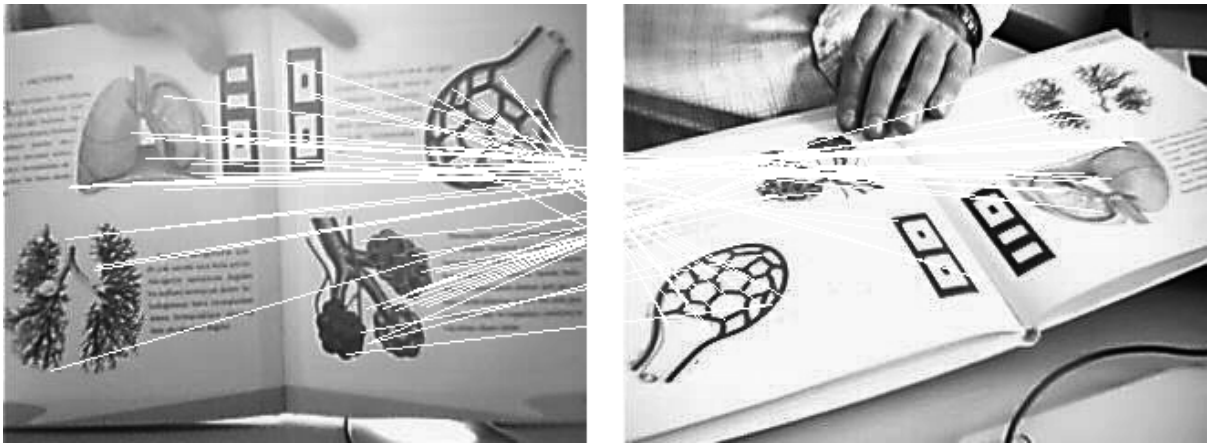


Figure 3 ASIFT evaluation from RotaSoft

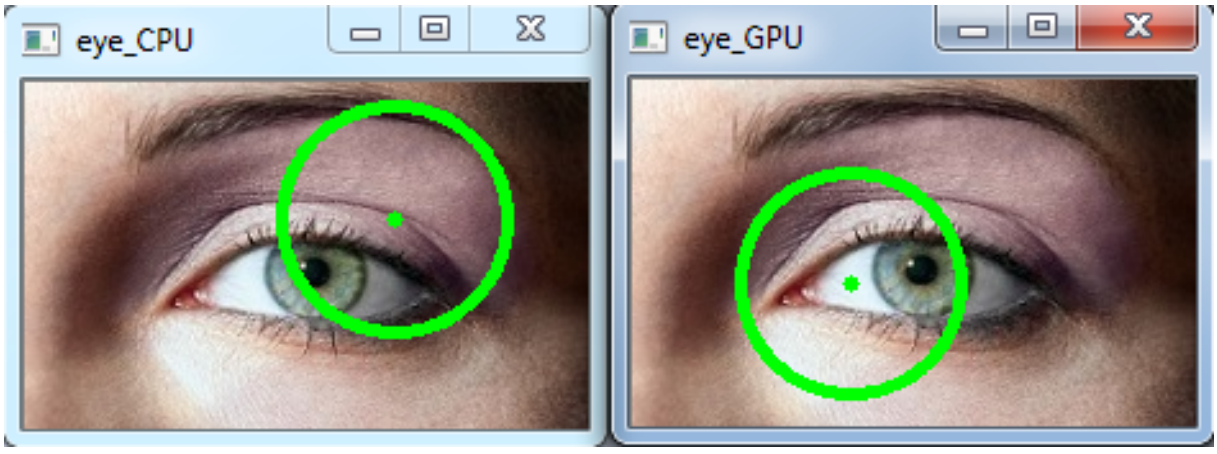
Table 4 Acceleration performance of ASIFT from RotaSoft

	Core i3@2.1GHz+GT520M (time in seconds)	**Core i7@3.4GHz+GTX680 (time in seconds)**
Original	69.5	25.9
OpenMP	25.7	6.7
Manual GPU	12.5	1.9
Toolkit	14.6	3.2

It appears that on average the C2GPU toolkit accelerates the whole application performance up to 6x times for a low-grade system, and up to 13.6x for a high performance system.

3.1.4 ANS

The application from ANS is to recognize iris information from a high-resolution video by applying some image processing methods. The iris detection algorithm is a Hough transformation based method, which aims to find out the circle of the iris from a thresholded eye image, as shown in Figure 4.



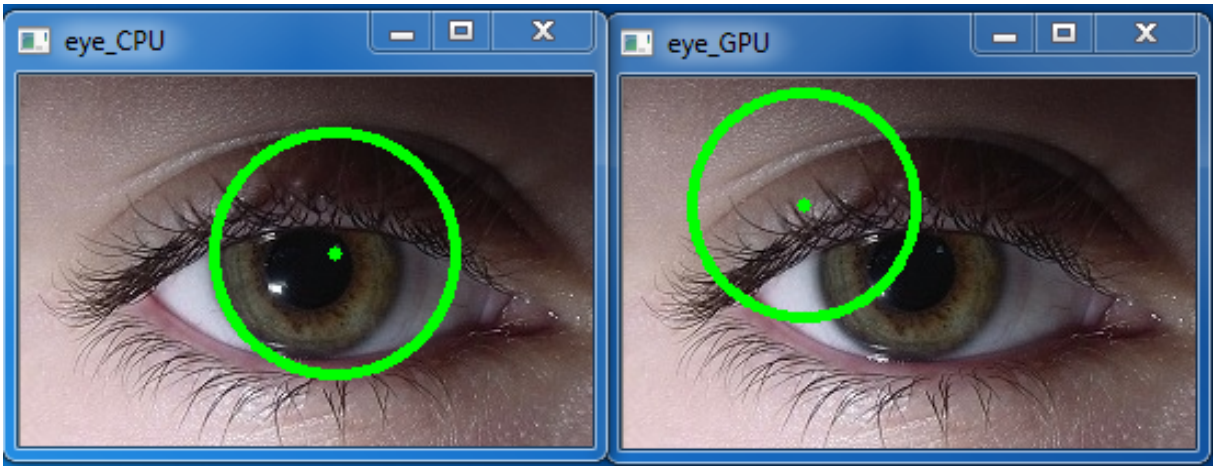


Figure 4 Iris Recognition by hough transformation from ANS.

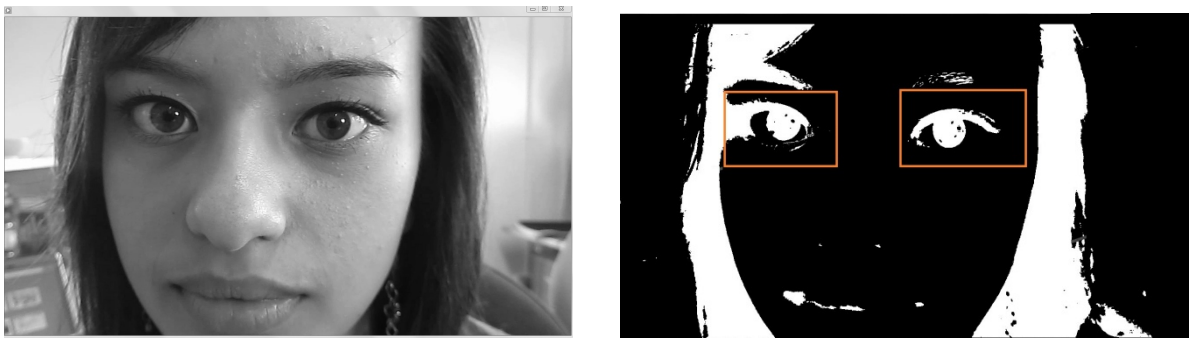
After using the C2GPU toolkit, the performance evaluation results are shown in Table 5

Table 5 Acceleration performance of Hough Transformation from ANS

	CPU	GPU	Speedup
Image 1 (261 × 168)	3193ms	2325ms	1.37
Image 2 (220 × 135)	1117ms	1636ms	0.68
Image 3 (320 × 240)	4521ms	2345ms	1.92
Image 4 (640 × 480)	89602ms	××××	×××

It appears that on average the C2GPU toolkit accelerates the iris recognition application up to 1-2x times. Normally, images with high resolutions can achieve more speedup than images with low resolutions. However, the bottleneck is that limited register memory on GPU cannot store the transformed 3D array from high resolution images.

The second application from ANS is to use morphological filter to detect the eye's region position from videos. While OpenCV provides some functions to detect the position of eye regions, the performance is limited by a variety of reasons, such as lighting, head position, noise, etc. Therefore, ANS has developed own morphological filter based algorithms to segment the eye regions from videos. The morphological filter replies on the repeated use of dilation and erosion operations on a binary image. However, the repeated use of dilation and erosion is a very time-consuming task, particularly for high resolution images with large size of window kernel. The C2GPU toolkit is capable of successfully accelerating the performance of morphological filters. The results are as shown in Figure 5 and Table 6.



a) Original image b) CPU processed image

Figure 5 Eye detection by morphological filtering from ANS.

Table 6 Acceleration performance of 9 * 9 morphological filtering from AnSmart

CPU i7-2740 vs GPU MT 540	*CPU*	*GPU*	*Speedup*	*Times of Dilation*	*Times of Erosion*
Image 1 (1285 × 751)	452ms	285ms	1.58	1	1
Image 2 (1279 × 721)	560ms	322ms	1.75	1	1
Image 3 (640 × 480)	324ms	278ms	1.23	1	1
Image 1 (1285 × 751)	2458ms	847ms	2.91	2	2
Image 2 (1279 × 721)	2870ms	780ms	3.72	2	2
Image 3 (640 × 480)	1232ms	458ms	2.68	2	2

From Table 6, it appears that the C2GPU toolkit can effectively speed up the morphological filtering for the ANS eye detection algorithm up to 3x. Increasing the number of iterations for the dilation and erosion will lead to more acceleration in the performance of the C2GPU toolkit. Another issue is that the window size of the running kernel could affect the acceleration performance. 9x9 morphological kernel is used in this case. If the window size increases, the speedup ratio can be enhanced significantly. Oppositely, for small window size kernel, the performance of the C2GPU toolkit is not obvious.

To sum up, the acceleration performance of the C2GPU toolkit is generally accepted by all the SME partners. On average their applications can speed up 3-4x times, even up to over 10 times on a high performance GPU system. This overall acceleration is very good and solves a lot of problem in the practical applications.

3.2 Applicability

Applicability is another important factor of the C2GPU toolkit. We expect the toolkit to be applicable to a wide range of industrial applications. To this end, the SME partners have evaluated the C2GPU toolkit in the area of Document Segmentation application and Sliding Window Image Filtering.

3.2.1 Document Segmentation

Large-scale document digitalisation is a popular topic for many libraries and museums in recent years. It involves a significant amount of document layout analysis, region segmentation and text line segmentation. For large scale document digitalisation, this is a time-consuming task due to the amount of the newspapers, magazines and other documents required to be scanned at high-resolution on daily basis. The SME partners have used dilations and erosions algorithms to process some sample newspaper documents images from IMPACT, which is the most successful large-scale document digitalisation project in the last 10 years. The processed newspaper document images are set to be evaluated by a region segmentation method. The image resolution is 3595 × 5194. The C2GPU toolkit is capable of processing C++ code, and the results are shown in Table 7.

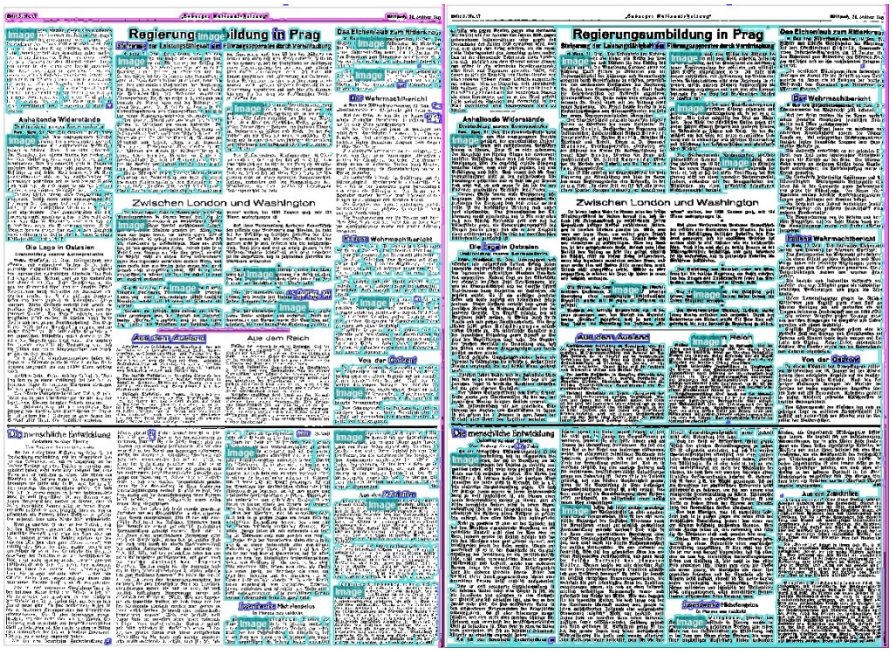
Table 7 Acceleration performance of Document Segmentation

	CPU (s)	**GPU (s)**	**Times**	**Details**
GTX 690	0.268	1.003	0.267	3 × 3 dilation operator
GT 540				3 × 3 dilation operator
GTX 690	1.195	1.289	0.927	5 × 5 dilation operator
GT 540				5 × 5 dilation operator
GTX 690	3.694	1.262	2.927	9 × 9 dilation operator
GT 540				9 × 9 dilation operator

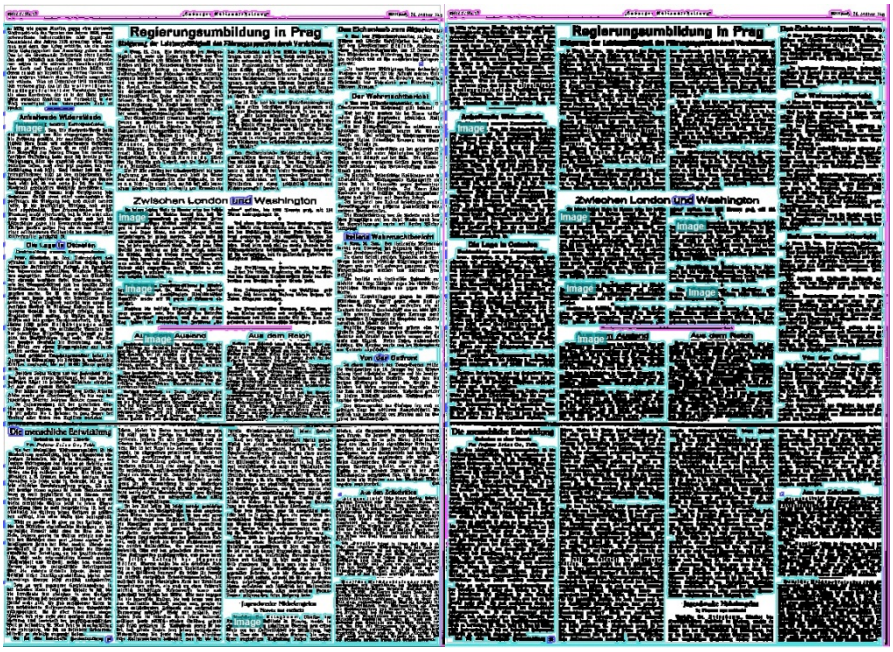
GTX 690	0.247	1.032	0.239	3 × 3 erosion operator
GT 540				3 × 3 erosion operator
GTX 690	1.107	1.03	1.07	5 × 5 erosion operator
GT 540				5 × 5 erosion operator
GTX 690	3.354	1.17	2.867	9 × 9 erosion operator
GT 540				9 × 9 erosion operator

The evaluation results in Table 7 show that for dilation or erosion operator over 5 × 5 sub-windows, the C2GPU toolkit can speed up the application performance up to 1-3 times. For dilation or erosion operator less than 5 × 5 sub-windows, the GPU performance is even slower than the CPU performance. This phenomenon implies that if the dilation or erosion operator is less than 5 × 5 sub-windows, the benefit of GPU acceleration on kernel is cancelled out by the introduced overheads (e.g. data transmission between CPU and GPU) and by other commitment introduced on the CPU side, e.g. the extra CPU code employed for the purpose of processing the pragmas, etc.

Figure 6 illustrates the evaluation results of using a region segmentation method to extract the regions of the sample newspaper images. This region segmentation method is based on a hybrid way of erosion and dilation. In the original newspaper image, a large amount of text regions are missed due to the low density of characters. By using 3 × 3 dilation operators, most of the text regions are segmented but there are two pieces of text region in the middle of document are missed. By using 5 × 5 dilation operators, the performance is improved but there is still one piece of text region in the middle of document missed. By using 9 × 9 dilation operators, all text regions in the newspaper are successfully segmented.



a) Original newspaper image b) 3 × 3 dilation operator



c) 5 × 5 dilation operator b) 9 × 9 dilation operator

Figure 6 Document Region Segmentation Results of GPUtisalition toolkit

Therefore, it is concluded that 9 × 9 dilation operators can give the best result for newspaper document image segmentation. Considering the results in Table 7, C2GPU is capable of speeding up the application performance up to 3 times.

3.2.2 Sliding Window Image Filter

Sliding Window Operation is a very popular technique in image processing. Typically, Sliding Window Operation repeatedly applies an image filter to a predefined small size sub-window that is shifted across a target image. This operation involves high computing complexity if image filter contains many loops or iterations with high floating-point arithmetic intensity. This particular structure fits very well with the GPU date parallel programming model.

The SME partners have implemented several statistic measurements based image filter algorithms. Ten typical SWO image operators as benchmarks were selected to apply in a high-resolution image by using different size of sliding windows. The benchmarks are listed in Table 8. The size of the evaluated sliding window is respectively given as 3×3, 5×5, 7×7, 9×9. The resolution of the evaluated image is 3325×4765. The baseline is the performance of the original CPU code running on conventional hardware without using multi-threads. It compares the speedup ratio of C2GPU-generated CUDA, MINT-generated CUDA and OpenMP over this baseline. The evaluation platform consists of an Intel Core i7-2670QM CPU and NVIDIA GeForce GT 540M, and another system consisting of an Intel Core i7-3770K CPU and NVIDIA GeForce GTX 690. Both the CPU and the GPU are programmed using NVIDIA GPU SDK version 4.1. OpenMP programs were compiled using Visual Studio 2008. All computations were run in double precision.

Table 8 Benchmarks for Evaluating the C2GPU model

Directives	*Descriptions*
MinFilter	Get maximum value among all elements
MaxFilter	Get minimum value among all elements
MedianFilter	Get middle value after all elements are sorted numerially
MidPointFilter	Get an average value of maximum and minimum among all elements
AlphaTrimmedMeanFilter	Disgard the most atypical elements and calcuate mean value using the rest of them
StandardDeviationFilter	Used to emphasize the local variability in an image
ModeFilter	Used to emphasize the local variability in an image
MeanFilter	Used to emphasize the local variability in an image
Erosion	Used to emphasize the local variability in an image
Dilation	Used to emphasize the local variability in an image

The primary goal is to explore the acceleration performance of the C2GPU tookit over the CPU code execution baseline. For the simplicity, here we only demonstrate the speedup ratio of the above ten benchmarks with sliding window 5×5. The results are shown in Figure 7.

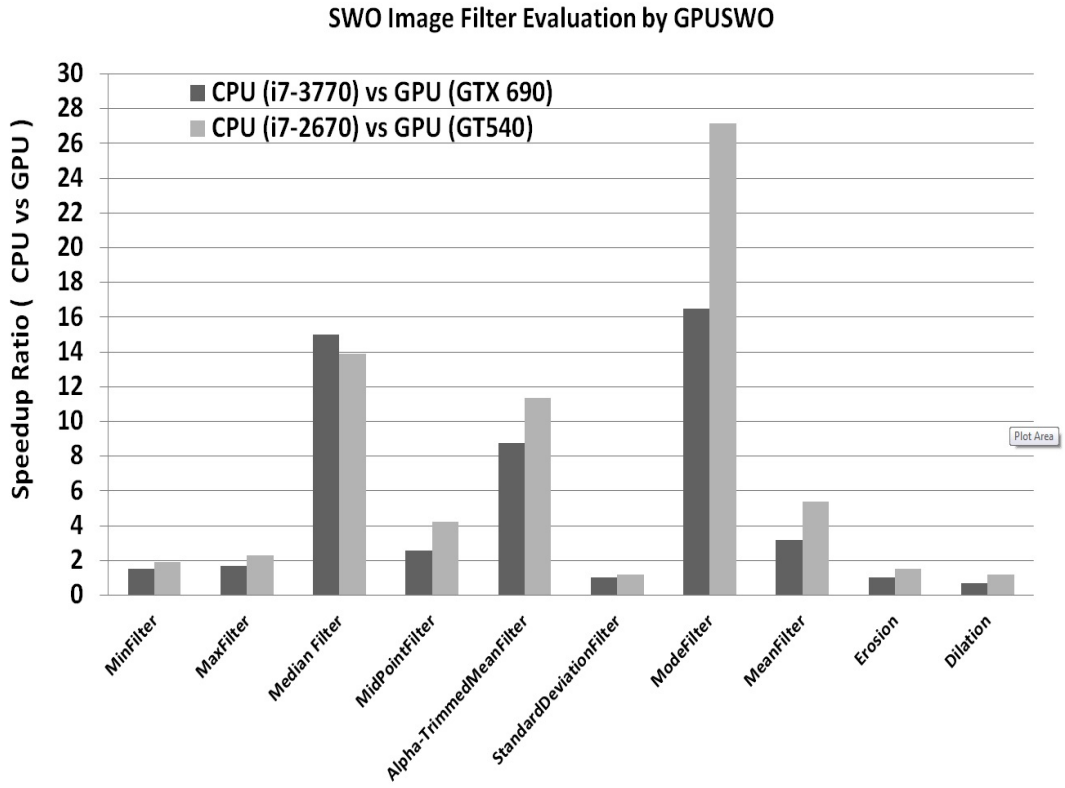


Figure 7. Performance evaluations of the SWO Image Filters

Figure 7 shows the acceleration performance of each benchmark over the baseline CPU execution under two different platforms. On both platforms, except dilation and standard deviation filter, the speedup ratios of the rest eight benchmarks are over one. Mean Filter and Mid-Point Filter can be accelerated by the C2GPU toolkit up to 2-5 times. The performance of the benchmarks with highly intensive computation (Median filter, Alpha-Trimmed Mean Filter and Mode Filter) is particularly impressive by reaching up to 10-20 speedup ratios. This result reflects that the C2GPU toolkit can effectively speedup the performance of different types of SWO image filters, particularly effective on some intensive computation SWO image filters. But for image filters with low computation, the effectiveness of the C2GPU toolkit is not obvious. This is because the running time of the parallel regions in these filters takes a relatively low proportion of the entire running time of the CPU algorithms. The speedup gained by the

C2GPU toolkit is diluted so that the overall running time of the CPU code and the GPU code seems to be equal.

We have also evaluate different factors that influence the performance. The processing time of the SWO image processing algorithms is decided by three factors:

- the size of the target image,
- the size of the sliding window
- the computation complexity of the image filters.

The first factor should have no impact on the C2GPU-generated GPU code. So we evaluate the impact of the rest two factors on the acceleration performance of the C2GPU toolkit.

With respect to the influence of the size of the sliding window, we conclude the speedup ratio of C2GPU-generated GPU code over CPU baseline with 3×3, 5×5, 7×7, 9×9. The result is shown in Figure 8. Figure 8 demonstrates that as the increasing sliding window size, the speedup ratios of each benchmark of the C2GPU-generated GPU code over the CPU baseline are significantly enhanced. The performance of filter with intensive computation is continuously accelerated up to 20 times by the C2GPU toolkit. The filters with low computation kernel also have a higher speedup ratio at larger sliding window size. This phenomenon implies that the C2GPU toolkit is well suitable to deal with large size sliding window based image processing applications. However, it is important to note that both Median Filter and Alpha-Trimmed Mean Filter do not have the figures on sliding window size 9×9. This is because the C2GPU toolkit does not provide the optimization of using shared memory in the kernel. This causes the problem of memory shortage if the filters have both large sliding windows and intensive computation in the kernel. This is a limitation of the C2GPU toolkit.

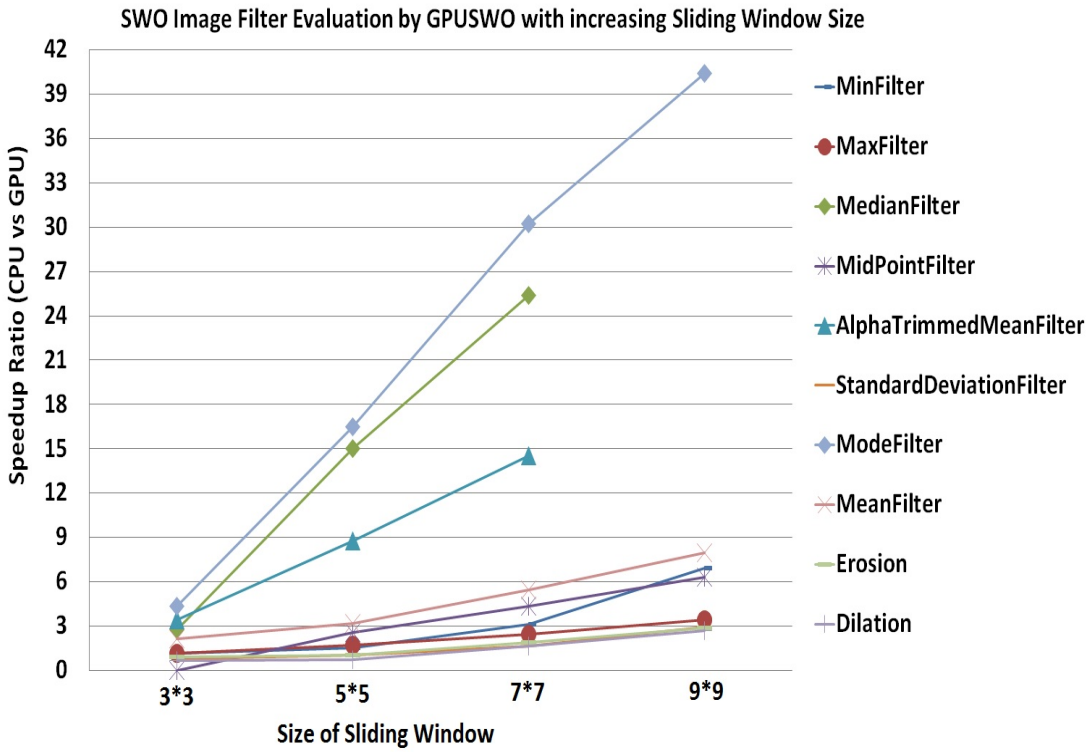


Figure 8. The Impact of the Sliding Window Size on Speedup Ratio

To evaluate the impact of kernel computation complexity, the CPU code of ten benchmarks using 5×5 sliding window were measured. In these kernels, the statements of “for” and “if” are identified to measure the kernel computation complexity. For instance, if there are two “if”

statements within a “for” loop from 1 to 50, the kernel computation complexity is 50 by 2. The impact of kernel computation complexity on speedup ratio is shown in Figure 9.

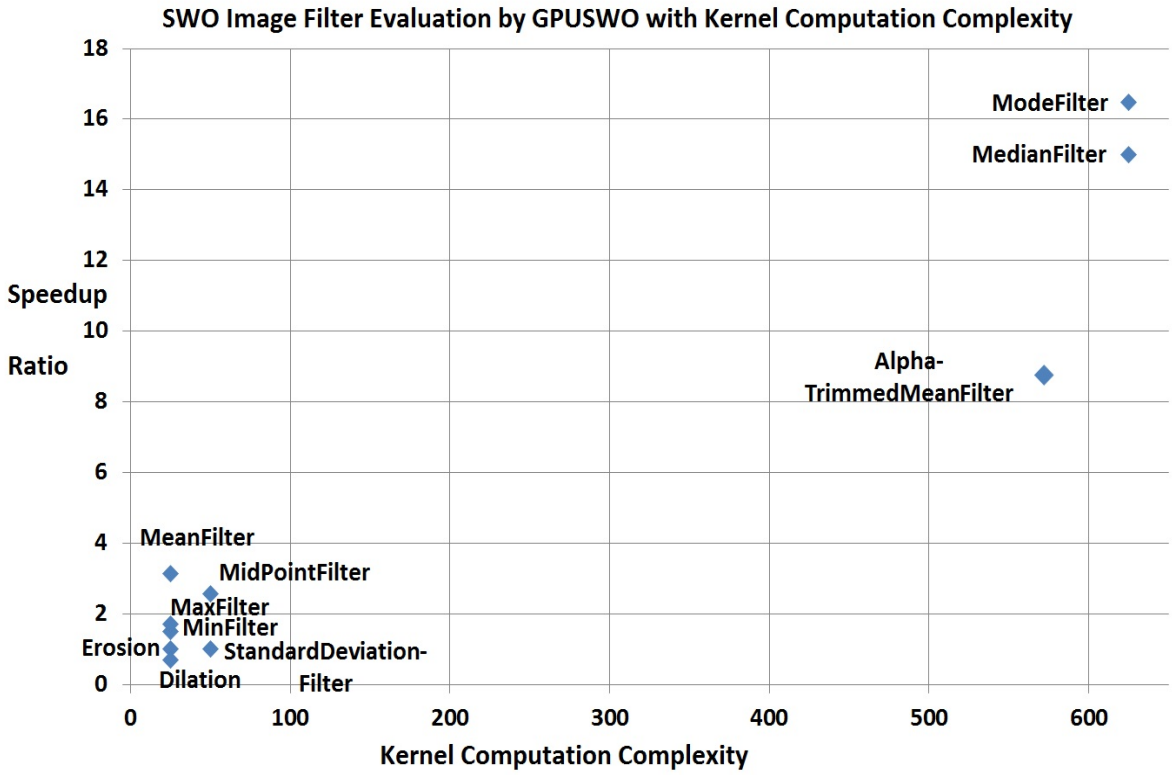


Figure 9 The impact of kernel computation complexity on speed ratio

From Figure 9, it appears that the positions of the benchmarks are distributed at the left-bottom and right-top corners of the diagram. With a given size of sliding window, the acceleration ratio dramatically increases as the growing complexity of the kernel computation. This trend reflects that the C2GPU toolkit is well suitable to process SWO image filters with intensive kernel computation.

According to the above evaluation findings, we can see that the C2GPU toolkit is capable of effectively accelerating the performance of most of typical SWO image filters. The applicability of the C2GPU toolkit is particularly suitable for large-scale image processing applications with the use of highly intensive computation filters. The major bottleneck of the C2GPU toolkit is that the limited size of global memory on some GPU devices cannot fully support the application using both large size of sliding window and intensive computation filter.

3.2.3 Blur moment invariants

Blur moment invariants are widely used in digital image processing. They are functional invariant with respect to blur. In the SME applications, blur invariants of order p+q is generated using the recursive equation shown below.

$$B(p,q) = \mu_{pq} - \alpha\mu_{qp} - \frac{1}{\mu_{00}} = \sum_{n=0}^{K}\sum_{i=m_1}^{m_{2i}} \binom{p}{t-2i}\binom{q}{2j} B(p-t+2i, q-2i)\mu_{t-2i,2i},$$

where

$$K = [(p+q-4)/2],$$

$$t = 2(K-n+1),$$

$$m_1 = \max(0, [(t-p+1)/2]),$$

$$m_2 = \min(t/2, [q/2]),$$

$$\alpha = 1 \Leftrightarrow p \wedge q \text{ are even},$$

$$\alpha = 0 \Leftrightarrow p \vee q \text{ are odd}.$$

These blur invariants are employed by IME to identify near-duplicated regions in a digital image. This is carried out in a few main steps:

1. Tiling the image with overlapping blocks,
2. Moment blur invariants representation of the overlapped blocks,
3. Principal component transformation,
4. K-d tree representation,
5. Blocks and neighbors analyses (matching),
6. Near-duplication map creation.

The image is tiled by overlapping blocks of R × R pixels. Blocks slide by one pixel along the image from the upper left corner right and down to the lower right corner. The total number of overlapped blocks for an image of M × N pixels is (M – R + 1) × (N – R + 1). For instance, an image with the size of 2000 × 2000 with blocks of size 16 × 16 will product 3.940.225 overlapped blocks. Moment blur invariants representation of each block is computed separately which makes the run-time of the method too expensive. Thus, this is the part in we can accelerate using the C2GPU toolkit. The experimental results are shown in Table 9

Table 9 Evaluating the C2GPU in blur moment invariant

	CPU	*CPU without OpenCV*	*Manual GPU*	*Auto GPU by C2GPU toolkit*	*Speedup Ratio*
Photos of 1000 × 1000	70.167s	69.694s	22.563s	23.438s	2.99
Photos of 2000 × 2000	287.165s	285.496s	90.557s	96.163s	2.98
Photos of 3000 × 3000	652.001s	647.388s	207.514s	218.002s	2.987

From Table 9, it appears that the revised CPU application by removing the use of OpenCV library has a slightly faster performance than original CPU code. The auto GPU code generated by the C2GPU toolkit can speed up the original CPU application about 3x. The manual GPU application has a slightly improved performance over the machine generated GPU code. Also, the resolution of the target images has no strong impact on the performance of the C2GPU toolkit.

3.2.4 PRNU estimation in video signals

PRNU stands for photo response nonuniformity (PRNU) and it is the key information estimated from the video signals enabling us to provide image and video ballistics services. Having a video signal consisted of thousands of frames, PRNU is estimated from each frame separately which is computationally very expensive. An essential step in estimating PRNU is de-noising the image in every JPEG block (compressed block) separately. Moreover, in every block we need to compute the residual of the image and its de-noised version. This should be done in thousands of frames for an HD video. For example, 1280×720 pixels video of 10 minute length having 30 frames per second generates 4.320.000 blocks which should be analyzed separately. Thus, the need to GPU acceleration is obvious. The experimental results are shown in Table 10

Table 10 Evaluating the C2GPU in PRNU estimation in video signals

	CPU	*CPU without OpenCV*	*Manual GPU*	*Auto GPU by C2GPU toolkit*	*Speedup Ratio*
Photos of 1000 × 1000	0.344s	0.110s	0.143s	0.082s	4.19
Photos of 2000 × 2000	1.348s	0.434s	0.257s	0.213s	6.328
Photos of 3000 × 3000	2.988s	0.967s	0.495s	0.451s	6.625
Photos of 4000 × 4000	5.252s	1.691s	0.821s	0.729s	7.204
Photos of 5000 × 5000	8.21s	2.624s	1.192s	1.104s	7.436
Photos of 6000 × 6000	31.435s	9.177s	3.892s	3.760s	8.36

From Table 10, it appears that the revised CPU application by removing the use of OpenCV library has a significant improvement than the original CPU code (three times faster). The machine-generated GPU code can speed up the original CPU application about 6-8x. The C2GPU toolkit is well suitable to deal with this application.

3.3. Competitivity

In order to know how the C2GPU toolkit behaves compared to other CPU-to-GPU translators, we attempt to use MINT, Bones, Par4All, Polyhedral Benchmark, OpenACC and OpenMP to evaluate some sample codes.

We identify a number of typical directive based source translators and compare their performance in Table 11. OpenACC and PGI are both commercial GPU programming tools with stable applicability but not outstanding speedup performance in practical applications. CUDA-lite introduces some directives to improve memory hierarchy of CUDA, but it cannot directly support C++. hiCuda can optimize CUDA code by dealing with global memory and transformations to leverage the complex memory hierarchy. But it requires users to have some GPU programming experience. Compared to hiCuda, MINT is an easy-use CPU-to-GPU source translator containing only five types of pragmas. It is designed for accelerating stencil computations on the NVIDIA GPU. This translator accepts the input of C source with some intuitive MINT directives, and then generates CUDA C with speedup performance up to 10x.

Table 11. Comparison of properties of typical directive-based tools

	hiCUDA[7]	***PGI (OpenACC)[9]***	***MINT[5]***	***CUDA-lite [6]***	***C2GPU toolkit***
Language support	C-to-CUDA	C/Fortan-to-CUDA	C-to-CUDA	CUDA-to-CUDA	C/C++-to-CUDA/OpenCL

Easy-use of directives	Complex	Very complex	Easy	Easy	Easy
Applicability	Good	Outstanding	Limited	Good	Outstanding
Speedup performance	Good	Good	Outstanding	Good	Good
Optimisation option	Use of shared memory	No particular one	Shared memory and loop aggregation	Improved memory hierarchy	Improved memory hierarachy (use CUDA Texture)
Readability of GPU code	Moderate	No	Good	Good	Outstanding

The following issues have been observed towards these existing CPU-to-GPU translators.

- Applications written in C++ cannot be processed by most of the above tools. Bones and Par4All do not accept C++ language as an input source, so they cannot process the given applications. Polyhedral Benchmark also has similar problems. Meanwhile, Bones and Polyhedral Benchmar are algorithm skeleton based tools, their applicability is quite limited.
- Secondly, while MINT and OpenMP can be extended to support C++ language, it is indispensable to rewrite the original CPU code as an acceptable format CPU code for each tool. The actions of removing the use of external library and breaking up the variables dependence in parallelized regions are required. Consequently, we evaluate them to process three intensively computation filters: Mean Filter, Mode Filter, AlphaTrimmed MeanFilter. The input applications are written by C++ under Visual Studio 2008, with using external library OpenCV 2.4.3. The sliding window is given as 5×5 and 9×9, the tested image resolution is 3325×4765. The evaluation platform is under CPU i7-3770 (3.5GHZ) and GPU GTX 690. The results are shown in Figure 10

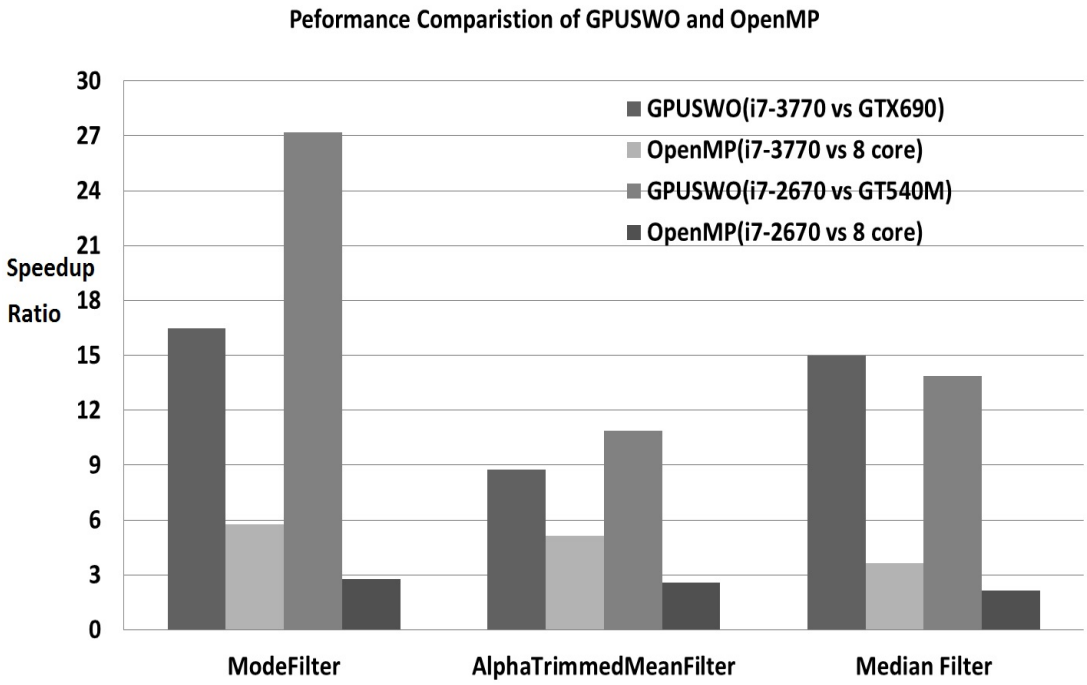


Figure 10. C2GPU toolkit performance over OpenMP

Figure 10 shows that for normal hardware, the C2GPU toolkit can accelerate the typical benchmark filers up to 12-27x, which is much higher than OpenMP's performance. It suggests that the C2GPU toolkit has a significantly competitive advantage than OpenMP.

We do not demonstrate its speedup performance of MINT in this Figure 10 because the generated results have huge error. This is because MINT is particularly designed for solving stencial computing problems..

To sum up, the C2GPU toolkit is highly competitive against the state-of-the-art CPU-to-GPU source translators. Apart from its outstanding acceleration performance, it also supports C++ language as input source code. This advantage makes it as a more end-user friendly than other research focused tools, such as hiCUDA, MINT, Par4All and Bones. Compared to the commercial products like PGI, the output code of the C2GPU toolkit is more readable and revisable CUDA or OpenCL code. This feature gives users the opportunities to learn GPU technology and to further modify the existing code.

3.4 Other issues

3.4.1 Security

The security requirement aims to protect the source code of the SME users. The current C2GPU web-interface provides a user registration system to access the toolkit. It provides the registered users with private-keys to view their source code. In general the security scheme can satisfy the user requirement to protect their code. One issue that needs some further attention is that the user password and private key are currently stored into the cookie of the browsers unless users delete the cookies.

The users can also delete their uploaded files. If they do not delete their files, these files are encrypted to store on server for 30 days. After 30 days, the files will be deleted so users have to upload the files again if they need.

3.4.2 Usability

The usability of the C2GPU toolkit mainly lies in the friendliness of the C2GPU web-interface. The SMEs partners have used the C2GPU web-interface to upload, convert their C/C++ source code, and download the machine generated CUDA or OpenCL code. The evaluation procedure involves the test of the server functions, user-friendliness, efficiency and accuracy.

Most of the essential functions stated in the user requirements have been achieved by providing the server service. This includes the transfer of source codes for analysis, converting CPU source code for GPU processing, running performance diagnostics with the toolkit, validation of converted source codes and creating reports/logs. In addition, the sample files can be accessed in the web-interface of C2GPU toolkit after user log in; a reminder message for private key automatically occurs when users log in at their first time; users can add pragma by either keying in or using a dialogue box.

The efficiency of the C2GPU toolkit is good. The processing time of running the toolkit for each operation is less than 5 seconds, which is acceptable by all the SME partners.

3.4.3 Adaptability

The adaptability of the C2GPU toolkit indicates how easily and efficiently for novices to learn how to use the C2GPU toolkit. GPU programming requires a steep learning curve for novices. The C2GPU toolkit features a great potential in bringing a cost-effective solution for accessing GPU power. The evaluation of the adaptability involves four parts, including the understanding of loop patterns, algorithm skeletons, pragmas and warning messages. In summary, the adaptability of the C2GPU toolkit is good. While the understanding of the kernel generation pragmas is still hard to new users, the loop pattern and algorithm skeleton appear to be easy to understand by users. Also, the use of warning messages are well-received by users.

4. Conclusions and Recommendations

In conclusion, the C2GPU toolkit has satisfied all the essential requirements from SMEs' requirements to accelerate their applications with god the usability and adaptability of the .The C2GPU toolkit is highly competitive in the state-of-the-art automatic CPU-to-GPU source translators.

The SME users feel the benefit of the toolkit is visible and it can save a lot of time, especially for those programmers who do not have rich experience in GPU programming. A parallelized version can be generated using the C2GPU toolkit without going into the details of CUDA or parallel programming.

The consortium members also have investigated the work for future development.

- To strength some automatic dependencies analysis of CPU code, which is capable of assisting users to accurately identify where need to be parallelized.
- To improve user experience of editing source codes online by using some popular 3rd party libraries.
- To further improve the speedup performance of the C2GPU toolkit by introducing further optimisation methods.

III. IMPACT

1. Project Impact

The overall benefit of the GPSME toolkit (which is now branded a new title called C2GPU toolkit) includes:

- Significant performance gain by utilising the latest GPU architecture in a fully automated manner.
- Easy and automatic access to GPU power without modifying the structure of the targeted software.

Consequently, the users (programmers) are able to convert their written CPU code into GPU implementation with an expectation of significant performance gain. An attractive feature of the toolkit is that it keeps the current structure of the targeted software and thus does not affect the other software components. The toolkit will require only very limited GPU knowledge and labour/time input from the users. They will be able to continue to use their familiar language and environment to construct their program, which will later be converted into the GPU version to obtain significant performance gains.

By this means, the SME participants are able to improve the performance of their techniques to allow execution within practically acceptable runtimes. Moreover, many advanced techniques and computing models that are prohibitive to CPUs due to their excessive requirement of computing resources can be adopted owing to the involvement of GPU.

1.1 IME

IME is set to use the toolkit to reduce the processing time of the advanced forgery detection techniques developed by the company. The accelerated techniques will be integrated into the main product, which the company will continuously invest in the following years after the completion of the project. The easy access to the GPU power will boost the performance and hence significantly increase the competitiveness of the company.

IME initially plans to incorporate the results of GPSME into two projects of the company. IME's major product, Verifeyed, enables digital image and video authentication and the results obtained in the GPSME project have shown the power and benefit of GPU acceleration of imaging forensics methods. In particular, the running-time of several methods including the detection of copy-move forgery detection have been accelerated by, on average, 3x to 6x. This increase in efficiency enables them to integrate computationally extensive advanced methods into Verifeyed, and these will have values for law enforcement, in the digital forensic market, and in media industry, resulting in a competitive advantage for the company. Specifically, so far, the method searching for near-duplicated (enhanced by the C2GPU toolkit) has also been released to the public.

GPU-accelerated image-forensic methods will be integrated into Verifeyed step by step, and in several phases. In the first phase, they will enter a new vertical market (the media industry) and will attract a selected set of clients of this sector to become early adaptors of advanced image and video authentication methods. These selected end-users will receive beta versions of the GPU-accelerated product. This phase will be followed by user experience and feedback collection in order to finalise the GPU-accelerated product for market release (Q1 of 2014).

IME also has an on-going video categorisation project which has lacked effective implementation because of its heavy computational demands. The results of the GPSME project and the associated GPU acceleration have made it possible for IME to re-start this project. This is likely to have a noticeable impact on the video categorisation market and the fight against videos that contain inappropriate content.

The target market will be insurance companies, media, police and forensic laboratories, scientific journals and publishers. There are strong indications of great potential in the future market from forgery image detection based on the evidence of growth in the targeted areas. For example, the insurance industry is growing extremely fast, particularly in developing countries such as China and India, but there is evidence that it is impeded by fraud in many countries. A UK report identified an increasing number of fraudulent applications, and one of the main reasons behind losses to insurance industry is fraud.

Disklabs, which commands more than 50% of the computer forensics market, has predicted that the expansion of computer forensics is a significant trend that points to an untapped market with tremendous potential. In scientific journals, it has been reported that the overall journal growth characteristics clearly show the predominance of 3.3% compound annual growth under a number of different socio-political climates. Given the number of high profile fraud cases in scientific research, such as the doctored images published by Professor Hwang Woo-Suk in stem cell research, IME expects an increase of activities and demands for fraud prevention in scientific journals.

The business model is based on licensing and providing professional services. The sales force will directly offer the system to the European potential customers. For the US and other non-European customers, the company has established a partnership with international software integrators (e.g. Accenture, IBM, Logica). Using distributive channels via an international system integrator is a good choice for increasing sales and system integration capabilities. The next stream is to build a partnership with software companies that specialise in insurance fraud management systems such as SAS Institute, Adastra, etc. The company will implement the solution according to the customer application portfolio; implementation by selected partners is also possible.

GPSME will improve the main product of IME, which will be continuously invested by the company in the following years after the completion of the project. Therefore, the company has a very long term commitment and business plan for the product. IME is a relatively new company with a high level of innovation and growth potential. The product is based on many years of research and is very distinctive in the market at the moment. Having easy access to the computational power of GPUs will open possibilities for IME to develop more forgery detection methods that have runtime performance within acceptable limits by directly using the toolkit after the project. This will help them to strengthen the current position and attract new customers

The C2GPU toolkit has also been introduced to, and discussed with, the Plug and Play Tech Center, one of the largest technology incubators in the US (specifically to VP of Technology of Fund). Since a high number of technology startups are potential users of the C2GPU toolkit, the Plug and Play Tech Center might refer potential users to the GPSME website.

The capability of the GPSME project and corresponding acceleration of image forensic methods have also been pitched to The National Center for Missing & Exploited Children (specifically, to Michael T. Geraghty Vice President, Chief Information Officer). There is on-going discussion about a pilot project in which the results obtained from the C2GPU toolkit would be applied to the photo database of The National Center for Missing & Exploited Children.

1.2 ROTA

ROTA works on Augmented Reality and uses image-processing techniques in most of its projects. The G2CPU Toolkit provides the ability to create applications optimised for the use of graphics cards which are faster than the standard products.

The company will exploit the C2GPU Toolkit regularly in its existing product line and in its continued product development, with the intention of introducing an extended and diversified product range and, it is predicted, an improved revenue stream. The new product will feature

improved immersiveness by having more advanced capacity in image processing and analysis. ROTA will firstly start to convert one of their books using the C2GPU toolkit.

The sales of electronic books (e-books) are expanding exponentially compared with traditional books. In the USA, it is predicted that digital textbook sales will surpass 18% of combined new textbook sales for the Higher Education and Career Education markets. Similar cases are also reported in Europe. A new French study shows that between 15% and 20% of the book-reading public will own electronic devices and up to 25% of books will be sold in digital form by 2015.

ROTA has a long-term commitment to, and business plan for, the AR book. It is one of the main products of ROTA, which will be continuously invested into by the company in the years following the project. To make the company more competitive, ROTA will continue to work on different types of Augmented Reality systems, such as IR camera based systems, inertial AR systems, to which the results of GPSME will also contribute. For example, the new projector-based AR system that they are working on for the education market can also use the C2GPU toolkit.

In common with many other SMEs, ROTA has difficulty in recruiting staff capable of GPU programming. The use of the C2GPU Toolkit will enable new products to be developed using GPU technology and brought to the market much more cheaply (and more effectively for the long-term health of the company) than if external contractors were employed to supply the necessary expertise.

1.3 B3C

B3C continues to develop an open software framework called MAF (www.openmaf.org), which is a widely used software library for medical imaging vertical applications. Those applications often contain either image processing or computational geometry algorithms that are computationally intensive; within the GPSME project, an exemplar of these algorithms is used as a test case, the centerline extraction, which is one of the most time-consuming steps in vascular imaging applications. Beside this, there are other notable examples, such as the customisation of a level-set algorithm for segmentation purposes in a vascular imaging application; an innovative semi-automatic curved multi planar reconstruction for a dental application; an impingement detection engine based on collision detection techniques in an orthopaedic application (HipOp). These tasks typically hang the application for a time that may range between one and thirty minutes: as the intended user is a clinical professional, this delay causes great inefficiency in the productive workflow of the customer. Besides, a sensible improvement will greatly enhance the user experience.

It is not possible to quantify an economic impact of these enhancements on product sales, as these applications are currently prototypes in the engineering phase, but on the costs side, the benefit is evident. In fact, the programmers estimate that the time needed to port these pieces of code to the GPU by normal means as six months: as our programmers are typically involved in consultancy contracts, the cost of such development can be seen as lost revenue of €50k for each feature. We expect that, by using the C2GPU Toolkit, these development costs can be reduced by a factor of 80%, while retaining the same level of performance speed-up.

1.4 ANS

ANS is involved in developing mobile apps for use in a medical context. It has a collaboration with Moorfields Eye Hospital, the leading centre for ophthalmology in the UK, covering several different applications all of which are, as one may expect, highly involved with images and their analysis for which the GPU is particularly suited. Mobile apps are becoming ever more sophisticated and computationally demanding, and the evolution of GPU technology to mobile platforms is making it possible to consider extending the applications and broadening their range even further.

Mobile phones are also increasingly being connected to devices that can be used to capture data from subjects. This can take place, for example, if a physician needs to monitor a specific condition remotely, where the readings can take place automatically or be under the control of the patient, or if general citizens themselves wish to exert greater control over their general health and lifestyle. In the context of ophthalmology, the recent introduction of a high quality lens that can be attached to a smartphone to enable remote capture of the retina, is a very exciting development that can greatly change the way in which certain conditions can be treated, so this is a field that we anticipate will grow rapidly in the near future.

ANS is using the C2GPU toolkit to further enhance its eye image analysis techniques. The C2GPU toolkit allows ANS to improve the processing speed of their products by a large margin and hence supports the execution of the detection within clinically acceptable runtime. The new product is expected to be ready within approximately two to three years from the completion of GPSME.

ANS is also considering building a web service to offer free eye image analysis information on eye disease. In this way, the company expects to attract attention from the general public.

The eye image analysis software is one of the series healthcare products developed at ANS. The successful application of the C2GPU toolkit will lead to further enhancement of many products, all of which place great demands on processing speed. The healthcare series for ophthalmic patients are the company's key products, and ANS will commit to the work by making continuous investment into the long term. To this end, GPSME has been very timely for ANS in terms of help[ing to steer its future direction.

Notably, ANS has been providing the host for the web-based C2GPU toolkit in the duration of the project. It has been confirmed that such a host will continue after the completion of the project. ANS has organised event and started to broadcast news of the toolkit in its network. The online service is described in other technical deliverables of the project. Through the online web service, ANS will take charge of building a user community for the C2GPU toolkit from the registered users. Once the user base has reached a critical mass, further community services will be supported by ANS in the form of a forum, wiki, etc.

ANS will continue to carry out evaluation to the toolkit for any possible future development. An online user evaluation page is provided alongside the online web service, which allows users to provide feedback to the service.

2. Dissemination Activities

2.1 Project Website

The GPSME website available at http://www.gp-sme.eu, and it contains both a public and a private area. The following figure shows the page to access the C2GPU toolkit with newly designed logo and image.

Figure 1 A newly designed interface for the online web service

2.2 International Conferences and Courses

The project and its related work have been presented in a number of events worldwide.

- Course: Introduction to GPGPU and CUDA programming - 9-10 May, 2013 Bologna
- International Conference on Parallel Processing and Applied Mathematics, 8-11 Sept, 2013, Warsaw, Poland
- International Conference on High Performance Computing & Simulation, 1-5 July, Helsinki, Finland
- International Conference on Computer Medical Applications, 20-22 Jan, 2013, Tunisia
- 6th Pacific-Rim Symposium on Image and Video Technology October 28th - November 1st, 2013. Guanajuato, México.

2.3 Publications

The following papers have been published on the following refereed conferences.

1. D. Williams, V. Codreanu, P.Yang, B. Liu, F. Dong, B. Yasar, B. Mahdian, A. Chiarini, X. Zhao and J.B.T.M. Roerdink, Evaluation of autoparallelization toolkits for commodity graphics hardware, In Proceedings of 10th International Conference on Parallel Processing and Applied Mathematics, Sept 8-11, 2013 (accepted).

2. V. Codreanu, F. Dong, B. Liu, J.B.T.M. Roerdink, D. Williams, P. Yang and B. Yasar , GPU-ASIFT: A Fast Fully Affine-Invariant Feature Extraction Algorithm. In International Conference on High Performance Computing & Simulation 2013, July 1-5. IEEE, 2013.

3. D. Williams, V. Codreanu, J.B.T.M. Roerdink, P. Yang, B. Liu, F. Dong and A. Chiarini. Accelerating Colonic Polyp Detection Using Commodity Graphics Hardware. In Proceedings of the International Conference on Computer Medical Applications. 2013.

4. B. Mahdian, S. Saic, P. Yang, B. Liu, and F. Dong, Source camera identification using multiplicative sensor noise component, PSIVT 2013: 6th Pacific-Rim Symposium on Image and Video Technology 28 Oct - 1 Nov 2013, Guanajuato, Mexico

2.4 Flyers & Posters

We have created flyers and posters and distribute them in various events that the consortium members participated. Figure 2 shows a sample of the project flyer and poster.

Figure 2 Project Flyers

2.5 Videos

The project has created a number of videos for project introduction as well as for tutorial purpose. The videos have been uploaded onto the project website & Youtube (http://www.youtube.com/watch?v=XjVv8AoRHZ8). Figure 3 shows a snapshot of the tutorial video.

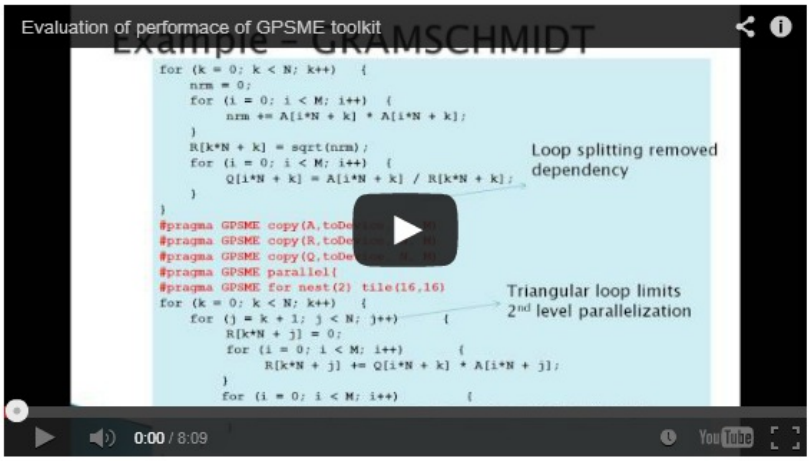


Figure 3 An illustration of the tutorial videos

2.6 Press interviews and articles

There have been a number of interviews with press. Figure 4 provides a snapshot.


Sunday, Sep 29th Last update 10:02:27 AM GMT **Headlines:** Lawyers who will never want for work

HOME | NEWS | TECH TRAIL | TRADE FLOOR | EXPORT TO | THE KILLER50 | PUBLICATION

Business Awards 2013
Enter your company
before December 20th

The Killer 50
We showcase the 50 hot
disruptive technology co

YOU ARE HERE: HI-TECH ▸ LIFESAVING SOFTWARE GAINS EUROPEAN FUNDING

Cambridge Healthcare

Tuesday, 20 September 2011 10:51 BUSINESS WEEKLY

LIFESAVING SOFTWARE GAINS EUROPEAN FUNDING

Life-saving software could soon be developed much more quickly and easily thanks to a European research project being led by the UK's University of Bedfordshire.

The university is the lead partner of a pan-European consortium that won European Commission Framework Programme 7 (FP7) funding. The FP7 grant will see €1.13m of EC cash invested into a €1.5m project for the development of new tools.

These will enable small firms to access the processing power of graphical processing units (GPUs) when developing new software.

"It will help progress technology development in life-saving fields such as medical imaging and forensic image analysis," says Feng Dong, Professor of Visual Computing in the university's Department of Computer Science and Technology.

Partners from the Czech Republic and Turkey will be working on the collaboration. Prague-based ImageMetry is a research business specialising in image forensics and image processing while RotaSoft focuses on augmented reality research and 3D image processing and rendering. "Both offer capabilities that are essential to the project consortium," said Feng.

Feng said it was a first contact with Business Link that led to a partnership request to the Enterprise Europe Network which in turn developed into a successful grant application.

Figure 4 a snapshot of GPSME in Press

2.7 Dissemination activities at individual partners

ALL the partners have been actively involved in a very wide variety of dissemination activities, which have all been reported in D6.3. We do not repeat them here due to the page limit.

2.8 Plan for the future dissemination

The SMEs are interested in the potential of the project and will be committed to further dissemination activities. The result of GPSME will be further exploited through a range of activities, leading to further investigations on business avenues for commercial activities:

- The web-based C2GPU toolkit will be used as one of the major means for dissemination and exploitation by offering free services to programmers. Once a considerable user base is achieved, we will consider starting to build user communities by using social network .
- All the project resources, including the course materials, open-source toolkit will be carefully maintained.
- New academic papers, GPSME courses and tutorials, exhibition will be submitted to journals and conferences to create impact among research and industry communities, taking into account the IPR protection issues.
- Joint dissemination activities can be carried out with established research communities in which GPU has a great potential.
- We will disseminate news to computer giants such nVidia for possible future collaboration.
- We will continue to use Trade fairs, user meetings and in-house magazines to promote the C2GPU toolkit. Exhibition days will be arranged to demo the latest work to the public. Subject to further arrangement, we will run similar exhibitions at conferences.

- Courses and symposiums will open to the public, which is expected to attract the communities and may lead to opportunities for presenting the work at other locations. Tutorial on the C2GPU toolkit will be presented at future events that the SMEs are going to participate.
- The materials of the GPSME project, such as flyers and posters will be constantly updated. E-newsletters will be developed and be made available to the public.
- Updated video-clips about GPSME will be made and uploaded to the Youtube where necessary.
- At some stage, publishers will be approached to see if there is interest in producing a book (or ebook) presenting the C2GPU toolkit (e.g. dummies guide to the C2GPU toolkit).

These activities will target the general public(including school students, local developers, industries and funding bodies) to raise the recognition of the GPSME results, which will enhance the public engagement.

3. Exploitation

3.1 Exploitable foregrounds of Individual Partners

The exploitable foregrounds of the SMEs were introduced in Section 1. More details of the exploitation from individual partners were also described in Deliverable D6.4.

3.2 Competitor analysis of the C2GPU toolkit

A comparative study is carried out to identify the main competitors of the C2GPU toolkit by looking into its strengths and weaknesses. Currently, the main competitor of the C2GPU toolkit are MINT, OpenACC, hiCUDA and CUDA-lite.

While OpenACC and PGI are commercial GPU programming tools with stable applicability, they do not have outstanding speed-up performance in many practical applications.

MINT is an easy-use CPU-to-GPU source translator containing only five types of pragmas. It is designed for accelerating stencil computations on the NVIDIA GPU. This translator accepts the input of C source with some intuitive MINT directives, and then generates highly optimized CUDA C with speedup performance up to 10x. but the range of programming patterns that can be handled by MINT is very limited.

OpenACC is a relatively new technology, with a first version finalised in 2011 and the latest version, OpenACC 2.0, finalised during 2013. Coupled with the lack of freely available and mature implementations, this has meant that the technology has not yet been widely evaluated by the academic community. The few available evaluations focused primarily on small test cases though some application to real-world code has also been performed; in all cases, significant speedup was observed on sections of parallelisable code. It should be noted that the OpenACC compilers are still undergoing rapid development due to the standard being so new.

Performance between OpenACC and our toolkit is very close, which is to be expected since they perform a similar set of operations and are running on the same hardware, with slightly better performance by the C2GPU Toolkit (due to efficient register and shared memory usage), though it should also be noted that OpenACC is more generic and so may perform better on other applications.

CUDA-lite introduces some directives to improve memory hierarchy of CUDA, but it cannot directly support the CPU language.

hiCuda can optimize CUDA code by dealing with global memory and transformations to leverage the complex memory hierarchy, but it requires considerable user knowledge of GPU programming experience to specify the use of threads and threads block.

3.3 Exploitation of the C2GPU Toolkit

The only collective exploitable item is the C2GPU Toolkit. This provides a useful means of enhancing the computational capabilities of SMEs.

Successful GPUified products developed by the SMEs in the consortium serve as convincing case studies and strong proof of concept of the C2GPU Toolkit. A systematic demonstration activity framework will provide an effective introduction to the market of the results that can be achieved by use of the Toolkit and provide incentives for its deployment in other areas.

To achieve these goals, the following actions can be performed:

- market-oriented adaptations, integrations, and testing;
- develop and release an open web service for automatic GPUification;
- put into production the SMEs' results obtained by automatic GPUification;
- provide end-users and SMEs with suitable technical support and maintenance;
- create success stories, feasibility studies, and proofs of concept.

Possible models for software developers include toolkit-centred products, online services or integration into existing products. Of these, the first two are viable options for the C2GPU Toolkit being developed in GPSME, and these suggest the following three specific possibilities:

a) make the code available as a "sealed" product that users can acquire;
b) make the source code available to users so that they can incorporate it into their products;
c) provide a service by which users submit code and receive the outcomes of the processing performed by the C2GPU Toolkit.

All of these can either be either sold or provided at no cost.

It is likely that many industries will be able to benefit directly from the ability to convert written CPU code into GPU implementation with an expectation of significant performance gain. Examples provided were Bioinformatics, Computational Finance, Computational Fluid Dynamics, Data Mining, Defence, Electronic Design Automation, Imaging and Computer Vision, Material Science, Medical Imaging, Molecular Dynamics, Numerical Analysis, Physics, Quantum Chemistry, Oil and Gas/Seismic, Structural Mechanics, Visualization and Docking, Weather and Climate.

3.3.1 Open Services and Open Source Software

Given the early stage of development of the toolkit, it is not yet feasible to market it as a fully fledged commercial product, so there are three main priorities for the immediate future:

- to create a user base to give the product credibility and demonstrate a level of acceptance;
- to apply it to a wide variety of CPU software to ensure its versatility and robustness;
- to establish it as a "brand" that has reliability, effectiveness and efficiency.

These priorities are inter-related and steps taken to establish address one of them will generally also have an impact on the other two.

Once the user base is established and the software has gained acceptance, services such as consultancy or training can be offered to the user community to further enhance the brand.

It is considered that, at its current state of development, the C2GPU Toolkit is not viable as a commercial product in the form of option (a) above at present, and that the priorities listed above can be achieved most effectively by its release as a free product, at least in the short term. However, the experiences of the SME partners have already demonstrated that it provides a useful means of enhancing the computational capabilities of SMEs and the

successful GPUified products they are developing will serve as convincing case studies and a strong proof of concept of the C2GPU Toolkit.

Option (c) above, the web-based translation service, will be offered in the first instance. This will suit companies who wish to treat the Toolkit as a “black box” and simply implement the code that it produces without modification, which is the most suitable option for the SME market in its current state of awareness of GPU technology.

This service will be provided as an open service to users at its current site http://gp-sme.co.uk/web_face/. Users will be able to sign up to the service for their free account, from which they can upload, convert and then download their code. The service comes with an online editor, with which the users can edit their code, taking advantage of the editing facilities available (e.g. semi automatic pragma insertion).

To supplement the GPSME website, which will continue in operation after the conclusion of the project, partner ANS will also make available the existing tutorials (and any further such materials) on its web site and the other SME partners will provide descriptions of success stories resulting from use of the Toolkit and will retain links to the GPSME site for the information of visitors.

The service will be free in the initial stages. Once a stable user base is reached, and the software has been further developed to include advanced features, we will introduce fee-charged services; by that time, it may also be possible to start to attract advertisements.

Option (b) above, the provision of the code as open source, would be more suited to companies that have a sizeable IT department with personnel capable of understanding the code and the context in which it is applied in the GPU processing pipeline. This will be developed in the second phase when the code is more mature. This would be linked with the creation of a programmer-based User Group which would bring in assistance to develop further the functionality available in the Toolkit; Partner ANS has expressed a willingness to coordinate such activities. Certain websites provide support for code sharing and relevant user groups, and these would be used to assist in this. Subsections below discuss a number of licence options under this category.

This business model is very important for open source products. We anticipate offering the basic software for free, while charging for the premium software which would have advanced features. The purpose would be to attract new users and hence to promote the sale of our advanced product or the use of our advanced services, which would both be associated with a fee charge. The financial return on open-source software can also come from selling services, such as training and support, rather than the software itself. Advertising can be another important avenue for business exploitation - the more traffic we have, the more we can charge for adverts.

3.3.2 Licensing & Protection

The precise form of licence to be used for any open source distribution of the C2GPU Toolkit will be kept under review. Possible choices include:

Open Source Licences

Many developers want to release their software as open-source projects as they want others to be able to build on and share their code. The open-source community is vibrant because of this. For this, licensing is an alternative to either simply releasing the work into the public domain or granting permissions on a case-by-case basis. In the former case, the author relinquishes any copyright, and nobody using the work is obliged, formally or informally, to recognise him/her as the originator. The latter case may require a lot of unproductive time to be spent dealing with individual permissions.

GNU General Public License (GPL)

The GPL is possibly the most commonly used licence for open-source projects. It grants and guarantees a wide range of rights to developers working on open-source projects, allowing them legally to copy, distribute and modify the software.

There is also a Lesser General Public License (LGPL) which grants fewer rights to a work than the standard GPL.

BSD License

BSD licenses have fewer restrictions on distribution than the GNU General Public License. The most frequently used versions are the New BSD License/Modified BSD License, and the Simplified BSD License/FreeBSD License, both of which have been verified as GPL-compatible free software licences, and have been accepted as open source licences by the Open Source Initiative.

MIT License

The MIT License is the shortest and probably broadest of all the popular open-source licences, and It basically says that anyone can do whatever they want with the licensed material, as long as it is accompanied by the licence.

Apache License

The Apache License, Version 2.0, grants a number of rights to users; these can be applied to both copyrights and patents (some licences can be applied only to copyrights and not to patents). The Apache License allows the following:

- Rights are perpetual and irrevocable – once they have been granted, they cannot later be rescinded, nor is there a time limit on their use.
- Rights are worldwide – once granted in any country, they are granted in all countries.
- Rights are granted for no fee or royalty – there is no upfront fee nor are any fee related to usage.
- Rights are non-exclusive – you can use the licensed work, and so can anyone else.