

GPU accelerated visual analytics tool for people flow tracking data

Miika Lehtimäki

School of Electrical Engineering

Bachelor's thesis
Espoo 09.05.2017

Thesis supervisors:

D.Sc. (Tech.) Pekka Forsman

Thesis advisors:

Prof. Jaakko Lehtinen

D.Sc. (Tech.) Marja-Liisa
Siikonen



Author: Miika Lehtimäki

Title: GPU accelerated visual analytics tool for people flow tracking data

Date: 09.05.2017

Language: English

Number of pages: 5+23

Degree programme: Automation and Information Technology

Supervisor: D.Sc. (Tech.) Pekka Forsman

Advisors: Prof. Jaakko Lehtinen, D.Sc. (Tech.) Marja-Liisa Siikonen

Data visualization is ubiquitous on education and science. Recent advances in information technology in respect of data gathering, transferring, storing and processing capabilities have made development of visual tools with powerful capabilities possible. Visual analytics is a emergent field studying cognitive process formed by human and computer via interactive visual interface.

Understanding people flow is crucial when designing buildings and public areas. Quality of daily lives of people can substantially increase by planning urban environment carefully. Furthermore, in case of evacuation insufficient planning can lead to unnecessary injuries or deaths. With advanced simulation and analysis tools safer, more intelligent and user-friendly environments can be designed.

This thesis studies possibility of real-time visual analysis software designed for people flow data. Particular subject of focus is rendering of kernel density estimation or "heatmap" visualization using graphics processing unit present in modern computers. An experimental software was implemented and performance measurements of heatmap rendering and data filtering were conducted.

From the research results it can be concluded that such real-time application running at 30 frames per second can be created for dataset consisting of tens of thousands of paths using PC hardware equipped with GPU. Further development is necessary for larger datasets and systems with weaker performance. The thesis outlines multiple potential paths for improvements over the experimental software.

Keywords: Data Visualization, Visual Analytics, People Flow, Real-Time Rendering, Kernel Density Estimation

Tekijä: Miika Lehtimäki	Päivämäärä: 09.05.2017	Kieli: Englanti	Sivumäärä: 5+23
Koulutusohjelma: Automaatio- ja informaatiotekniikka			
Vastuuopettaja: TkT Pekka Forsman			
Työn ohjaaja: Prof. Jaakko Lehtinen, TkT Marja-Liisa Siikonen			
<p>Datan visualisointi on läsnä kaikkialla opetuksessa ja tieteessä. Tietotekniikan kehittymisen datan keruun, siirron, tallentamisen ja prosessoinnin osalta on mahdollistanut uudenlaisten, tehokkaiden visuaalisten työkalujen kehittämisen. Visuaalinen analyysi on tuore ja kasvava ala, joka tutkii ihmisen ja tietokoneen visuaalisen rajapinnan kautta muodostamaa kognitiivista prosessia.</p> <p>Ihmisvirtojen ymmärtäminen on tärkeää rakennuksia ja julkisia tiloja suunniteltaessa. Ihmisten jokapäiväinen elämänlaatu voi parantua merkittävästi, mikäli kaupunkiympäristöt suunnitellaan huolellisesti. Lisäksi evakuatiolanteissa riittämätön suunnittelu voi johtaa tarpeettomiin henkilövahinkoihin. Edistyneitä simulaatio- ja analytiikkatyökaluja käyttäen voidaan suunnitella turvallisempia, älykkäämpiä ja käyttäjäkunnon paremmin huomioivia ympäristöjä.</p> <p>Tämä opinnäytetyö tutkii mahdollisuutta kehittää reaalialkainen työkalu ihmisvirradatán visuaalista analyysiä varten. Erityisesti tutkimus kohdistuu ydinestimaatiovisualisaation eli ns. "lämpökartan" renderöintiin nykyaisista tietokoneista löytyvästä grafiikkakäihdytintä käyttäen. Kokeellinen sovellus implementoitiin ja suorituskykymittaukset sekä lämpökartan renderöinnistä etä datan suodatuksesta tehtiin.</p> <p>Mittaustuloksista voidaan todeta, että tällainen 30 Hz ruudunpäivitysnopeutta käyttävä reaalialka-applikaatio on mahdollista toteuttaa kymmenien tuhansien polkujen tietoaineistolle grafiikkasuorittimella varustettua PC-laitteistoa käyttäen. Jatkokehitys on tarpeen, mikäli on tarvetta käyttää suurempia tietoaineistoja tai heikompitahoista laitteistoa. Työ esittää pääpiirteittäin monia potentiaalisia jatkokehityssuuntia kokeellisen sovelluksen suorituskyvyn parantamiseksi.</p>			
Avainsanat: Datan visualisointi, Visuaalinen analyysi, Ihmisvirrat, Reaalialkarenneröinti, Ydinestimointi			

Contents

Abstract	ii
Abstract (in Finnish)	iii
Contents	iv
Abbreviations	v
1 Introduction	1
2 Background	2
2.1 People flow analysis and building planning	2
2.2 Histogram and kernel density estimation	3
2.3 From information visualization to visual analytics	4
2.4 Human-computer interaction and real-time rendering	7
3 Research on experimental visualization tool	10
3.1 People flow data	10
3.2 Tool features	11
3.3 Implementation and testing platform	12
4 Results	14
5 Conclusion and further development	17
References	18
A Finnish summary - suomenkielinen tiivistelmä	21
A.1 Johdanto	21
A.2 Ihmisvirtojen analysointi ja rakennusten suunnittelu	21
A.3 Visuaalinen analyysi ja reaalialikarenderöinti	22
A.4 Kokeellinen työkalu ja tulokset	22
A.5 Johtopäätökset ja jatkokehitys	23

Abbreviations

Acronyms

API Application Programming Interface

CPU Central Processing Unit

CSV Comma Separated Values

FPS or fps Frames Per Second, screen refresh rate

GLSL OpenGL Shading Language

GPGPU General Purpose computation on a GPU

GPU Graphics Processing Unit

KDE Kernel Density Estimation

PC Personal Computer

XML Extensible Markup Language

1 Introduction

Visualizations are one of the most convenient and powerful tools for analyzing and transmit information. They are commonly used in education and science for their features such as supporting learning and highlighting emergent features in data. Graphs, charts, maps and diagrams are familiar to virtually everyone regardless of field of employment.

Recently, fields of scientific and data visualization have been augmented by the advances in development of interactive representations and tools. This emergent field referred to as visual analytics increases cognitive capabilities such as working memory, recognition of patterns and decision making through the utilization of powerful image processing ability of human vision.

Converting large datasets into human-comprehendable form requires vast amount of computing resources. Traditionally the amount of data used would have had to be reduced or the process would have been inconveniently time-consuming. Real-time applications for data visualizations were made for only small datasets limiting the usefulness of such softwares substancially.

Increases in capabilites of computing hardware, especially extremely parallelized graphics processing units has enabled the creation of real-time visualizations of very large datasets. In company with development of user interaction and rendering technologies seamless visualization interfaces can be designed. These interfaces enable the user to conveniently generate distinct visualizations of the data as necessary.

The objective of this thesis is to examine and evaluate methods for creating a visual analytics tool for filtering and visualizing people tracking data consisting of paths on a 2D plane. Following this the gathered knowledge was applied to implement an experimental software demostrating plausibility of such manner of an approach. A dataset captured by Kone Oyj was used to evaluate the performance of the software.

Structure of this thesis is the following: chapter 2 outlines the current state of fields related to this kind of application such as people flow research, information visualization, visual analytics and real-time rendering as well as motivation behind them. Chapter 3 describes the features and implementation of the experimental software alongside with the data set used. Chapter 4 presents the measurement results and Chapter 5 contains the conclusion and ideas for future development.

2 Background

2.1 People flow analysis and building planning

Understanding flow of people groups is relevant on wide range of fields including *Computer Vision*, *Computer Graphics* and *Pedestrian Evacuation Dynamics*. Accurate analysis and synthesis of behaviour of individuals and small and large groups leads to applications in surveillance, social psychology, and design of new buildings and public places. Particularly the design process can benefit from this by taking account dynamically changing flow of people and minimizing the time required for evacuation.[1]

Regardless of the fact that all aforementioned fields of research endeavour to model behaviour of crowds of people, approaches have evolved independently. As a result each field manifests unique perspectives and techniques. In *Computer Vision* subjects of focus are surveillance, object detection and tracking. Analysis of collected data leads normal and abnormal behaviour detection, segmentation and classification of motion patterns and mathematical modeling of interactions among pedestrians in the crowd.[2, 3] *Computer Graphics* covers the aspects of simulating and visualizing people flow in real-world or synthetic environments. Methods include agent-based motion simulation, efficient real-time representation of virtual people, obstacle and collision avoidance, and modeling and motion planning for large, homogeneous or heterogeneous groups of people.[4, 5, 6, 7, 8] In *Evacuation Dynamics* similar effort for developing motion, interaction and self-organization models is underway. However, difference to computer vision and graphics is emphasis on empirical validation of simulated behaviours and movement models.[9, 10]

When planning buildings and their equipment related to people flow such as elevators, escalators, automatic doors, turnstiles, guidance and building information lot of factors affecting the users' experience has to taken into consideration. These include functionality, integrated building systems, interior design, ergonomics and ambiance of the building. Furthermore, for providing unified user experience for every individual, building user groups and flow of people and goods should be studied carefully. Examples of different users are employees, people delivering goods, visitors, people with restricted mobility and service personnel. When planning optimized people flow for such diversity of users whole journey from entrance to destination is to be considered rather than individual hotspots, services or devices. Such task requires data about most commonly used entrances, dwell times(time people spend in same area), bottlenecks, crossflows, sight lines and actions taken by users. With this approach, the experience can be planned based roles of individual users.[11]

It is clear that such multivariate design process requires assistance of advanced planning tools. A simulation tool is used to synthesize people flow data in different layout configurations. Analysis is done based on number of result parameters, most important being total journey time. It indicates time elapsed from entrance to final destination. Furthermore, vertical and horizontal components of the travel can be separately analyzed to achieve insight on performance of the elevators, usage of space or paths taken by users. Behaviour-related phenomena such as preferred routes and

groups emerge as a result of sophisticated models and algorithms within the simulator. Input for the simulation is presented in a form of profile consisting of intervals indicating intensity - the percentage of buildings total population currently moving, and traffic pattern - proportions of incoming, outgoing and interfloor traffic. For new buildings, simulations are done with assumed population and general premade profiles depending of type of the building. In case of renovation or modernization, actual profile gathered with sensory equipment is required to make optimal improvement suggestions and scheduling.[11]

2.2 Histogram and kernel density estimation

For data exploration and analyzing purposes, histogram is probably the most common visual representation tool used. It was first introduced by Pearson[12]. Histogram provides visual summary of single variable distribution consisting frequency counts of the data over discrete intervals, called *bins*. Weakness of the histogram is determining the bin size as there is no optimal number of bins and different bin sizes can reveal different aspects of the data.[13]

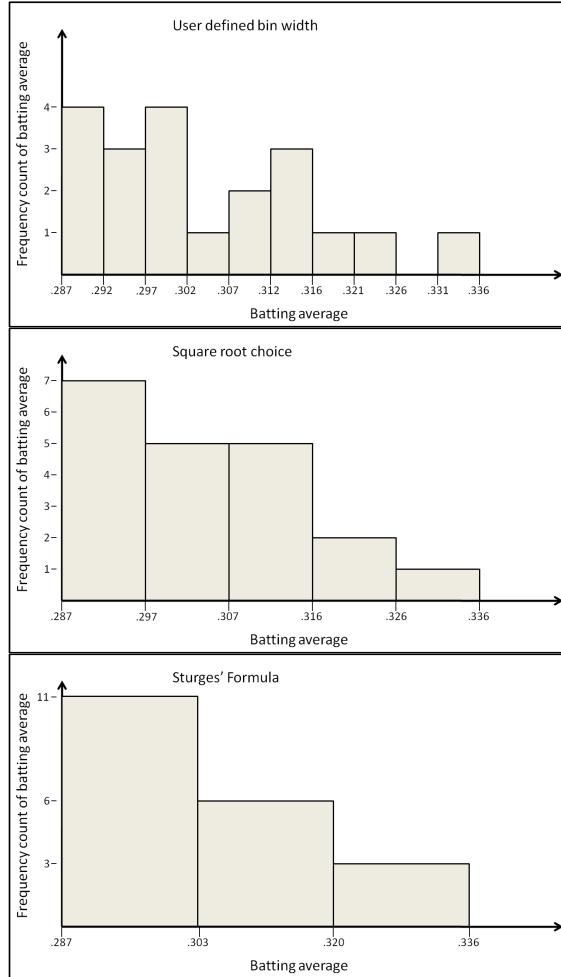


Figure 1: Histograms formed from the same dataset with differing bin widths.[13]

Figure 1 illustrates the effect of bin width variation. Three of the most common ways of choosing the bin width are represented. First shows user-defined bin width defined by

$$k = \left\lceil \frac{\max(x) - \min(x)}{h} \right\rceil \quad (1)$$

where h is the desired bin width. Second uses square root relationship to the number of data points, n :

$$k = \lceil \sqrt{n} \rceil \quad (2)$$

Last one illustrates Sturges' formula:

$$k = \lceil \log_2(n) + 1 \rceil \quad (3)$$

As choosing appropriate bin sizes for histogram is problematic, an alternative method for data distribution exploration is called kernel density estimation (KDE). It also approximates the underlying probability distribution of data set from observed data. Simplest form of KDE is defined as

$$\hat{f}(\mathbf{x}) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{\mathbf{x} - X_i}{h}\right) \quad (4)$$

where h is the *smoothing parameter* or *bandwidth*, N is the number of samples and K is the kernel window function. Kernel function can be seen as analogous to concept of bin size in histograms. It determines the shape of the window, most common being the Gaussian kernel.[13]

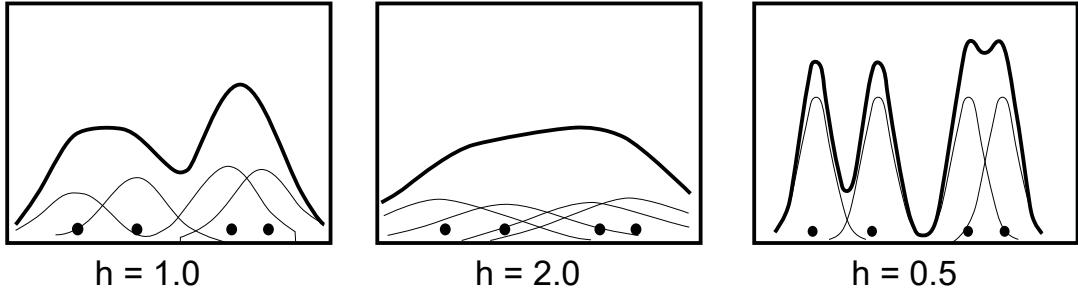


Figure 2: Kernel estimates with different bandwidth.[13]

Figure 2 illustrates effects of varying kernel bandwidth on the same dataset, showing individual kernels. As can be noticed, h too small forms estimation representing series of spikes and h too large washes out all meaningful detail.[13]

2.3 From information visualization to visual analytics

Visualizing information is essential on nearly every field of scientific analytical reasoning. Being the broadest and most flexible input path into brain, utilizing human eyesight provides key advantages over plain numerical data representation models. This allows users to obtain deeper intuition and to make faster analytical reasoning about the observed phenomenon.[14]

In essence the challenge rises from limits of human comprehension in presence of excessive amounts of information. In the 1990s data became increasingly abundant and complex, acquired from multiple heterogeneous sources such as databases, documents, images, audio, video and sensors. Since cognitive abilities of humans remained relatively constant, new methods beyond computer modeling, spreadsheets, business-assessment dashboards and key word searches were clearly required.[15] Understanding and exploiting the powerful pattern-seeking properties of human visual perception system enables design of effective visualizations. On the other hand, ignoring its weaknesses and limitations can lead to misleading or incomprehensive results.[16]

Traditionally field of information visualization has been divided into scientific and data visualization. Former comprises of methods for transforming numerical data into visualizations providing insight to properties of interest or patterns on phenomena. This data can be simulated or gathered from real world. Latter on the other hand structures abstract information via means of visualization structures such as trees or graphs allowing easier understanding of abstract concepts.[16]

Multiple aspects of modern information technology oriented world have laid foundation to formation of field of visual analytics, as shown in Figure 3. Advances in computer performance and explosion in data amount and variety are to be considered the primary driving forces for developing new ways of processing, modeling and visualizing data.[16]

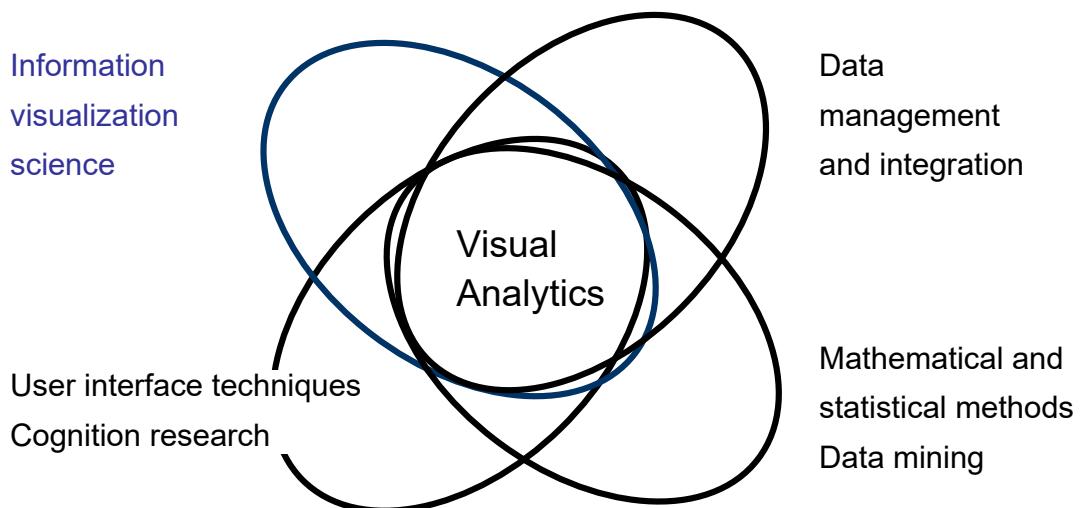


Figure 3: Building blocks of visual analytics [16]

In their 2005 article commonly considered to be the starting point for the field, Thomas and Cook define visual analytics as *the science of analytical reasoning facilitated by interactive visual interfaces*.[17] In their vision cognitive system for visual analytics is formed by two sides: human visual system, flexible pattern finder with adaptive decision-making mechanism and information resources and processing power of a computer with internet connection. Interactive visualization defines the interface between these two sides. By improving this interface, performance

of the entire system could increase substantially, enabling novel discoveries and insights imperceptible with traditional methods.[15, 18] Simply put, objective for visual analytics is to develop tools facilitating the cognitive process of turning large data sets into knowledge via means of data mining and interactive visualization, as represented in Figure 4.[16]

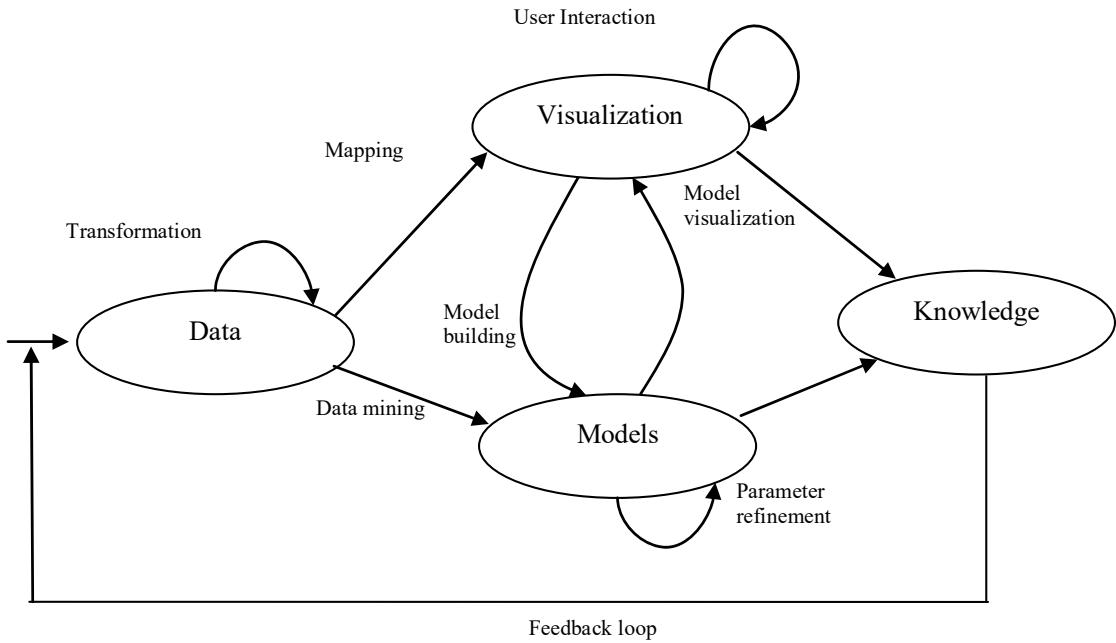


Figure 4: Visual analytics process [16]

In addition to being capable of handling large data, visual analytics tools should be able to handle aforementioned heterogeneous complex data. This requirement arises from the assumption that analysis of large and complex data may lead to discoveries of new previously unknown knowledge. This assumption holds if the tool enables user to discover all included information.[15] The complex data of differing quality requires preprocessing such as filtering, noise removal and transforming to different format. After such processing visualization should be able to highlight important features, commonalities and anomalies enabling effortless data exploration and achievement of new knowledge.[16]

Figure 5 illustrates the assumed relationship between sophistication of visual analytics tools and potential insight achievable, for data of different complexity. Research in visual analytics for complex data aims at increasing the slope of the assumed functional relationship.[15]

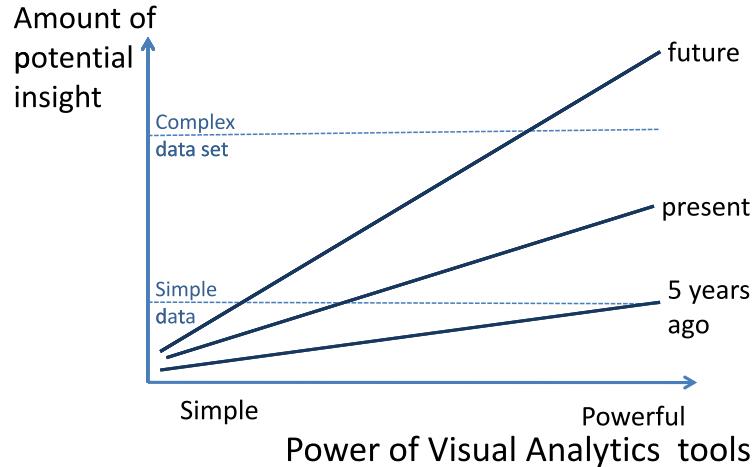


Figure 5: Assumed relationship between sophistication of visual analytics tools and potential insight.[15]

Following these design principles, successful visual analytics software should feature:

- Visualization of huge quantities of data in compact and efficient manner.
- Formation of different views of data, abstractions and emergent features.
- Finding patterns: relationships, anomalies and similarities.
- Detection of erroneous data, providing quality control.
- Understanding of large-scale and small-scale features of data, causal connections.
- Information extraction and distillation leading to hypothesis formation.

[16, 18]

2.4 Human-computer interaction and real-time rendering

Performance is key factor when designing interactive user interfaces. Waiting for something to happen after giving input to the software is frustrating; The more immediate the response, the more acceptable the interaction experience becomes.[14] This latency directly affects performance of the human-computer interface and thus the entire cognitive process described in previous chapter.

This impact of feedback latency is especially true for real-time interfaces such as the one discussed in this work. Visualization orientation modifications such as scaling, rotating and translating require appropriate hand-eye coordination and thus consistent performance. Most commonly accepted metric for meeting this criterion is the fusion flicker frequency. This value describes the minimum refresh rate for human to perceive stream of images as continuous motion. It is affected by multiple factors such as illumination intensity or age and fatigue of the observer.

Application updating the screen once a second, or running at 1 fps would not provide much sense of interactivity. At 6 fps this sensation starts to increase while

around 15 fps is the typical threshold for most people to experience the application as "real-time".[19] 24 fps is commonly used refresh rate in movie industry and in modern computer games(particularly requiring the aforementioned hand-eye coordination) refresh rates from 30 fps to as high as 144 fps are being used.

To achieve the goal of rendering images megapixels in size tens of times in second, capabilities of modern computing hardware are to be exploited. Ubiquitous in modern computing platforms ranging from mobile devices to laptops and PCs, graphics processing unit(GPU) is the tool intended precisely for the task.[20]

While rate of progress in single-core computing performance has decelerated, development of microprocessors has been directed towards parallelization. In the future amount of cores in computing systems will keep on increasing instead of single-core performance.[21] Graphics processing units represent current peak of the trend featuring immersive amount of cores and simultaneously running threads while still remaining cost-effective. Figure 6 illustrates hierarchical parallel GPU computing model adopted by NVIDIA's CUDA framework.[20]

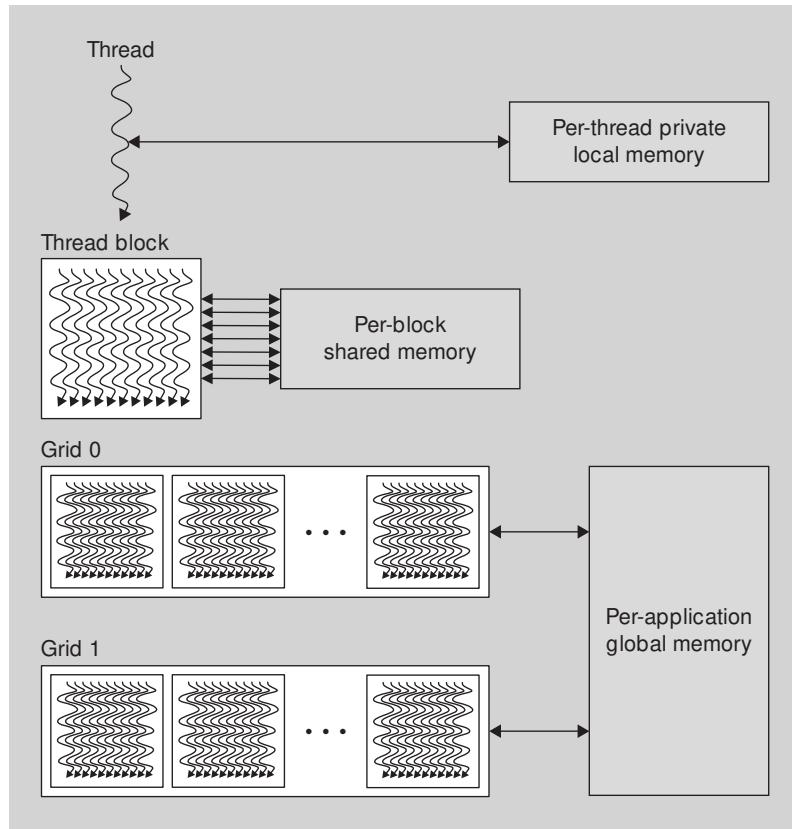


Figure 6: The CUDA computing model[20]

Driving force for rapid GPU development since introduction of 3Dfx Voodoo 1 in 1996 continues to be complex 3D scene rendering tasks demanded by modern day video games.[19, 20] On early days GPU was a fixed-function processor tightly coupled to 3D graphics rendering pipeline. Since then increasingly many of the pipeline stages have become programmable, evolving GPU into massively parallel

general-purpose processor with enormous arithmetic capacity.[20, 21] This ability to perform tasks outside graphics applications such as scientific computing is referred as GPGPU (General Purpose computing on a GPU). GPGPU applications are generally implemented using specialized frameworks or APIs such as NVIDIA’s CUDA or open-source OpenCL.[22]

This effort on pipeline programmability has concentrated mainly on two stages: the vertex stage and the fragment stage. Vertex stage is performed for every vertex of the primitives to be rasterized. Fragment stage on the other hand is performed for every pixel or subpixel produced by rasterization stage in between. Previously these two stages would feature predefined functionality such as transformations or lighting calculations. In programmable pipeline these fixed-function operations have been replaced by *vertex program* and *fragment program*. Figure 7 outlines the modern graphics pipeline. The vertex and fragment processor stages are both programmable by the user.[22]

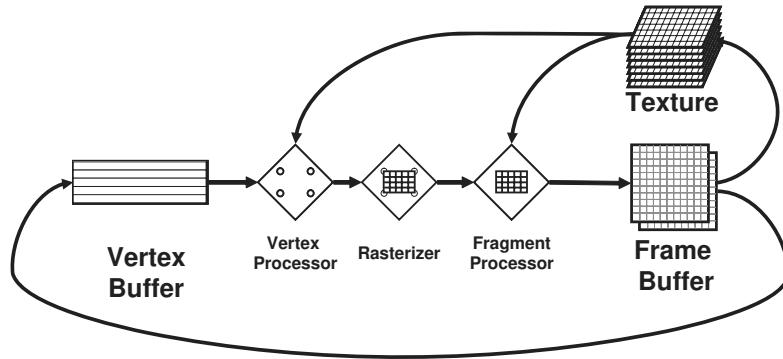


Figure 7: The modern graphics hardware pipeline.[22]

3 Research on experimental visualization tool

This section covers desired features and implementation details of the experimental software tool alongside with description of the data used. As stated earlier, this research has been done in request of Kone Oyj with an intention for creating a tool for horizontal people flow analytics. Therefore specification for the features and data has been provided by the company. Primary focus in the scope of this thesis is GPU acceleration of the KDE heatmap visualization.

3.1 People flow data

People flow data used by the tool can be captured from real world or generated by a simulation software. The data consists of paths on a 2D plane labeled with unique identifiers. It has been formatted into multiple sequential CSV files with each line representing a data point formatted into columns as follows: time stamp, the unique path id, position X coordinate and position Y coordinate. These files could also contain special lines for intermediate identifiers to separate the data into distinctive blocks for filtering purposes. Table 1 shows a sample of example CSV file.

Timestamp	Path Id	Pos X	Pos Y
1464321558429	416443	-4557.1177	361.38525
1464321558429	416442	-4998.405	999.2334
1464321558529	416443	-4581.088	414.57812
1464321558529	416442	-5030.461	1030.8047
1464321558629	416443	-4623.51	468.22412

Table 1: Sample of people flow tracking data CSV file

Each CSV file corresponds to single area (or floor) and is coupled with corresponding metadata and possible background image file. Metadata file specifies properties of data represented in the CSV file such as units being used, positioning information or specification for intermediate identifiers. Background images are stored into .svg, .jpg or .png files. In addition to these floor-specific files an XML file is present, specifying areas, their names and respective filenames in format represented in Listing 1. To ease the usage and prevent file relation breakage caused by manual file transfer, forementioned files are contained in single .zip package.

Listing 1: XML specification file example

```

1 <?xml version="1.0"?>
2 <HeatMap>
3   <Area Name="Floor 1" Background="floor_1.svg" Metadata=
      md_floor_1.csv" Blobs="blobs_floor_1.csv"/>
4   <Area Name="Floor 2" Background="floor_2.svg" Metadata=
      md_floor_2.csv" Blobs="blobs_floor_2.csv"/>
5 </HeatMap>
```

When simulating people flow using an agent based model, two different simulation methods are used: serial and profile. In case of serial simulation, simulation is run in multiple steps. During each step, people flow intensity is kept constant but when proceeding to next step simulation is reset and flow intensity is increased cumulatively. Profile simulation is also run multiple times, but instead of constant flow intensity a predefined profile is used. This profile consists of time slices, each corresponding to a flow intensity value. The profile is run for the consequent steps unmodified, providing sufficiently data for detecting anomalies. For analyzing results of different simulation steps, the point data can be divided using the aforementioned intermediate identifiers.

Data set used in this work is real-word data captured from Kone Building lobby hall at Keilasatama 3, Espoo on time period between 26.5.2016 and 13.7.2016. It consists of 17 732 916 data points in 94 642 paths sample with 100ms interval.

3.2 Tool features

The complete tool was specified to be implemented as a stand-alone desktop application featuring

- Ability to handle simulated or sensor data using unified input format.
- Ability to handle multiple areas within the input data package.
- Real-time KDE heatmap visualization with background image and viewport reorientation capabilities.
- Date and date range selection in the case of real-world data.
- Simulation step selection in the case of simulated data.
- Filtering tools:
 - Time of day range filtering.
 - Area filtering: paths going through selected area.
 - Speed filtering, according to average speed of a path.
 - Multiarea flow filtering: paths going from one area to another through arbitrary many intermediate areas.
 - "Countline" tool: count paths going through a selected line to forward/backward direction.
- Metrics calculation:
 - Total/min/max/average number of agents for each selected area.
 - Min/max/average dwell time of agents for each selected area.
 - Min/max/average agent density of agents for each selected area.
 - Number of agent forward/backward crossings for a countline.

- Number of agents through multiarea sequence.
- Metrics output into CSV file.

3.3 Implementation and testing platform

Experimental tool with limited functionality was implemented, featuring raw CSV file support, real-time heatmap visualization, and single area and time of day filtering. Implementation was done in C++ programming language for maximum performance and existing library support. For graphics acceleration, OpenGL 3.3 was used. SFML was used for windowing, OpenGL context creation and real-time user input handling. For CPU-side parallelization OpenMP framework was used. Software was run on a desktop PC equipped with Intel Core i7-6700K CPU running at 4.00GHz, NVIDIA GeForce GTX 970 running at 1177MHz and 32Gb of system memory running 64-bit Windows 10 operating system.

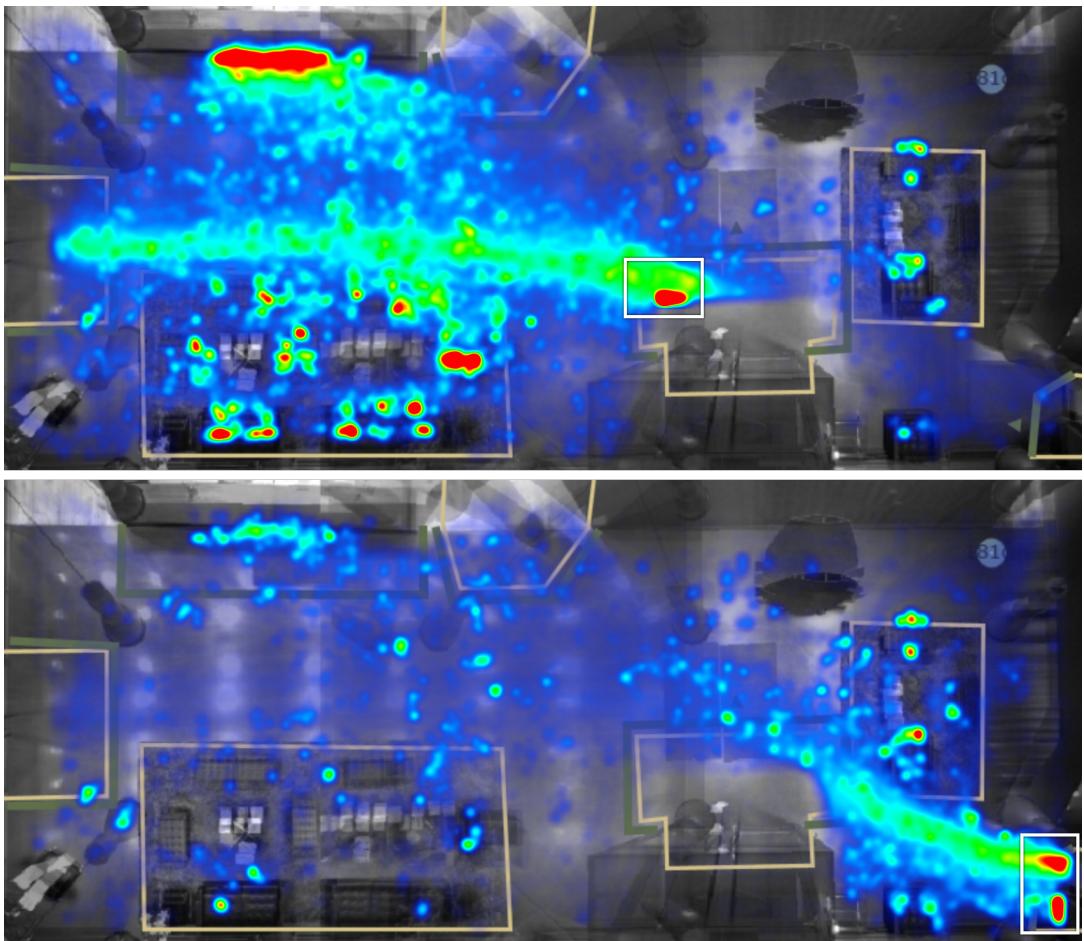


Figure 8: Heatmap visualization with 2 different area filters

As stated, primary subject for research was acceleration of KDE visualization by using the GPU. Therefore the convolution operation was implemented for both CPU and GPU for performance measurements and comparison. The algorithm proceeds

as follows: first, filtered 2D point map is formed according to current viewport and filtering. For KDE, this map is convolved with discretized 2D gaussian kernel. After this, the map is colorized and merged with background image, producing final heatmap, as seen in Figure 8 Since 2D gaussian convolution is separable (horizontal and vertical component can be convolved separately for identical result), calculation of 1-dimensional kernel is sufficient.

Gaussian kernel window function used in the software is defined by

$$K(u, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{u^2}{2\sigma^2}\right) \quad (5)$$

where σ is the standard deviation.

Standard deviation σ is required to be chosen according to blur radius measured in pixels. Blur radius in turn is determined by ratio of simulation area dimensions and resolution of the viewport to estimate size of an average person. Since $K(u, \sigma)$ is defined on the range $(-\infty, \infty)$, it is required to be limited for the discrete convolution to be computationally feasible. Limits should be chosen as small as possible without causing visible artifacts at the edge of the kernel. For the experimental software limits were determined by requiring 99% of integral of the kernel function to lie inside the limits.

That is, solution by respect to a of

$$\int_{-r}^r K(u, ar) du = 0.99 \Rightarrow a \approx 0.388224 \quad (6)$$

where r is blur radius and a is constant multiplier for converting r to σ .

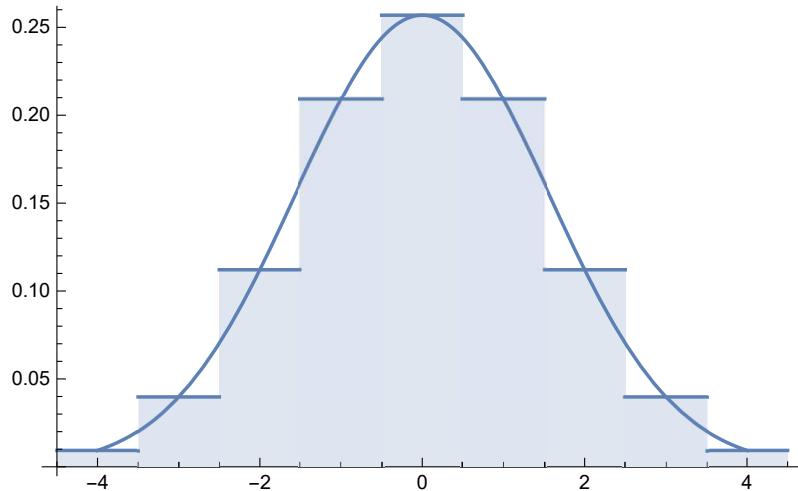


Figure 9: Example of discretized kernel with blur radius $r = 4$

4 Results

Processing times of point map filtering and heatmap convolution and colorization stages were measured. Heatmap rendering (convolution and colorization) was measured on CPU and GPU methods for comparison purposes. Point map filtering was done only on CPU, so no comparison data is available. 30 fps was selected as a limit for the interface to be considered as real-time. That equals to $1000\text{ ms} / 30 = 33.33\text{ ms}$ time budget for each frame. Both stages are required to fit within this budget, but in case majority of frames only rendering is required as re-filtering is done only when new filter was introduced: for example, when user selects a new area of interest.

Rendering was done on two different resolutions: 500×500 and 1000×1000 . Figures 11 and 12 show comparison of CPU and GPU performance as a function of increasing blur radius(render window size in pixels). Figures 13 and 14 illustrate the relative increase in performance when using GPU computation instead of CPU. Time required for point map filtering in respect to number of paths passing the filter is shown in Figure 10.

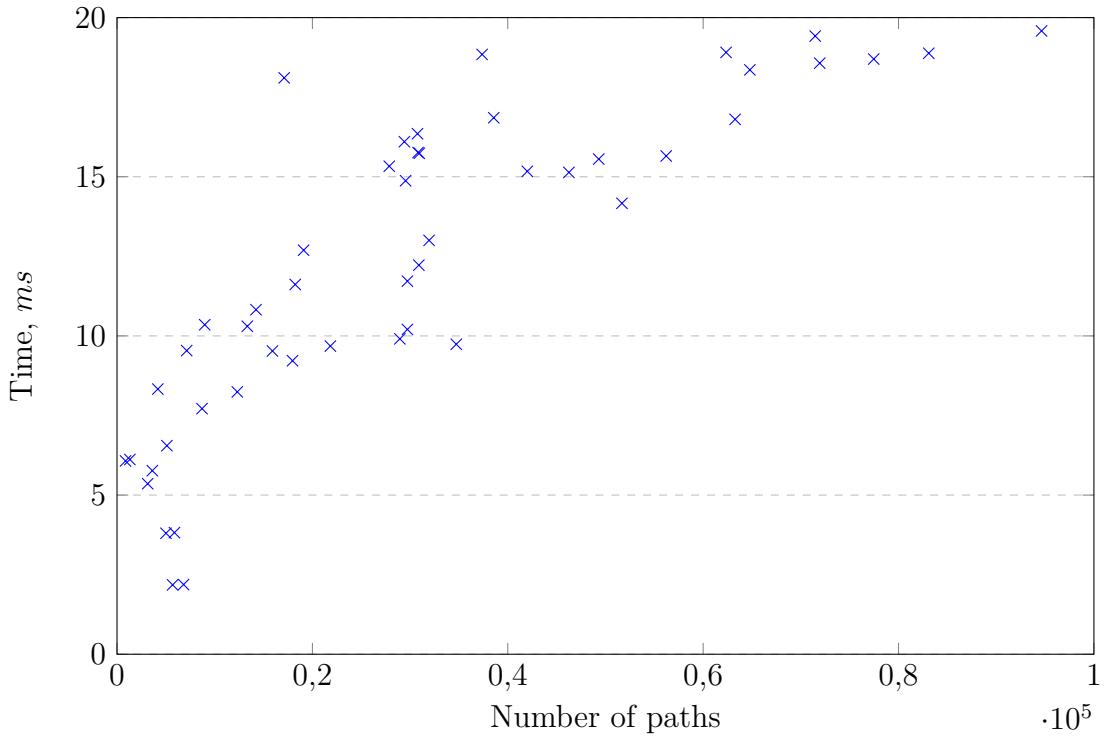


Figure 10: Filtered point map processing time

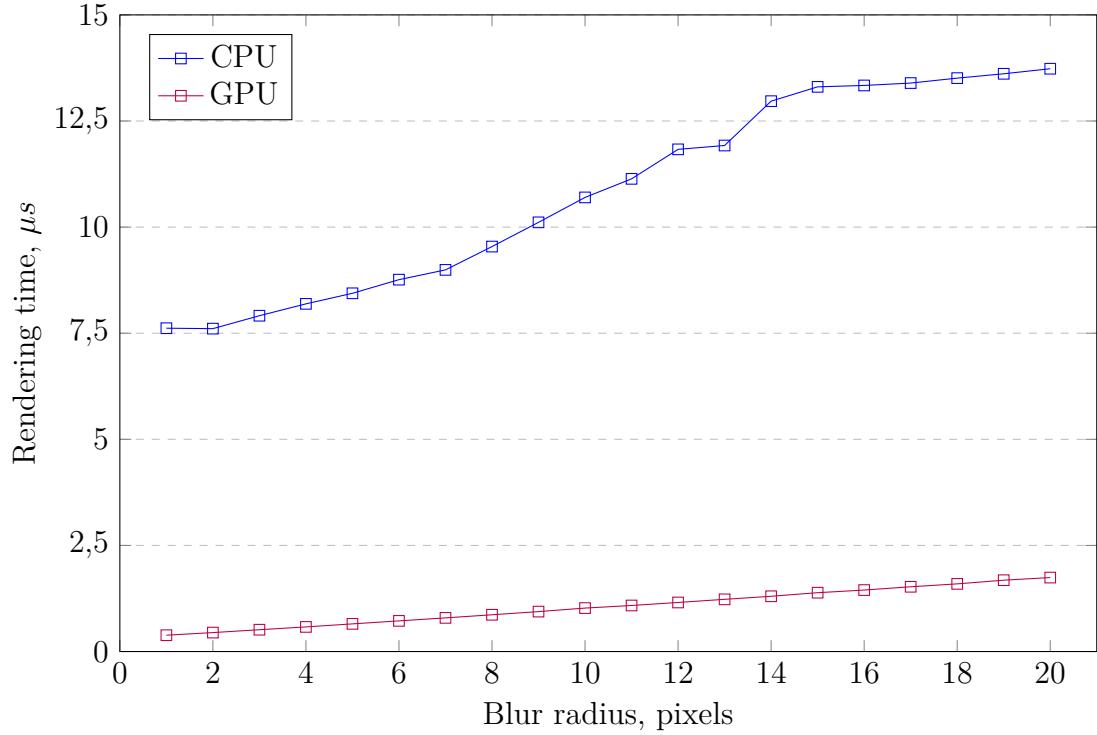


Figure 11: Heatmap rendering time, 500×500 resolution

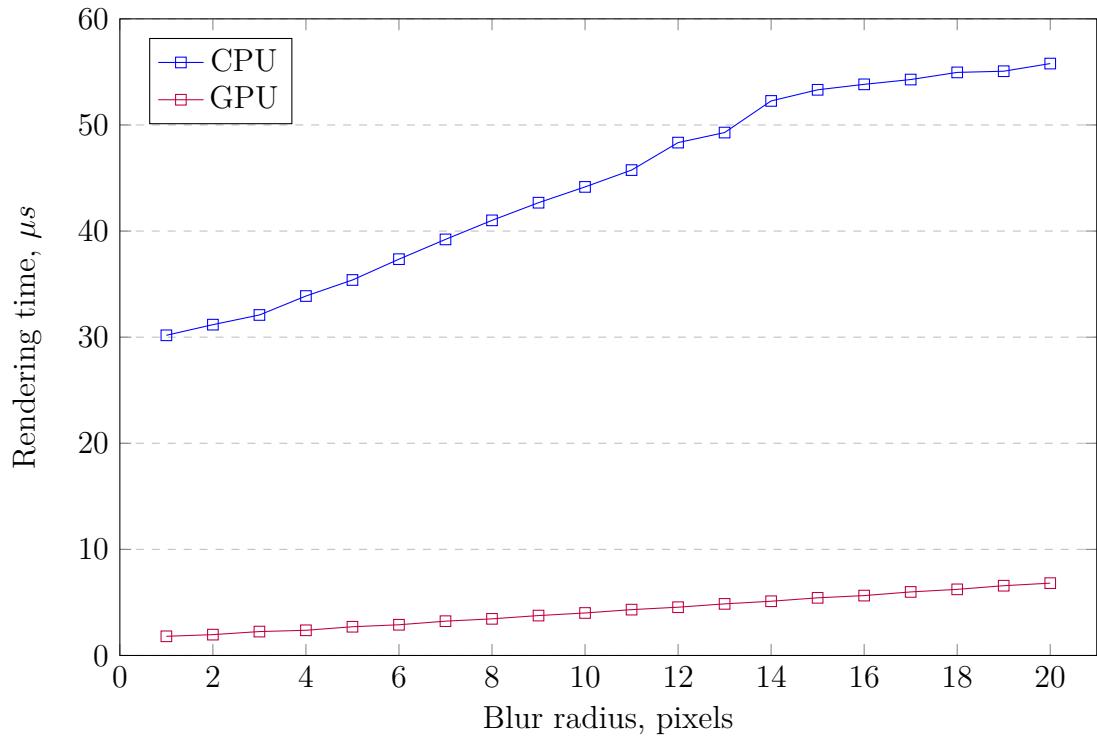


Figure 12: Heatmap rendering time, 1000×1000 resolution

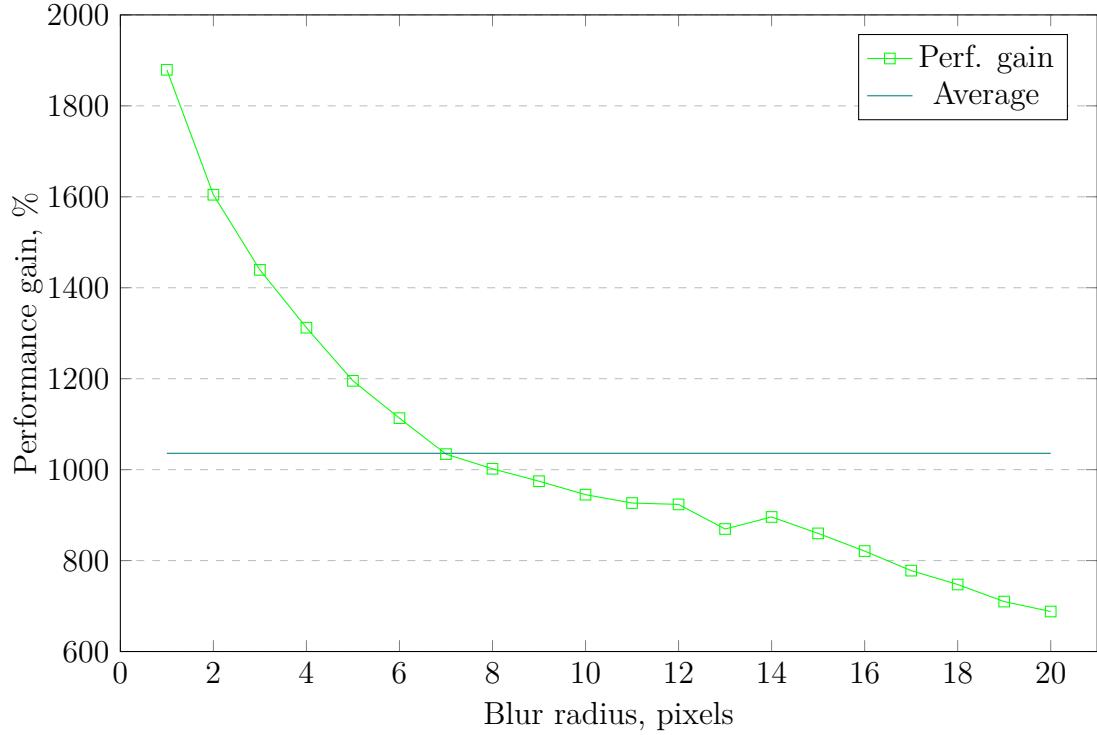


Figure 13: Rendering performance gain, 500×500 resolution

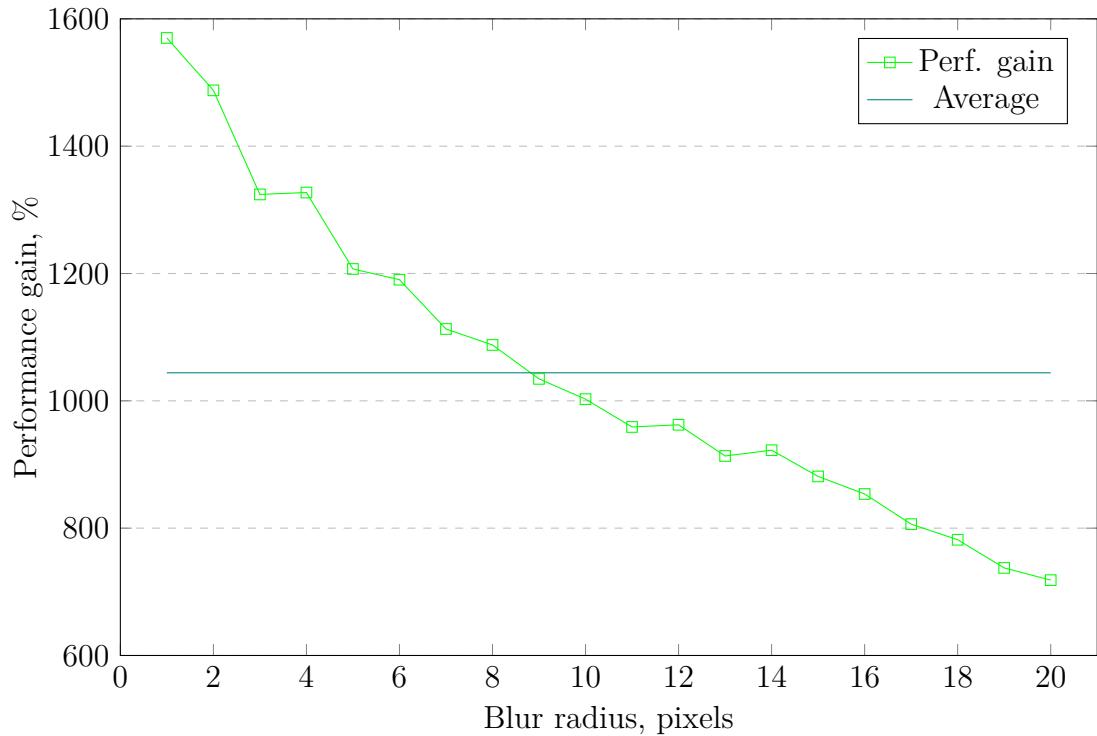


Figure 14: Rendering performance gain, 1000×1000 resolution

5 Conclusion and further development

Rendering time measurements show that with sufficiently powerful computing hardware using CPU can be plausible option for rendering when viewport of very moderate resolution. As computing requirements increase linearly with the number of pixels CPU rendering no longer can meet the 33.33 ms time budget when increased resolution is used, as is shown in Figure 12.

As clearly can be seen from Figures 13 and 14, GPU rendering can increase performance by an order of magnitude, allowing real-time requirement to be met with even higher resolutions and kernel function sizes. Decreasing relative performance gain can be explained with the method of computing the kernel: in CPU method the kernel is precalculated to avoid expensive evaluations when iteratively processing the image. When using GPU, kernel is calculated for each pixel, vastly increasing the computational cost.

Point map processing time raises some reasons for concern as time required rises close to the time budget with larger amount of paths, as shown in Figure 10. Furthermore time data exhibits high variance with equal number of paths, making predicting of filtering time significantly harder. Considering both the filtering and rendering time, if any larger dataset or less powerful hardware needs to be used, real-time performance can no longer be provided.

Improving filtering stage performance would be the next subject for research. Fortunately, path filtering as a computational task is suitable for massive parallelization since each filtering operation is independent from others. Thus, extensive performance gains could potentially be achieved by using GPGPU capabilities of modern GPUs, as described in chapter 2.4.

GPU filtering could improve the performance only up to limit though, as computational cost of filtering and system memory requirement increase as the size of the dataset increases. If handling of significantly larger amount of data is desired, advanced preprocessing methods such as the ones described in [2] and [23] are required. Processing and modeling allow data abstraction forming, compressing it in size and complexity, decreasing aforementioned system requirements.

As a conclusion, objective for the thesis can be considered met since the experimental software demonstrates that computationally most demanding stages of the software can be processed in real-time with current PC hardware. Rest of the features specified in chapter 3.2 do not contribute for significant increase in real-time computational load. Overall the thesis outlines number of paths for improving over the experimental software.

References

- [1] Ali, Saad, Ko Nishino, Dinesh Manocha, and Mubarak Shah: *Modeling, simulation and visual analysis of crowds: a multidisciplinary perspective*. Springer, 2013.
- [2] Ali, S. and M. Shah: *A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis*. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, June 2007.
- [3] Scovanner, P. and M. F. Tappen: *Learning pedestrian dynamics from the real world*. In *2009 IEEE 12th International Conference on Computer Vision*, pages 381–388, Sept 2009.
- [4] LaValle, S. M.: *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at <http://planning.cs.uiuc.edu/>.
- [5] Maim, J., B. Yersin, J. Pettre, and D. Thalmann: *Yaq: An architecture for real-time navigation and rendering of varied crowds*. IEEE Computer Graphics and Applications, 29(4):44–53, July 2009, ISSN 0272-1716.
- [6] Van Den Berg, J., S.J. Guy, M. Lin, and D. Manocha: *Reciprocal n-body collision avoidance*. Springer Tracts in Advanced Robotics, 70(STAR):3–19, 2011. https://www.scopus.com/inward/record.uri?eid=2-s2.0-79957981823&doi=10.1007%2f978-3-642-19457-3_1&partnerID=40&md5=665f21307fdfe3c245e9bef99e6dae56, cited By 175.
- [7] Hughes, R.L.: *A continuum theory for the flow of pedestrians*. Transportation Research Part B: Methodological, 36(6):507–535, 2002. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0036643318&doi=10.1016%2fS0191-2615%2801%2900015-7&partnerID=40&md5=8717345c4af0ee78e30ed75bc25666b8>, cited By 412.
- [8] Pettre, J., H. Grillon, and D. Thalmann: *Crowds of moving objects: Navigation planning and simulation*. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3062–3067, April 2007.
- [9] Peacock, Richard D, Jason D Averill, et al.: *Pedestrian and evacuation dynamics*. Springer Science & Business Media, 2011.
- [10] Heliövaara, Simo: *Pedestrian behavior in evacuations – simulation models and experiments; käyttäytyminen evakuointilanteissa – simulointimallit ja kokeelliset tulokset*, 2014, ISBN 978-952-60-5684-5 (ELECTRONIC); 978-952-60-5683-8 (PRINTED). ISSN 1799-4942 (ELECTRONIC); 1799-4934 (PRINTED); 1799-4934 (ISSN-L). <http://urn.fi/URN:ISBN:978-952-60-5684-5>.

- [11] Korhonen, P., J M. Kuusinen, M. Piironen, and T. Voutilainen: *Improving the value of buildings through a smart lobby approach*. In *CTBUH 2016 Shenzhen, Guangzhou, Hong Kong Conference*, 2016.
- [12] Pearson, Karl: *Contributions to the mathematical theory of evolution. ii. skew variation in homogeneous material*. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 186:343–414, 1895, ISSN 0264-3820. <http://rsta.royalsocietypublishing.org/content/186/343>.
- [13] Maciejewski, R.: *Data representations, transformations, and statistics for visual reasoning*. Morgan & Claypool, San Rafael, Calif., 2011, ISBN 9781608456260.
- [14] Yoon, S.: *Real-time massive model rendering*. Morgan & Claypool, San Rafael, Calif., 2008, ISBN 9781598297935.
- [15] Dill, J., R. Earnshaw, D. Kasik, J. Vince, and P.C. Wong: *Expanding the frontiers of visual analytics and visualization*. Springer cop., 2012, ISBN 9781447128045.
- [16] Järvinen, P., K. Puolamäki, P. Siltanen, and M. Ylikerälä: *Visual analytics final report*. Technical report, VTT Technical Research Centre of Finland, March 2009, ISBN 978-951-38-7178-9.
- [17] Thomas, J.J. and K.A. Cook: *A visual analytics agenda*. IEEE Computer Graphics and Applications, 26(1):10–13, 2006.
- [18] Ware, C.: *Information visualization : perception for design*. Morgan Kaufmann, 2012, ISBN 9780123814654.
- [19] Akenine-Möller, Tomas, Eric Haines, and Naty Hoffman: *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., Natick, MA, USA, 2008, ISBN 987-1-56881-424-7.
- [20] Nickolls, J. and W.J. Dally: *The gpu computing era*. IEEE Micro, 30(2):56–69, 2010. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-77951154340&doi=10.1109/MM.2010.41&partnerID=40&md5=fc92b7a64c80e80165ab60042c6e300b>, cited By 385.
- [21] Owens, J.D., M. Houston, D. Luebke, S. Green, J.E. Stone, and J.C. Phillips: *Gpu computing*. Proceedings of the IEEE, 96(5):879–899, 2008. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-49049088756&doi=10.1109/JPROC.2008.917757&partnerID=40&md5=7b7741b155de6df0f57393863a2496f5>, cited By 931.
- [22] Owens, J.D., D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A.E. Lefohn, and T.J. Purcell: *A survey of general-purpose computation on graphics hardware*. Computer Graphics Forum, 26(1):80–113, 2007. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-34547262121&doi=10.1111/j.1467-8659.2007.01139.x&partnerID=40&md5=7b7741b155de6df0f57393863a2496f5>

0-33947588048&doi=10.1111%2fj.1467-8659.2007.01012.x&partnerID=40&md5=8b41e1ad4af0990d719b1ed666b682fa, cited By 1012.

- [23] Andrienko, G., N. Andrienko, P. Bak, D. Keim, and S. Wrobel: *Visual Analytics of Movement*. Morgan & Claypool, 2013, ISBN 9781598297935.

A Finnish summary - suomenkielinen tiivistelmä

A.1 Johdanto

Visualisaatiot, kuten kaaviot, kuvaajat ja diagrammit ovat tuttuja useimmille. Niitä käytetään laajalti opetuksessa ja tieteessä niiden oppimista tukevien ja dataa havainnolistaisten ominaisuuksien ansiosta. Viime aikoina datan visualointikeinot ovat laajentuneet interaktiivisten esitystapojen ja työkalujen kehittyessä. Visuaalinen analyysi tieteenalana keskittyy ihmisen cognitiivisten kykyjen, kuten työmuistin, rakenneidenhahmotuskyvyn sekä päättöksentekokyvyn laajentamiseen ihmisiän teokkaan kuvankäsittelyjärjestelmän avulla.

Suurien tietomäärien muuttaminen ihmisen ymmärtämään muotoon vaatii huomattavia määriä laskentaresursseja. Perinteisesti käsiteltävien aineistojen kokoa on täytynyt rajata, tai prosessi vei liikaa aikaa. Reaalialkaisia datavisualisaatioappilaitoita on voitu luoda vain pienille tietomääritteille rajoittaen näiden hyödyllisyyttä huomattavasti. Laskentatehon lisääntyessä sekä prosessointiarkkitehtuurien rinnakkaistuessa suurienkin datamäärien reaalialkarenderöinti on mahdollistunut.

Tämä opinnäytettyö on tehty Kone Oyj:lle ja se tutkii mahdollisuutta kehittää reaalialkainen työkalu ihmisvirtadatan visuaalista analyysiä varten. Erittyisesti tutkimus kohdistuu ydinestimaatiovisualisaation eli ns. "lämpökartan" renderöintiin nykyaisista tietokoneista löytyvästä grafiikkakiihdystä käyttäen. Kokeellinen sovellus implementoitiin ja suorituskykymittaukset sekä lämpökartan renderöinnistä ettiä datan suodatuksesta tehtiin.

A.2 Ihmisvirtojen analysointi ja rakennusten suunnittelu

Ihmisvirtojen ymmärtäminen on olennaista monilla tieteen aloilla, kuten koneopimisessa, tietokonegrafiikassa sekä evakuointidynamiikassa. Yksilöiden sekä pienien ja isojoen ryhmien käyttäytymisen tarkka analyysi ja synteesi mahdollistaa sovelluksia valvonnassa, sosiaalipsykologiassa sekä rakennusten ja julkisten alueiden suunnittelussa.

Rakennusten sekä ihmisvirtoihin vaikuttavaa varustelua, kuten hissejä, liukupuertaita, automaattiovia ja informaatiotauluja suunniteltaessa on otettava monia käyttökokemukseen vaikuttavia tekijöitä huomioon. Rakennuksella on usein myös useita käyttäjäkuntia, kuten työntekijöitä, tavarantoimittajia sekä liikuntarajoitteisia henkilöitä. Jotta kaikille voidaan tarjota yhdenvertainen käyttökokemus, on tarkasteltava koko matkaa sisääntulolta määränpäähän. Tämä vaatii tietoa ihmisvirtojen risteymistä, pullonkauloista sekä käyttäjien toimista.

Tällainen monia muuttujia vaativa suunnittelutyö tarvitsee avukseen edistyneitä työkaluja. Simulaatiotyökalulla voidaan syntetisoida ihmisvirtoja eri pohjapiirustuksilla. Syötteenä simulaattorille annetaan ihmiliikenteen intensiteettiprofili, joka kertoo rakennukseen sisääntulevien, siitä lähtevien sekä kerrostenväillä liikkuvien ihmisten määrän aikaikunnoittain. Uusille rakennuksille simulointi tehdään käytäen yleispäteviä profileja, mutta modernisointitarkoitukseen tarvitaan todellinen, rakennuksesta mitattu profili.

A.3 Visuaalinen analyysi ja reaalialikarenderöinti

Koska näkö on laajin ja mukautuvin kanava ihmisen aivoihin, sen hyödyntäminen analyyttisessä päättelyssä tarjoaa monia etuja pelkkien lukujen sijaan. Visualisaatioiden avulla voidaan saavuttaa syvempiä näkemyksiä huomattavasti nopeammin. Kuitenkin 1990-luvulla tietolähteiden määrä ja monipuolisus kasvoivat niin nopeasti, ettei ihmisen vakiona pysyvä käsittelykyky enää riittänyt perinteisiä menetelmiä käyttäen. Uudet keinot olivat selvästi tarpeen.

Vuonna 2005 Thomas ja Cook määritteliävät artikkelissaan kahdesta osasta koostuvan visuaalista analyysiä hyödyntävän kognitiivisen prosessin. Tässä mallissa toisella puolella oli ihmisen rakenteiden havaitsemiseen erikoistunut näköhermosto sekä adaptiivinen päätöksentekokyky ja toisella tietokoneen laskentateho sekä internetin tietolähteet. Nämä puolet yhdistää interaktiivinen visualisaatorajapinta. Tämän rajapinnan suorituskykyä parantaen koko kognitiivisen järjestelmän suorituskyky paranisi.

Suorituskyky on tärkeää interaktiivisia sovelluksia luodessa: syötteeseen reagoimisen odottelu on turhauttavaa. Mitä nopeammin applikaatio reagoi, sitä parempi on käyttökokemus. Tämä latenssi vaikuttaa suoraan viime kappaleessa määritellyn kognitiivisen prosessin suorituskykyyn. Esimerkiksi visuaalisaation orientaation vaikuttavat syötteet, kuten siirtäminen, skaalaaminen tai pyörittäminen vaativat toimivaa silmä-käsi-koordinaatiota ja täten tasalaatuista suorituskykyä. Yleisesti käytössä oleva raja-arvo on virkistystaajuus, jossa ihminen kokee yksittäisten kuvien sarjan jatkuvana liikkeenä. Tyypillisesti tämä arvo on n. 15 kuvala sekunnissa. Nykyisin esimerkiksi juuri em. silmä-käsi-koordinaatiota vaativissa tietokonepeleissä käytetään 30-144 Hz virkistystaajuksia.

Jotta megapikslien kokoisten kuvien reaalialikarenderöinti olisi mahdollista, nykytietokoneiden laskentaresursseja tulee hyödyntää. Käytännössä jokaisesta laskenta-alustasta mobiililaitteista pöytäkoneisiin löytyy grafiikkasuoritin(GPU) -massiivisesti rinnakkaisesti toteutettu prosessori, joka on tarkoitettu nimenomaan tähän tehtävään. Vaikka alun perin sekä vielä nykypäivänäkin suurin motivaatio grafiikkasuoritinten kehitykseen tulee tietokonepeleistä, on niitä alettu lisääntyvissä määrin käyttää myös tieteelliseen laskentaan ns. GPGPU-rajapintojen kautta, jotka mahdollistavat minkä tahansa laskennan suorittamisen grafiikkasuorittimella.

A.4 Kokeellinen työkalu ja tulokset

Työkalun käyttämä data koostuu sekventiaalisesti CSV-tiedostoon tallennetuista datapisteistä, jotka muodostavat polkuja. Tiedoston jokainen rivi sisältää aikaleiman, uniikin polkutunniston sekä X- ja Y-koordinaatit. Tässä työssä käytettävä ihmisvirtodata on kerätty Keilasatama 3:ssa sijaitsevan Kone Building:n aulasta aikavälillä 26.5.2016-13.7.2016. Se koostuu 94 642 polkuun kuuluvasta 17 732 916 datapistestä, joiden näyteväli on 100 ms.

Kokeellinen työkalun ominaisuuksiin kuuluu tuki CSV-tiedostoformaatiille, reaalialkainen lämpökarttavisualisaatio sekä suodatus yksittäisen alueen ja päivänajan suhteen. Implementaatio tehtiin käyttäen C++-ohjelmointikieltä, OpenGL-

grafiikkarajapintaa sekä OpenMP-rinnakkaistusratapihantaa. Kokeet tehtiin PC-tietokoneella, joka oli varustettu Intel Core i7-6700K -suorittimella, NVIDIA GeForce GTX 970 - näytönohjaimella, 32Gb keskusmuistia sekä Windows 10 - käyttöjärjestelmällä.

Sovelluksen laskentaprosessi on seuraavanlainen: aina suodatuksen muuttuessa lasketaan kaksiulotteinen pistekartta, joka sisältää suodatuksen läpäisseiden polkujen datapisteet. Tämän jälkeen kartta konvoloidaan käyttäen gaussin kellokäyrään perustuvaa ydinfunktiota. Koska kaksiulotteinen konvoluutio kyseistä funktiota käyttäen on separaituva, voidaan laskenta tehdä kahdessa vaiheessa yksiulotteista näytteistettyä funktiota käyttäen. Lopuksi saatu kuva väritetään sekä yhdistetään taustaan lopullisen visualisaation muodostamiseksi. Konvolointivaihe tehtiin sekä keskus- että grafiikkasuorittimella vertailun tekemiseksi. Sekä suodatus- että renderointivaiheiden suorituskyky mitattiin.

A.5 Johtopäätökset ja jatkokehitys

Mittaustuloksista voidaan havaita, että grafiikkasuorittimen hyödyntäminen konvolointivaiheessa paranti sen suorituskykyä suuruusluokalla. Esilaskemalla ydin-funktion arvot myös grafiikkasuorittimella tehdyssä konvoloinnissa voitaisiin suorituskykyä luultavasti parantaa vielä entisestään. Suodatusvaihe puolestaan antaa aihetta huollelle, sillä varsinkin isoilla polkumääriillä suoritusaika nousi lähelle kriittistä rajaa. Tähän vaiheeseen voitaisiin luultavasti käyttää GPGPU-menetelmiä.

Johtopäätökseni voidaan todeta, että tällainen 30 Hz ruudunpäivitysnopeutta käyttävä reaalialaika-applikaatio on mahdollista toteuttaa kymmenien tuhansien polkujen tietoaineistolle grafiikkasuorittimella varustettua PC-laitteistoa käyttäen. Jatkokehitys on tarpeen, mikäli on tarvetta käyttää suurempia tietoaineistoja tai heikompitahoista laitteistoa. Työ esittää pääpiirteittäin monia potentiaalisia jatkokehityssuuntia kokeellisen sovelluksen suorituskyvyn parantamiseksi.