UROP: Code generation for numerical quadrature on parallel multiprocessors for finite element simulations

Abstract

The finite element method is a well establish technique for discretizing partial differential equations. Such equations are formulated by means of discrete multi-linear forms, which, in the context of numerical simulations, are realized into matrices and vectors by means of numerical quadrature. For complex non-linear problems, quadrature has to be performed multiple times within the solution process, hence contributing substantially to the overall computational time.

Utopia is a *C++ embedded domain specific language* which targets parallel finite element simulations. It separates the developer intention from the actual implementation.
The user specifies the multi-linear forms with a high-level code interface which generates an expression template forest (ETF). The ETF can be used by several back-ends for computing the needed quadrature values following different strategies.

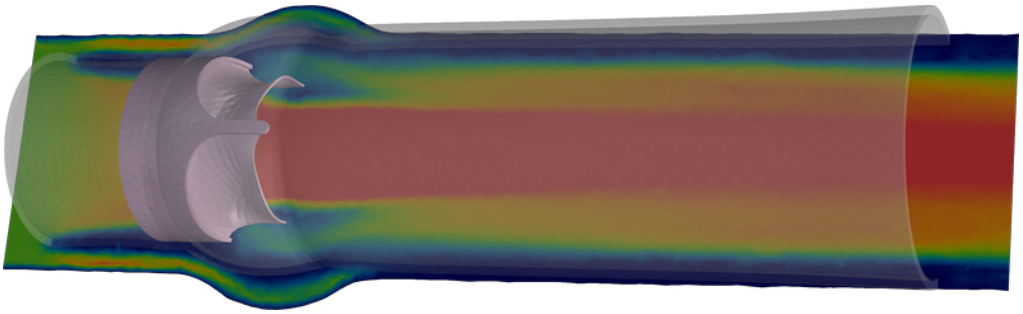
The aim of this project is to explore one of such strategies, code generation targeting acceleration hardware such as *GPU*s and *multi-core CPU*s. Eventually, the student can extend the work in a thesis by focusing on any of the following aspects:

- ETF differentiation. Automatic computation of derivatives using symbolic differentiation or automatic differentiation.
- Support for multi-physics applications using existing code for coupling different models (*e.g.,* Fluid-structure interaction).

Prerequisites

Knowledge of C++ and basic concepts of compilers.

Advisors: Prof. Dr. Rolf Krause, Dr. Patrick Zulian



Simulation of a heart valve

Summer Project Proposal: Extreme Testing of Self Driving Cars

Paolo Tonella, Andrea Stocco
Software Institute@USI
paolo.tonella@usi.ch,andrea.stocco@usi.ch

Autonomous self-driving vehicles are just around the corner. Most major manufacturers including Tesla, GM, Ford, Mercedes, BMW, and Waymo/Google are building and actively testing different types of autonomous vehicles. Recent results show that autonomous cars have become very efficient in practice and have already driven millions of miles without any human intervention.

Architecture of a Self Driving Car

Software systems known as Advanced Driver Assistance Systems (ADAS) are main enablers of self-driving cars. Examples of ADAS include automated steering angle prediction, automatic parking, night vision and collision avoidance systems.

The key component of an autonomous vehicle is the perception module controlled by the underlying Deep Neural Network (DNN). The DNN takes input from different sensors like camera, light detection, ranging sensor, and infrared sensor, which measure the environment, and outputs the steering angle or braking commands necessary to operate the car safely under currently expected conditions (see Figure 1).

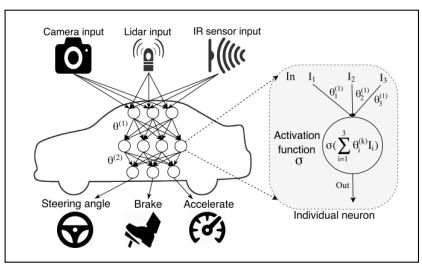


Figure 1: High-Level Architecture of a self-driving car.

On-Road Testing

On-road testing of autonomous cars is typically restricted to a small number of vehicles driven by professional drivers during specific hours on some designated roads with specific speed limits. Such testing is often expensive and time-consuming. It is further impractical to perform a full-fledged on-road vehicle-level testing after every change to self-driving software systems. To ensure safety of self-driving technologies, vehicle-level testing alone is neither enough nor practical. Therefore, it is complemented by testing methods performed on software simulators (see Figure 2).

Simulation-based Testing

Simulation-based testing of autonomous cars is very popular as it allows engineers to run a much larger number of test scenarios compared to vehicle-level testing without being limited by conditions enforced during on-road testing.

As a matter of fact, despite the tremendous progress, DNN-based software, including the one used for autonomous driving, occasionally exhibits incorrect/unexpected corner-case behaviours that can lead to dangerous consequences as – in the worst case – fatal collisions.

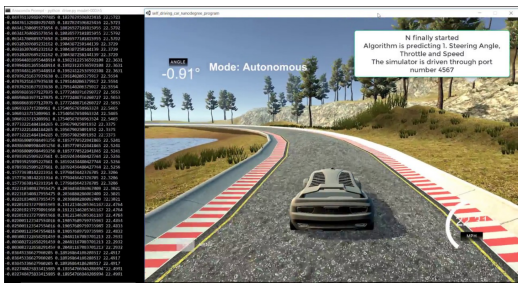


Figure 2: Simulation testing of self-driving cars.

Several such real-world cases have already been reported under rare previously unseen corner cases. For example, the fatal Tesla crash (March 23rd, 2018; Mountain View, California, USA) resulted from a failure to detect a white truck against the bright sky. Since these cars adapt behaviour based on their environment as measured by different sensors (e.g., camera, infrared obstacle detector), the space of possible inputs is extremely large. Thus, unguided simulations are highly unlikely to find erroneous behaviours.

Project Proposal

Students are required to first familiarize with the Udacity self driving cars simulator in order to create complex environments and identify critical test scenarios (i.e., failure revealing conditions) to test the self-driving cars.

Second, students are required to implement a software component to automatically generate *extreme driving environments*, where to test the ability of DNNs to generalize from the training conditions to new, unexpected ones. The student will learn how to train an autonomous driving system and let it operate in a possibly hostile environment. Research will be carried out on the characterization of the limit driving conditions for a trained autonomous driving network and how to re-train it to improve its behaviour.

Further information

The proposed work is part of the Precrime (Self-assessment Oracles for Anticipatory Testing) ERC project. The interested student can find more information about Precrime on the project's website:

https://www.pre-crime.eu/

Parallel Ethereum: Blockchain on Steroids

Blockchain has received much attention in recent years. This immense popularity has raised a number of concerns, scalability of blockchain systems being a common one.

Ethereum, for example, a generic blockchain model that has served as a platform for hundreds of services and applications, executes one transaction at a time.

The goal of this project is to investigate techniques to introduce concurrent transaction execution in the Ethereum Virtual Machine (EVM).
Needless to say, the potential impact of the project is huge.

Requirements: C/C++/Go knowledge, Blockchain/Ethereum knowledge not needed, lots of courage

Teaching assistant: Enrique Fynn

Computational methods for finding the range, roots, and extrema of a function

Given a continuous function f:D->R on a compact interval D, an important problem is to find the range f(D) of possible function values. While it is in general impossible to compute f(D) numerically, several computational methods have been proposed to approximate f(D) by an interval I that contains f(D) and is as close as possible to f(D).

In this UROP, we will investigate a new approach to this problem that extends the "remainder form" proposed by Cornelius and Lohner in 1984 by using local quadratic interpolation of f and its (3k)-th derivatives, where k=1,2,... and exploiting the well-known max norm error estimates for this kind of interpolation.

We will further use this approach in combination with the EVAL algorithm proposed by Mitchell in 1990 to isolate and with the Newton method to actually find the roots of the function f as well as its global minimum and maximum.

Time permitting, we shall also consider the generalization of these ideas to multivariate functions.

Advisor: Prof. Kai Hormann

References

- Cornelius, H. and Lohner, R. (1984). Computing the range of values of real functions with accuracy higher than second order. Computing, 33, pp. 331-347.

- Mitchell, Don P. (1990). Robust ray intersection with interval arithmetic. In: Proceedings on Graphics interface '90, pages 68-74.

Data Analytics System for Improving Mobile Sensing Studies

UROP project proposal at the Università della Svizzera Italiana (USI), Lugano. Switzerland

Background

The widespread adoption of mobile and wearables devices – such as smartphones and smartwatches – allows us to observe human behavior in a fine-grained yet unobtrusive manner. For instance, by collecting data about the position of a person – using the GPS receiver and other location sensors commonly available on smartphones – it is possible to reconstruct the mobility patterns of a person and observe how these evolve over time. Researchers have already shown that by analyzing these patterns it is possible to not only, e.g., detect the lack of physical activity in daily routines, but even predict the occurrence of depression episodes. The use of physiological sensors – e.g., using smartwatches that can measure heart rate or skin conductance – makes it possible to detect in real-time people's stress levels or even emotions. Combining this information allows us to build *mobile sensing* systems that are able to understand and adapt to humans' emotional and mental state.

Such a *mobile sensing* system usually consists of a sensing application running on the phone and a data analytics service running in the back-end. The application automatically collects data from the phone's built-in sensors and at the same time supports the collection of self-reports, i.e., answers to questionnaires filled-in by the user. In the back-end, a data analytics service processes the collected data to find models that link sensor data to the ground-truth information collected through questionnaires.

Despite their high potential, the use of these systems in real settings is still impractical. In fact, many challenges are still open. One of the challenges is to properly interpret the large amount of data gathered from mobile devices, for this purpose raw sensor data should be processed and aggregated into interpretable behavioral features (e.g. GPS longitude and latitude data points should be converted into mobility traces). The good quality of the data should be maintained throughout the study, indeed due to technical issues (e.g. broken devices, problems with the connection of the app with the server) data might be lost or include noise (e.g. presence of artifacts in the signals due to the wrong placement of the smartwatches). Another challenge is to maintain people's motivations to participate to the study and answer the questionnaires. One way for maintaining participants' adherence over time is to provide meaningful feedback and insights through the visualization of their data.

The goal of this project is to design and develop a robust data analytics service to analyze the data, to provide the users with meaningful insights about their data and to allow study investigators to more efficiently monitor the data collection. The main tasks of the data analytics service are: (1) *App data processing:* automatic generation of behavioral features from passive sensing and self-reports; (2) *Data visualization*: generate plot to show how sensors data (and consequently inferred behavior) changes over time; (3) *Data quality*: creation of daily reports to the research team to monitor the adherence of the participants, the quantity and the quality of their data; (3) *Feedback to the participants*: provide automatic report and feedback.

Required skills and knowledge

- Data analysis language (i.e., Python)
- Technologies for data visualization and feedback

Expected outcomes

The expected outcomes of this project are as follows:

- Extend an existing mobile sensing system by designing and implementing a data analytics service to monitor *in-the-wild* studies and provide feedback to the participants.
- Implement a dashboard to visualize data from smartphone and wearable devices for facilitating the analysis of the data.

Supervisors and contact information

Elena Di Lascio: elena.di.lascio@usi.ch, Shkurta Gashi: shkurta.gashi@usi.ch, Prof. Dr. Silvia Santini, silvia.santini@usi.ch

Real-time Feedback for Smartphone and Smartwatch Users

UROP project proposal at the Università della Svizzera Italiana (USI), Lugano. Switzerland

Background

Smartphones and smartwatches can nowadays be used to unobtrusively and continuously monitor and foster human's health, well-being, behavior and more. Several studies have shown that data collected with such devices can be used to infer participants' mental health illness (i.e., depression), behavioral changes (i.e., sleep and stress patterns) and more. Combining the information from these devices allows us to build real-time systems or even robots that are able to understand and adapt to human's social and emotional cues. Recent advancements in mobile sensing technology have enabled broader use of these devices in everyday life. This has enhanced ecological validity by revealing whether laboratory-based phenomena also occur in natural settings.

To be able to build models that predict participants' health and behavior we need a significant amount of labeled data. However, participants lose their motivation of being part of a study after some point, especially in long-term studies. A possible approach to maintain participants motivation is to provide feedback about their behavioral patterns. The feedback may be provided in different forms such as data visualization, behavioral intervention and gamification. Another challenge faced in studies performed in natural settings is the quality of collected data. This is because the physiological signals are prone to artifacts caused by movements, environmental conditions and recording procedure. Therefore, a continuous monitoring of the status of physiological signals is needed to notify the user when the data is corrupted.

The goal of this project is to investigate novel ways to provide feedback to smartphone and smartwatch users to track their behavioral patterns over time. The student is required to extend an existing mobile application with a real-time feedback mechanism that aims (1) to maintain the participation of people by providing motivational and educational feedback and (2) to ensure the good quality of the collected data.

Required skills and knowledge

- Mobile programming (i.e., Android)
- Willingness to learn about mobile sensing systems

Expected outcomes

- Extend an existing mobile application by implementing a real-time feedback model to provide users more insights about their behavioral data.
- Implement an automatic approach for preprocessing, extracting features and checking the quality of the data in real-time.

Supervisors and contact information

Shkurta Gashi: shkurta.gashi@usi.ch, Elena Di Lascio, elena.di.lascio@usi.ch,

Prof. Dr. Silvia Santini, silvia.santini@usi.ch

Synthesis of Digital Drawing Strokes

Contact: Michal Piovarci (michal.piovarci@usi.ch), Prof. Piotr Didyk (piotr.didyk@usi.ch)
Group website: https://www.pdf.inf.usi.ch/

Motivation

Drawing is a tool to share stories and emotion. In currents digital age, there is a huge demand from feature movie companies, the gaming industry, and many others for high-quality digital drawings. With continuously higher demand on a digital version of the art the drawing tools also improved. Companies like Wacom, Apple, and Microsoft are offering digital styli with negligible latency as well as high-precision orientation and pressure sensitivity. However, many effects that the artists intend to achieve are mediated through the characteristic marks a drawing implement leaves on paper. This leads companies like Adobe, PaperFiftyThree, and others to design digital brushes which approximate the look of real tools. The behavior of a tool depends on the velocity, applied pressure, angle, and other factors. Since capturing the behavior of a tool is complex the rendering is often done by carefully tuning parameters of a procedural model until the generated strokes look just right.

Goal

We want to design an alternative method with an automatic generation of drawing strokes. To this end, we will tackle two challenges: data gathering of strokes from analog drawing tools, and rendering of drawing strokes. For data gathering, we propose to use a recently released pressure sensitive tablet Sensel Morph. The tablet consists of a high-density array of pressure sensors. We can use the output of the tablet to reconstruct the velocity, pressure, and position of the drawing tool during drawing. Next, we can register the drawing parameters with a photo of the stroke. While there are several solutions for synthesizing strokes using an input from a tablet, one promising approach would be to leverage machine learning techniques. A potential model for rendering the drawing strokes is a time-dependent image-to-image translation neural network. The input to the model could be time-dependent pressure and the output will be the corresponding strokes.

Prerequisites

- Good programming skills
- Experience in processing image data
- Helpful but not required: experience/interest in machine learning techniques and tools (e.g., Tensorflow), basic knowledge of OpenCV

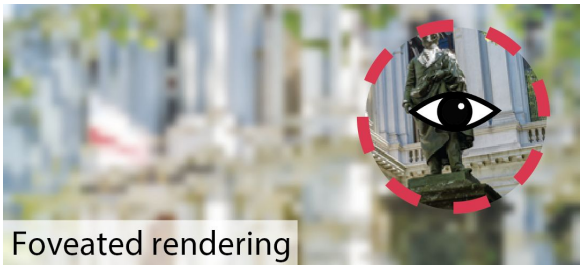
Please do not hesitate to contact us for more details

Towards optimal content generation for novel displays

Contact: Prof. Piotr Didyk (piotr.didyk@usi.ch)
Group website: https://www.pdf.inf.usi.ch/

Motivation

The constant demand for higher image quality pushes display manufacturers to constantly increase the capabilities of new display devices. Current trends include high resolution and framerate, wide field of view for virtual reality (VR) and augmented reality (AR), as well as high luminance range. Despite the significant advancements in graphics hardware, current techniques for generating and optimizing content for these new devices struggle do fulfill the increasing computational requirements. This hampers the visual quality, power efficiency, and, in consequence, the adoption of the newly emerging devices such as VR and AR headsets. Interestingly, high visual quality is not required in the entire visual field. For example, people do not see a high-resolution in their peripheries. Recent advancements in eye-tracking technology allow us to exploit such insights and save computational recourses by adjusting the quality of the image according to the current gaze location. The idea is known as *foveated rendering*. While it has been acknowledged that the technique may become a key enabler for the new, high-resolution, wide-field-of-view displays, it is still unclear how to best utilize the eye-tracking technology to minimize the computation while preserving visual quality.



Goal and research directions

The research activities in this project will be a part of a bigger endeavor of our group to push the limits of foveated rendering. More precisely, we want to provide a set of techniques which exploit the eye-tracking technology to provide high visual quality avoiding at the same time spending computational resources on irrelevant for observer image details. Currently, we plan to apply the solutions to the new VR displays, but the potential benefits extend also to future AR headsets. The possible directions for research within this project include but are not limited to:

1. A method that characterizes lens distortion in VR headsets and use this information to drive a foveated rendering technique
2. A machine-learning-based model for guiding the foveated rendering techniques
3. A foveated rendering technique exploiting NVIDIA Variable Rate Shading, which avoids visible temporal aliasing

Prerequisites

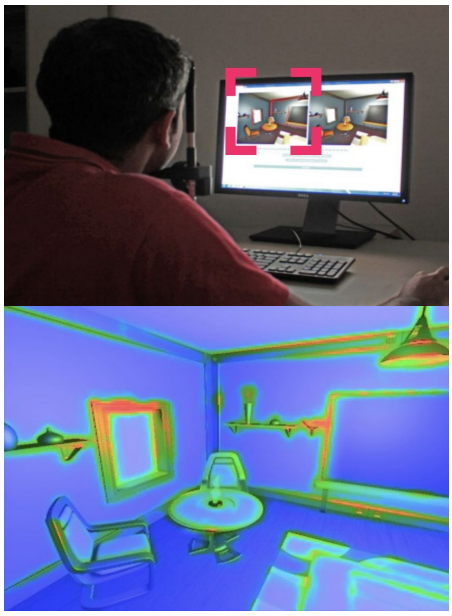
- Good programming skills, preferably C++
- Basic knowledge of image processing
- Helpful, welcomed, required depending on the project direction:
    - experience in using 3D graphics libraries, preferably OpenGL & GLSL
    - experience/interests in machine learning, more specifically ConvNets.

Please do not hesitate to contact us for more details

Eye-tracking Procedure for Training Visibility Metrics

Contact: Prof. Piotr Didyk (piotr.didyk@usi.ch)
Group website: https://www.pdf.inf.usi.ch/

Motivation



A huge number for imaging and computer graphics applications requires automatic methods for evaluating image/video quality as well as localizing visible distortions. The recent state-of-the-art image visibility and quality metrics rely on new machine learning techniques such as convolutional neural networks. They either predict the visible differences between the two input images or automatically judge image quality. Although the new techniques significantly improve prediction accuracy over standard metrics, there are still two obstacles hampering the application and development of better techniques:

1. Training metrics requires large datasets derived from user experiments, which often rely on user markings. This makes it hard to collect data sufficient for training general-purpose metrics.
2. The metrics perform better if they are trained on a tailored-to-application dataset.

As a result, data collection is the main challenge in training better image metrics.

Goal

In this project, we want to replace tedious and time-consuming procedures of collecting user data, which often involve image markings performed by users, with a much faster technique exploiting eye-tracking information. The main idea is that instead of people carefully marking visible differences between images, the users will simply watch changing images. The collected and analyzed eye-tracking data will be later used to derive information for training. The main goal of the project is to develop the method that turns the eye-tracking data into so-called visibility maps, which describe a local probability of detecting differences between two images. To this end, we would like to exploit human attention which is triggered by temporal changes in image data. By flipping two, similar images, the human gaze should be attracted by subtle differences between them giving us information about the image differences. The challenge here is how to transform the data from such an experiment to equivalent information coming from experiments where people observe static images side-by-side. Another possible outcome of this research might be a fast and easy procedure for evaluating differences between images using short eye-tracking sessions.

Prerequisites

- Good programming skills
- Experience in processing image data

Please do not hesitate to contact us for more details

UROP 2019 – Project Proposal:

Transforming Fairy Tales into Narratives for Video Games

In 2014 we designed, prototyped, and tested with and for primary school children in local schools “Fiabot! a simple iPad application. Fiabot! is based on Keynote, and enables the creation of interactive and multimedia stories by letting children store and combine in a structured manner into one presentation images, text, video and audio. It fits with class activities where Fiabot! provides children with the opportunity to increase their ability to create a specific story genre by giving indications of the structure and “ingredients” needed for each story type as well as reinterpreting and formalizing the workflow of the story creation.

In the past few years, we have extensively studied it from the user point of view. We have explored how Fiabot! helped creating stories and introduced it to older (intermedia school) and younger (pre-school) children. Now it is time to focus on how the stories produced by children could become narratives for video games!

This short project aims at delivering a web app to be used by children for creating their stories and transform them into narratives. We expect the candidate student to start from the analysis of the limitations of the apps available to help create narratives for adventure games. He/she will then proceed with the design, implementation and testing of a web app to be used by children.

This project will offer a student the chance to engage with the complexities of working with real users without having to go through the ordeal of running a user study from scratch. The chosen student will also enjoy working in a truly multidisciplinary team.

It would be an ideal experience for a student, who:

- has achieved a very high level of competence during the Web Atelier class,
- is a keen player and has a deep knowledge of different types of video games and their essential components, mainly narrative and gameplay,
- is looking forward developing apps for Web and tablets while working for real users.

This is the very first step of a study looking at the link between narrative and game play, how to support the construction of engaging narratives and how children can become co-designers of video-games.

Advisor: Dr. Monica Landoni

Picture-in-Picture detection with Neural Networks

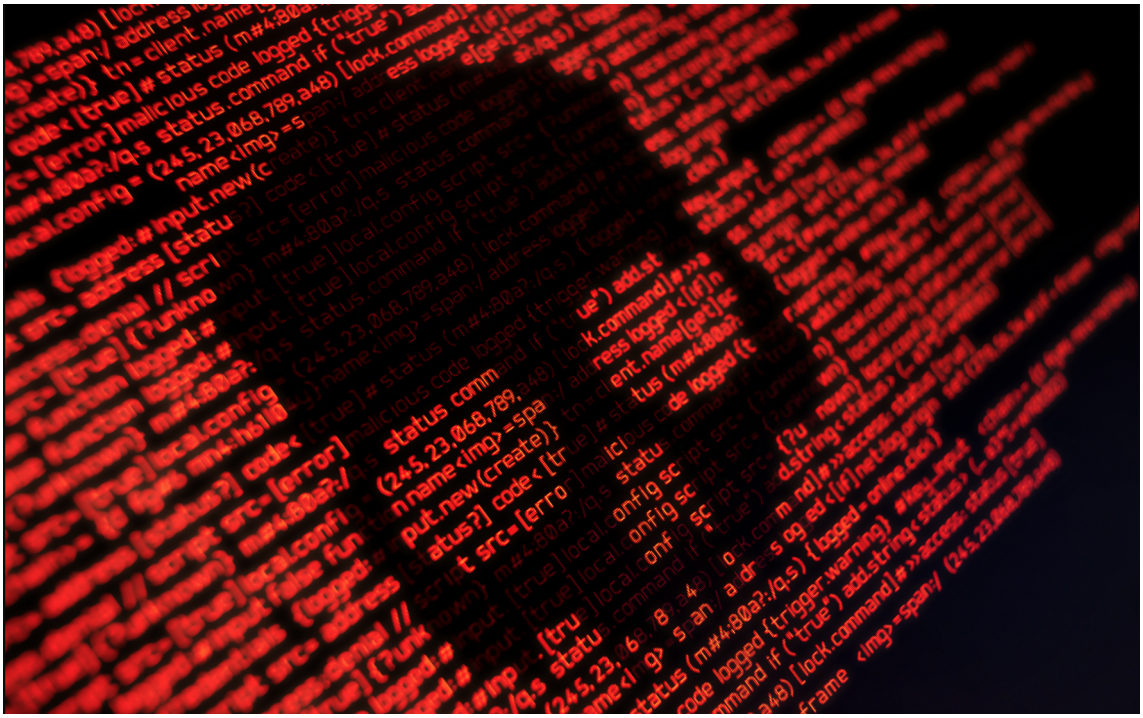
Picture-in-picture (PiP) is a particularly interesting transformation in the field of video copy detection and near-duplicate retrieval, as it is both very easy to create and especially difficult to detect. At the present time, even very advanced copy detection algorithms such as YouTube's content id are not able to properly detect this transformation yet.

Different methods have been developed to tackle this problem, but none of them can simultaneously address different families of PiP transformations, involving single or multiple, static or dynamic background and foreground videos.

The purpose of this project is to develop a new algorithm, based on deep neural networks, to automatically detect videos hidden within other ones using the PiP algorithm. The student will be followed during the whole development process, from the creation of the dataset (actually the extension of a currently existing one) to the implementation of the model, its evaluation, and final refinement.

Medium to advanced knowledge of Python and its libraries numpy and Tensorflow, together with a good background in mathematics (having followed a machine learning course is a plus), are strongly encouraged.

Geometric Deep Learning for Anomaly/intrusion detection



In the last decade, Deep Learning approaches (e.g. Convolutional Neural Networks and Recurrent Neural Networks) achieved unprecedented performance on a broad range of problems coming from a variety of different fields.

Geometric Deep Learning (GDL) deals with the extension of Deep Learning techniques to graph/manifold structured data. Being very abstract models of systems of relations and interactions, graphs naturally arise in various fields of science. For this reason, GDL techniques have been successfully applied in problems such as computer graphics and vision, protection against adversarial attacks, and recommendation systems, to mention a few.

The purpose of this project is to apply GDL in the field of computer security, training a system to perform anomaly and/or intrusion detection based on heterogeneous sources of information that include graph structured data.

Medium to advanced knowledge of Python and its libraries numpy and Tensorflow, together with a good background in mathematics (having followed a machine learning course is a plus), are strongly encouraged.