# LCAV – Audiovisual Communications Laboratory EPFL I&C Semester project 2020, Spring Semester Project description and planning

1. Project Description

This section is filled in by the project supervisor.

**1.1. Title:**

*Python simulations for event-driven sensor sampling algorithms*

**1.2. Description (around half a page):**

Event-driven cameras are vision sensors that output pixel-level brightness changes instead of standard intensity frames. Inspired by biological acquisition, they offer high-dynamic range, high-speed updates without motion blur.

The focus of this project is to assist with the foundations of a theory that can explain the performance of these cameras, how to exploit them and build optimal algorithms, and inform the development of future hardware design.

For simplicity, analysis in this project is restricted to analysis of a single sensor (pixel in an event-driven context). The primary objective of this project is to implement in python (with a jupyter notebook) the three following sampling algorithms:

[1] Feichtinger, Hans G., et al. "Approximate reconstruction of bandlimited functions for the integrate and fire sampler." Advances in computational mathematics 36.1 (2012): 67-78.

[2] Lazar, Aurel A., and László T. Tóth. "Time encoding and perfect recovery of bandlimited signals." *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*. Vol. 6. IEEE, 2003.

|  |
| --- |
|  |

[3] Alexandru, Roxana, and Pier Luigi Dragotti. "Time encoding and perfect recovery of non-bandlimited signals with an integrate-and-fire system." *2019 13th International conference on Sampling Theory and Applications (SampTA)*. IEEE, 2019.

A secondary objective is to test these algorithms with examples to show their performance.

A third objective (nice to have) is to test modified methods for on an event-based sensor.

|  |
| --- |
|  |
|  |
| APA |  |

|  |
| --- |
|  |
|  |
| APA |  |

**1.3.** **Type of Work (e.g., theory, programming):**Theory, python numerical programming.

**1.4. Prerequisites (e.g., signal processing for communications, C++):**

Python 3, some signal processing background, strong mathematical ability.

**1.5. Supervisors:**

Prof. Paul Hurley (main supervisor, Western Sydney University), Dr. Matthieu Simeoni (supervisor on EPFL-side, LCAV).

2. Student Information

This section is filled in by the student.

Name: Ghassen Karray

E-Mail:

School (e.g., I&C, STI): I&C

Program (Comm. Sys., Comp. Science):

Cycle (B.Sc./M.Sc./EDIC):

Semester (1, 2, 3, ...):

3. Project Planning

*This part is filled by the supervisor and discussed with the student. It should be completed and agreed upon before the end of the 2nd week (hard deadline on the \*when?\*) and sent to the responsible person (matthieu.simeoni@epfl.ch). After the submission of the plan, a modification is still possible, but it should be motivated at the midterm or the final presentation.*

**3.1. Deliverables:**

*Explain in a few sentences the expected concrete outcome of the project (e.g., a C program that removes red eyes, a Matlab simulation of sound propagation in a room, a subjective test on N persons of an algorithm).*

The deliverables of this project are:

1. A jupyter notebook that allows one to test all three methods. Then a series of test data that shows how they perform.

**3.2. Timeline:**

*Explain shortly (in a few sentences) what the student should achieve for every week of the project in order to reach the final goal described in the previous section. Please remind that the students are supposed to spend 30 hours to prepare the required background before the beginning of the semester (e.g., reading papers, revising Matlab/C). The amount of work during the semester should correspond to 12 hours per week. After the end of the semester, 30 extra hours should be spent to complete report and presentation.*

|  |  |
| --- | --- |
| **Week #** | **Planned Work** |
| 1 | Read and understand the Lazar paper [2] |
| 2 | Implement the Lazar algorithm |
| 3 | Read and understand the Feichtinger paper [1] |
| 4 | Implement the Feichtinger algorithm |
| 5 | Finish up tests for both algorithms  Present achievements to supervisors. |
| 6 | Read and understand the Alexandru paper [3] |
| 7 | Prepare slides for midterm. |
| Midterm presentation. | |
| 8 | Decide on performance tests to explain the sampling schemes |
| 9 | Implement performance tests |
| 10 | Implement performance tests |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 | Write up report |
| 15 | Prepare slides for final presentation. |
| Final presentation. | |