## Disparity Algorithm Implementation

Please note that all examples and laboratory exercises conducted during the laboratory sessions are not assessed. This coursework is the only marked assessment for this course.

A disparity algorithm is given in C language (see blackboard: Disparity_CCS5.zip). You are required to study the code and increase the performance in terms of speed.

**Tasks to perform:**

1. You are required to study the C code (**stereo_vision_c.c**), optimise the algorithm and use intrinsics and compiler switches such as #pragma to increase the performance. Always compile your code with -O3 optimization flag. Your code should be faster than the base line version (0.58s) with zero errors. You get full marks for achieving faster than 0.01s with zero errors.

   **Total mark 20%**

2. Write the linear assembly for the stereo vision code (**stereo_vision_sa.sa**), in order to increase the speed performance using the SIMD instructions. You get full marks for achieving faster than 0.01s with zero errors.

   **Total mark 60%**

   **Note: You get 30% if the code in linear assembly operates properly regardless of the performance (with zero errors).** If your modified algorithm achieved a higher performance but with slight errors, please submit it along with a zero-error version code and explain clearly in the report, we will mark them on a case-by-case basis.

3. Write a formal report explaining the implementation and showing/discussing the results. Presentation will be taken into account.

   **Total mark 20%**

Make sure to check the correctness of your code by looking at the printed error rate and the generated image as explained in the lecture. When you start working on linear assembly, remember to reset the output buffer from C code to zeros (line 76 in main.c).

Please submit **one PDF report and one zip file** with all the projects. (Please DO NOT submit only the source files). Marks will be deduced if a project will not compile.

ONE report per student, with name and email address included (**individual work**). Submission should be through Blackboard.

The report should follow the template given on Blackboard and should not exceed 6 pages; extra pages can be used in the appendix. Your report should include a table like this, showing the results of all your code:

| Method | Time in Seconds |
|---|---|
| Non-optimised C | 0.58 |
| Optimised C | 0.02 |
| Optimised Linear Assembly | 0.05 |