Netatmo complementary technical Test – Part 2

Vision/AI Team

Guidelines

- This technical test is not limited in time. Please take your time to do it. Send us the result by email when you are ready and please, don't use Github or any public repository to share it. Use.tar or .zip archives to share your code with us.
- We encourage you to comment your answers and your code, make assumptions and explain your choices as much as possible; first and foremost, we are interested in your reasoning.
- Algorithms and code architecture go hand in hand, so special care should be given to both.

Introduction

We suppose we have different frames coming from an acquisition pipeline. For each frame F, a previous algorithm **A1** provides a list of bounding boxes, delineating different regions of interest in the scene, in the frame coordinate system. Let's assume frame dimension is $M \times N$. We are interested in processing these regions of interest by a second algorithm A2 and relating the processing results to the original frame.

Question 1

As a preprocessing step of a second algorithm **A2**, we would like to combine all the regions corresponding to the bounding boxes into a new image FRegions of dimension $D \times D$, where D given by **A2** and $D < \min(M, N)$.

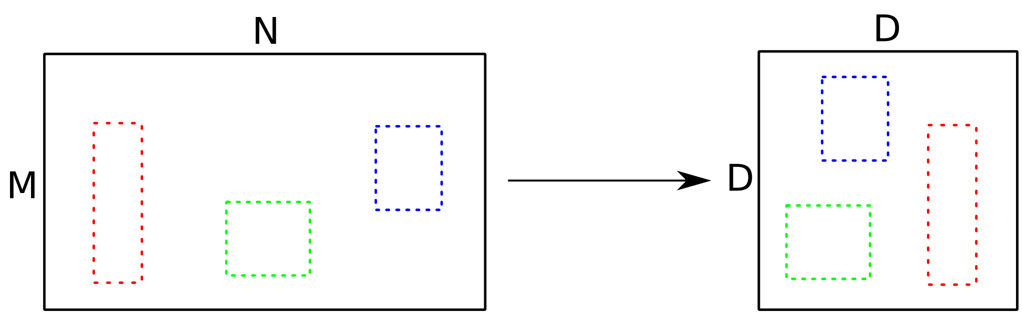

Figure 1: Example

You will write in **modern C++** an application which takes an image F and a list of bounding boxes B as input and outputs a new image $FRegions$ composed of only the regions corresponding to the given bounding boxes. The application should verify the following:

- only the orientation of regions is to be preserved, their relative order is not relevant
- $FRegions$ should retain the most information possible from F, namely you should try to keep the regions at their highest possible resolution
- if the given regions do not suffice to fill out $FRegions$ compeletly, remaining empty space can be filled with zeros.

Question 2

A2 then takes as input $FRegions$ and outputs new bounding boxes B'. We would like now to compute the location of each of these new bounding boxes in F reference. You will adapt your previous code to be able to address this problem: given F and $FRegions$, the program will take a bounding box $b' \in B'$ in $FRegions$'s coordinate system and return its coordinates in F's coordinate system. We assume for the sake of simplicity, that no bounding boxe in B' overlaps more than one region in $FRegions$.

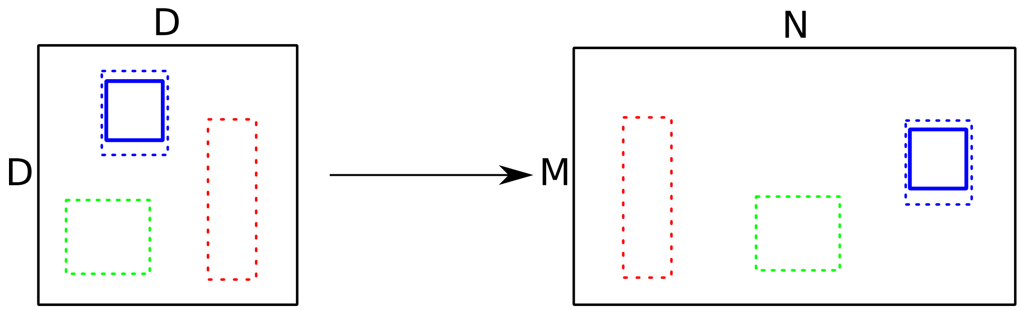

Figure 2: Example

Question 3

We have now a way to combine regions of interest of an image into a new image and to transform locations from the region-based image to the initial one. We would like now to be able to provide to the algorithm A2 either the region-based image or the initial image without transformation, which modifications to your code architecture do you suggest in order to handle this?