

TP1

Sliced Histogram Matching

In this TP, we will implement a color transfer technique that makes use of sliced optimal transport.

Introduction. Image manipulation

We provide you with a short C++ code that loads and save an image. Test it.

Images are stored as a 1-d array of unsigned chars. There are width*height*3 values, interleaving red, green and blue components, consecutively row by rows. To get used to the image structure, implement a method that saves a 512x1024 image where the top vertical half is pure blue, and the bottom half is green.

Results. Slice Optimal Transport

We will now implement the sliced optimal transport as seen during the last lecture. Recall that this method randomly draws 3-d directions, projects the two images as seen each as a point cloud, sort the projected samples, and brings the first image closer to the second along the line direction. Implement this approach.

It is in fact possible to draw several random lines at the same time, perform the optimal transport matching for all of these lines, and then bring the first image closer to the second using the sum of the displacements over all of these directions. This usually has the advantage to converge faster, and to allow for parallelization. Implement this method and parallelize it via OpenMP.

Compare the convergence rate of both approaches.