

From: Arthur Lambalk alambalk@spilberg.nl
Subject: Programming Assignment
Date: April 15, 2019 at 09:33
To: Miroslav Radojevic miroslav.radojevic@gmail.com

AL

Dear Miroslav,

As discussed hereby the assignment!

Can you send it back to me before 22th of April?

Good luck!

Best,
Arthur Lambalk



Met vriendelijke groet / Best regards,

Arthur Lambalk
Principal Consultant I

+31 (0) 20 2402 248
+31 (0)6 51 880 947

Algemene Voorwaarden



Op alle aanbiedingen en overeenkomsten zijn de algemene voorwaarden van Altus Staffing B.V. van toepassing, welke toegankelijk zijn via bovenstaande link.

Disclaimer

1. This e-mail is meant for the intended recipient only. If you have received this by mistake, you are kindly requested to notify the sender and then delete the e-mail from your system; access to, disclosure, copying and/or distribution of this e-mail by anyone other than the intended recipient is prohibited.
2. If you, as intended recipient, have received this e-mail incorrectly, please notify the sender immediately. This e-mail is confidential and may be legally privileged. Spilberg does not guarantee that the information disclosed in or attached to this e-mail is correct and does not accept any liability for damages related thereto.

Van: Bas Boom <BBoom@cyclomedia.com>

Aan: Arthur Lambalk <alambalk@spilberg.nl>

Onderwerp: Programming assignment

Hereby the programming assignment for Miroslav Radojevic:

Automatic detection of vanishing points

The goal of this assignment is to create a simple python/C++ program to detect the vanishing points (https://en.wikipedia.org/wiki/Vanishing_point) in an image.

An image can have multiple vanishing points. The idea is that you automatically find the dominant straight lines in the images and plot them. Based on these lines you can estimate the vanishing point (also describe how you deal with some of the uncertainties in your estimation, for instance that not all lines intersect the point in the same place). Another issue is that not all the lines will lead to a vanishing point. The vanishing point might also be outside the image, please extend the images (with maximum with 2 time the image size) with white pixels and draw the lines to the vanishing point.

If you create a C++ program, please give us a CMakeLists.txt so we can compile it on different platforms.

You are allowed use libraries like opencv and numpy, please think about what we would like to see in good (clean, modular, reusable) code.

Dr. Ing. Bas Boom

Computer Vision Expert | CycloMedia Technology B.V.

P.O. Box 2201, 5300 CE Zaltbommel (the Netherlands)
Van Voordenpark 1b, 5301 KP Zaltbommel (the Netherlands)

T +31 610 743 180, **F** +31 418 556 101
E bboom@cyclomedia.com, **I** www.cyclomedia.com



