## **[Source code Inverse Perspective Mapping C++, OpenCV](https://marcosnietoblog.wordpress.com/2014/02/22/source-code-inverse-perspective-mapping-c-opencv/)**

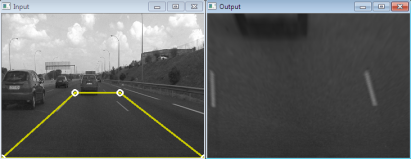
Posted on [February 22, 2014](https://marcosnietoblog.wordpress.com/2014/02/22/source-code-inverse-perspective-mapping-c-opencv/" \o "16:17)by [marcosnietodoncel](https://marcosnietoblog.wordpress.com/author/marcosnietodoncel/" \o "View all posts by marcosnietodoncel)

Hi all,

Today I bring a very simple code that might be of interest for some of you. It is a C++ class to compute Inverse Perspective Mappings (IPM), or sometimes called bird’s-eye views of a planar surface.

The link to the code: <https://sourceforge.net/projects/ipmapping/files/>

(I am providing here a CMakeLists.txt and some c++ files, so you need to create a solution with CMake, and the a C++ compiler).



It is nothing else than a plane-to-plane homography, but in my experience it is not that easy to compute. I am using here OpenCV remap functions.

In the example I have hard-coded the necessary four-point correspondences required to compute the homography, but you can compute then as you want. You might use vanishing points, camera calibration information, or any other source.

Of course this type of classes can get really useful when you add them information about the camera calibration.

Hope it is useful!

Kind regards,