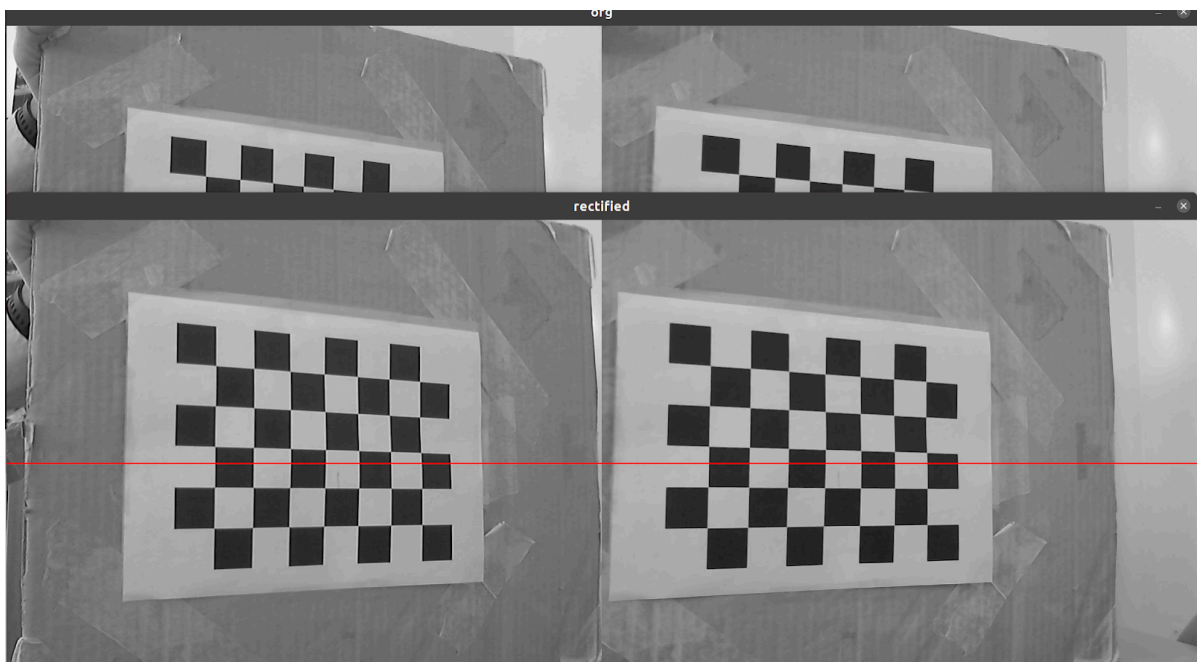


Práctica 2: Comprobación de imágenes rectificadas

Cree el programa *stereo_checkundistorted* que reciba como entrada una imagen estéreo y el correspondiente fichero de calibración estéreo. El programa debe mostrar dos ventanas. En la primera ventana se muestran las imágenes izquierda y derecha originales de la cámara estéreo. Al pasar el ratón por encima, se dibujará de forma dinámica una línea horizontal que cruza ambas imágenes.

Además, deberá mostrar otra ventana similar donde se muestran las imágenes izquierda y derecha después de la rectificación. Igualmente, el programa deberá pintar una línea horizontal al moverse sobre la ventana.

De esta manera, podremos comprobar visualmente cómo de bien la rectificación ha funcionado, creando dos cámaras paralelas.



Uso:

```
./stereo_checkundistorted stereo_image.jpg stereocalibrationfile.yml
```

Para rectificar las imágenes, puede usar el siguiente código:

```
//Structure that contains the Stereo Pair Calibration information.
//This will be calculated using stereo_calibrate
struct StereoParams{
    cv::Mat mtxL,distL,R_L,T_L;
    cv::Mat mtxR,distR,R_R,T_R;
    cv::Mat Rot, Trns, Emat, Fmat;
};

void rectifyStereoImages(const StereoParams &sti,cv::Mat &left,cv::Mat &rigth){
    cv::Mat rect_l, rect_r, proj_mat_l, proj_mat_r, Q;
    cv::Mat Left_Stereo_Map1, Left_Stereo_Map2;
    cv::Mat Right_Stereo_Map1, Right_Stereo_Map2;
    cv::stereoRectify(sti.mtxL,
sti.distL,sti.mtxR,sti.distR,left.size(),sti.Rot,sti.Trns,
                    rect_l,rect_r,proj_mat_l,proj_mat_r,
                    Q,cv::CALIB_ZERO_DISPARITY, 0);
    cv::initUndistortRectifyMap(sti.mtxL,sti.distL,rect_l,proj_mat_l,
                    left.size(),CV_16SC2,
                    Left_Stereo_Map1,Left_Stereo_Map2);
    cv::initUndistortRectifyMap(sti.mtxR,sti.distR,
                    rect_r,proj_mat_r,
                    left.size(),CV_16SC2,
                    Right_Stereo_Map1,Right_Stereo_Map2);
    cv::Mat AuxImage, Right_nice;
    cv::remap(left, AuxImage, Left_Stereo_Map1, Left_Stereo_Map2,
        cv::INTER_LANCZOS4,cv::BORDER_CONSTANT,0);
    AuxImage.copyTo(left);
    cv::remap(rigth, AuxImage, Right_Stereo_Map1, Right_Stereo_Map2,
        cv::INTER_LANCZOS4,cv::BORDER_CONSTANT,0);
    AuxImage.copyTo(rigth);
}
```