Práctica 3: Mapa Denso de Disparidad Estéreo

Cree el programa *stereo_disparity* que recibe como entrada una imágen estéreo, el fichero de calibración estéreo, y como resultado calcula la disparidad entre las imágenes en cada pixel. Para aquellos puntos con disparidad válida, genera como salida un fichero con formato <u>OBJ</u> que podrá visualizar con el programa <u>MeshLab</u>. Para probar puede utilizar las imágenes en el <u>siguiente enlace</u>.

Uso:

./stereo disparity stereo image.jpg calibration.yml out.obj

Para realizar el proceso, deberá

- 1) Cargar la imágen estéreo
- 2) Rectificar las imágenes
- 3) Utilizar la clase cv::StereoBM para realizar el cálculo de la disparidad.
- 4) Convierta la disparidad obtenida a valores 32bits flotantes

```
// Converting disparity values to CV_32F from CV_16S
    disp.convertTo(disparity,CV_32F, 1.0);
    disparity=disparity/16.f;
```

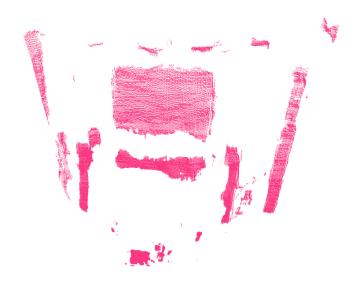
- 5) Para aquellos puntos con disparidad > 10, triangule usado las ecuaciones básica del par estéreo: Z= |T|*f /d; X= (x-cx)*Z/f; Y= (y-cy)*Z/f;
- 6) Guarde los puntos a formato obj. Puede utilizar la siguiente función para ello:

```
void writeToOBJ(std::string path,std::vector<cv::Point3f> points){
    std::ofstream file(path,std::ios::binary);
    for(auto p:points)
        file<<"v "<<p.x<<" "<<p.y<<" "<<p.z<<endl;
}</pre>
```

En el <u>siguiente enlace</u> tenéis el fichero resultado para la primera image reconstruction/m001.jpg



Ahora bien, si en lugar de usar la imagen con el tamaño original, la reducimos a la mitad obtenemos la <u>siguiente reconstrucción</u>.



Para hacerlo, se deberá reducir a la mitad también el f,cx y cy.