1. left_body_cam left_cam_body 的区别
   left_body_cam = left_cam_body.inverse();
2. 每个 Frame 都有多个 Transform，T_imu_cam, T_cam_imu, T_cam_world，这些的区别
   整个系统分为世界坐标系，body 坐标系（IMU 坐标系），相机坐标系；T_imu_cam, T_cam_imu, T_cam_world 等皆为不同坐标系之间的转换。
3. camera.cpp 里面的 project3 和 backproject3 里面用到的 focal_matrix 是怎么获取的
   相机标定，可见 kalibr 使用说明。
4. projector3 和 backprojector3 的几何关系给个直观的解释吧

```
bool PinholeCamera::backProject3(const Eigen::Ref<const Eigen::Vector2d>& keypoint,
                                 Eigen::Vector3d* out_point_3d) const {
  double x = (keypoint[0] - cx_) * fx_inv_;
  double y = (keypoint[1] - cy_) * fy_inv_;
  (*out_point_3d)[0] = x;
  (*out_pc
  (*out_pc
  return t
}

const ProjectionResult PinholeCamera::project3(
    const Eigen::Ref<const Eigen::Vector3d>& point_3d, Eigen::Vector2d* out_keypoint,
    Eigen::Matrix<double, 2, 3>* out_jacobian_point) const {
  const double z_inv = 1 / point_3d(2);
  // TODO(tcies) precalced member?
  const Eigen::DiagonalMatrix<double, 2> focal_matrix(fx_, fy_);
  const Eigen::Vector2d uv = point_3d.head<2>() * z_inv;
  (*out_keypoint) = focal_matrix * uv + Eigen::Vector2d(cx_, cy_);

  if (out_jacobian_point) {
    Eigen::Matrix<double, 2, 3> duv_dxy;
    duv_dxy.leftCols<2>() = Eigen::Matrix2d::Identity() * z_inv;
    duv_dxy.rightCols<1>() = -point_3d.head<2>() * z_inv * z_inv;

    (*out_jacobian_point) = focal_matrix * duv_dxy;
  }
  return ProjectionResult::Status::KEYPOINT_VISIBLE;
}
```

(parameter) Eigen::Vector3d *out_point_3d
Search Online

P(x,y,z)为世界坐标系中的一个三维点，$P_c = \mathrm{T}_{\mathrm{cam}_{\mathrm{world}}} * P = (x_c, y_c, z_c)$为转化到相机坐标系中的三维坐标， 像素点为$ux = f_x * \frac{x_c}{z_c} + c_x,\ uy = f_y * \frac{y_c}{z_c} + c_y$

5. 几个的 Option 类的意义，现在 DetectiorOptions 的意义比较明确了
6. pyramid 的解释
7. tracking 初始化的时候对各种参数的初始化是怎么标定，是否有类似这样的相机标定过程。

https://github.com/Nocami/PythonComputerVision-6-CameraCalibration

```
mtx:
 [[2.70717016e+03 0.00000000e+00 1.63764379e+03]
 [0.00000000e+00 2.70842435e+03 1.11011972e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

8 fisheyecamera 的 project3 需要讲解一下

OpenCV 中使用的鱼眼相机模型为：

$$\theta_d = k_0\theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9$$

即通过无畸变图像中的点能够计算出鱼眼图像中的畸变点。这种模型在 OpenCV 的鱼眼相机标定方法中是适用的，OpenCV 借助标定板对鱼眼相机进行标定。

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = RX + t$$

$$x_c = \frac{X_c}{Z_c}, y_c = \frac{Y_c}{Z_c}$$

$$r^2 = x_c^2 + y_c^2$$

$$\theta = \arctan(r)$$

$$a_d = k_0\theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9$$

$$x_d = \frac{\theta_d}{r} x_c, y_d = \frac{\theta_d}{r} y_c$$

$$u = f_x x_d + c_x, v = f_y y_d + c_y$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = RX + t$$

$$x_c = \frac{X_c}{Z_c}, y_c = \frac{Y_c}{Z_c}$$

$$r^2 = x_c^2 + y_c^2$$

$$\theta = \arctan(r)$$

$$a_d = k_0\theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9$$

$$x_d = \frac{\theta_d}{r} x_c, y_d = \frac{\theta_d}{r} y_c$$

$$u = f_x x_d + c_x, v = f_y y_d + c_y$$