Minth OpenMVG Pipeline

相机标定

标定相机初始的内外参

有两种方法：

- 使用Calibra进行相机标定，张正友标定法：

  标定

  - 内参：fx，fy，cx，cy
  - 外参：R，t
  - 畸变系数：
    - 径向畸变：k1，k2，k3
    - 切向畸变：p1，p2

  张正友相机标定
- 用colmap来进行重建、标定，基于marker

模板生成

用名义值生成模板，将CAD模型的中心点和一圈点往各个相机下投影，生成Linemod或Halcon模板来进行2d匹配

2d检测

1. 图像去畸变，利用初始的畸变系数：k1，k2，k3，p1，p2

   其中

   - k1 k2 k3用于处理径向畸变

     $$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
     $$y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
   - p1 p2用于处理切向畸变

     $$x' = x + (2p_1 y + p_2(r^2 + 2x^2))$$
     $$y' = y + (p_1(r^2 + 2y^2) + 2p_2 x)$$

2. 利用Halcon/Linemod进行模板匹配或深度学习，找到圆孔的中心点：

   1. ROI：获取到每个ROI（手工标注的ROI）

      格式：camera-CCDn.json代表CCDn相机view下的ROI

      - name：测点名字
      - lx：左上角点的x
      - ly：左上角点的y

- rx：右下角点的x
- ry：右下角点的y

2. 图像预处理：
   - 将ROI转为灰度图，（高斯滤波，拉普拉斯变换？）
   - 灰度拉伸

     $$I(x,y) = \frac{I(x,y) - I_{min}}{I_{max} - I_{min}}(MAX - MIN) + MIN$$

     灰度拉伸
   - 对比度增强

     ```
     Emphasize(ho_ImageScaleMax, &ho_ImageEmphasize, 7, 7, 1);
     ```

3. 模板匹配：
   - 读取模板
   - 将现有的，处理过的ROI与带scale的模板进行匹配，获取到模板的ID（如果分数过低，则去除，<0.49）
   - 对获取到的模板进行仿射变换affine，有SRT（plot为绿色）
   - 对模板轮廓进行平移缩放处理，获得一个环形ROI
   - 对该使用deriche算法进行边缘轮廓亚像素提取
   - 2d的ICP，将模板匹配的结果往边缘检测的结果上ICP，计算transformation
   - 使用ICP再次对模板轮廓进行平移缩放处理，矫正（plot为红色）

初始三角化

1. 首先剔除匹配分数较低的观察，不需要进行三角化
2. 进行三角化
   - 2个view的三角化，采用中点法，INVERSE_DEPTH_WEIGHTED_MIDPOINT
   - N个view的三角化

   code:

```
        // 三个view及以上的三角化
        if (Ps.size() > 2) {
            TriangulateNView(xs, Ps, &X);
            //记录3D点值
            pair.X[0] = X.hnormalized()[0];
            pair.X[1] = X.hnormalized()[1];
            pair.X[2] = X.hnormalized()[2];
        }
        // 两个view的三角化
        else if (Ps.size() == 2) {
            Vec3 X_two;
            Triangulate2View(
                Rs[0], Ts[0], Ks[0].inverse() * xs.col(0),
                Rs[1], Ts[1], Ks[1].inverse() * xs.col(1),
                X_two,
                // 中点法
                ETriangulationMethod::INVERSE_DEPTH_WEIGHTED_MIDPOINT);
            //记录3D点值
            pair.X[0] = X_two[0];
            pair.X[1] = X_two[1];
```

```
            pair.X[2] = X_two[2];
        }
```

3. 计算重投影误差，对于重投影误差过大的点，将其剔除（有可能是其2d检测有问题）。阈值为10，即重投影误差>10的view，则将其删除。

BA：优化3d点

1. 三角化
    - 2个view的三角化
    - N个view的三角化
2. 优化3d点

    进行BA优化：

```
    const bool bVerbose = true;
    const bool bMultithread = false;
    std::shared_ptr<Bundle_Adjustment_EB42X> ba_object =
        std::make_shared<Bundle_Adjustment_EB42X>(
            Bundle_Adjustment_EB42X::BA_EB42X_options(bVerbose,
bMultithread));


    ba_object->AdjustIndividual(sfm_data,
        EB42X_Optimize_Options(
            EB42X_Structure_SRT_Parameter_Type::NONE,
            Intrinsic_Parameter_Type::NONE,
            Extrinsic_Parameter_Type::NONE,
            EB42X_Structure_Parameter_Type::ADJUST_ALL),
        // landmark
        std::map<openMVG::IndexT, std::string>(),
    );
```

    target:

    - reprojection error

$$
\begin{aligned}
[x, y, 1] &= K[R|t][X, Y, Z, 1] \\
reprojection\ \ error &= sqrt((x - x')^2 + (y - y')^2) \\
x'\ \ and\ \ y'\ \ &is\ \ the\ \ point\ \ detected
\end{aligned}
$$

    parameters:

    - X
    - Y
    - Z

7Dof(7 Degree of freedom)配准，计算RTS

1. 将重建点和三坐标测量值进行RTS配准
    - 先算了一个两组点云的初始RTS，ICP算法，用Umeyama算法求解对应点关系矩阵

      Umeyama in Eigen
    - 对两组点云进行细配准，用LM算法优化

    计算出三个参数：

    重建点到三坐标测量值的旋转R，自由度：3

    重建点到三坐标测量值的平移t，自由度：3

    重建点到三坐标测量值的缩放s，自由度：1

    这里用了openmvg提供的函数，没有使用PCL

```
    if (!use_pcl) {
        // Compute the Similarity transform
        //1.找到 重建点和三坐标测量值之间的srt-这是openmvg提供的函数。
        FindRTS(x2, x1, &params.Sc, &params.tc, &params.Rc);
        Refine_RTS(x2, x1, &params.Sc, &params.tc, &params.Rc);

        //2.找到 重建点和名义值之间的srt（sca,rot,tra）
        FindRTS(x2, x3, &params.Sc_norminal, &params.tc_norminal,
&params.Rc_norminal);
        Refine_RTS(x2, x3, &params.Sc_norminal, &params.tc_norminal,
&params.Rc_norminal);
    }
```

2. 将RTS叠加到相机外参

```
    //将CAD坐标变换参数先叠加到相机内外参
    Mat3 R = params.Rc;
    double S = params.Sc;
    Vec3 t = params.tc;
    const openMVG::geometry::Similarity3 sim(openMVG::geometry::Pose3(R, -
R.transpose() * t / S), S);
    openMVG::sfm::ApplySimilarity(sim, sfm_data_calibr);
```

BA：利用三坐标测量值优化相机外参

~~**添加一圈点：为基准孔添加一圈点，会增加N*36个landmark**~~

1. ~~每个孔都会根据直径生成一圈点，但是以下的孔不会进行一圈点的生成：~~
    - ~~信息不全的孔，包括X、Y、Z以及D（基准孔X4X5除外）~~
    - ~~孔心不参与BA的不加入一圈点（螺纹孔）~~
2. ~~生成一圈点的过程是通过移动角度（360/N），通过三坐标测量值的直径生成一圈点~~

利用三坐标的测量值作为孔心点的优化初值

遍历三坐标测量值，赋值孔心点的坐标

迭代优化 epoch = 5

1. ~~优化工件整体的SRT和Z坐标~~
    - ~~SRT是工件相对于所有相机的位姿~~
    - ~~对于法向为Z的孔，调整其Z坐标，对于法向为Y的孔，调整其Y坐标~~

2. 优化相机的外参和Z坐标/Y坐标
    - 相机的外参是各个相机之间的位置
    - 对于法向为[0, 0, 1]的孔，调整其Z坐标，对于法向[0, 1, 0]的孔，调整其Y坐标

    加入一些约束（可选）：
    - XYZ基准

- [x] X4X5加权：

    50 * 重投影误差 or 10 * 重投影误差

    因为XYZ基准的建系需要以X4为原点，X4X5为Y轴，所以给X4X5加权可以使建系更准

- 小基准
    - [x] 激光网格点约束：

        重投影误差（pixel）+网格点到三坐标拟合出来的平面方程的距离（mm）

        权重分配：1pixel=1mm

        因为
        - 小基准的建系需要用到激光网格点拟合的平面，加权则会使小基准的建立更准
        - 增加了特征点

    - [x] G1G2加权

        因为小基准的建系需要用到G1G2，所以给G1G2加权可以使建系更准

    - [x] X4X5加权

        因为：？

        实验得出

- 大小圆
    - [x] 四个点Z均值约束：

        重投影误差 + abs（Z-Z'）
    - [x] 大小圆生成一圈点加Z均值约束

更新测量值

将优化后的测量值更新到三坐标测量值中去。对于Z孔，更新Z；对于Y孔，更新Y。

```
  //外参优化结束，更新测量的Z值
    for (auto& m : params.measureXYZ) {
        //Roi区域
        string roi_name = m.first;
        IndexT feat_id = params.landmarkIDs[roi_name];
        Vec3 old_v = m.second;
        //获取调整后的X
        Vec3 X = sfm_data.structure[feat_id].X;
        //进行坐标转换
        //Vec3 new_v = (1.0 / params.Sc_inv) * (params.Rc_inv.inverse() * (X -
params.tc_inv)).transpose();

        cout << feat_id << endl;
        cout << roi_name << "_dx: " << old_v[0] - X[0] * 1000 << endl;
        cout << roi_name << "_dy: " << old_v[1] - X[1] * 1000 << endl;
        cout << roi_name << "_dz: " << old_v[2] - X[2] * 1000 << endl;

        //更新调整后的测量值
        m.second[0] = X[0] * 1000;
        m.second[1] = X[1] * 1000;
        m.second[2] = X[2] * 1000;
    }
```

Other Ideas

- [x] 单task相机参数隔离

  各个task之间的相机参数单独优化，保存，重建，测量

  task

  - XYZ基准
  - 小基准
  - 大小圆

- [ ] 单孔相机参数隔离

  对于单个孔，都保存一份相机参数。对单个孔的相机参数进行单独的标定优化，保存，重建，测量。

  在标定优化A孔的时候，对A孔单独加权，加权尝试过两种方式，高斯加权效果会好一些

  - [ ] 仅对A孔加50倍权重，其他孔权重为1
  - [ ] 对A孔加50倍权重，其他孔权重取决于距A孔的距离，呈高斯分布，上下表面的孔权重不能传递

$$Weight = W_{max} * e^{-\frac{distance^2}{\alpha^2 r^2}}$$
$$\alpha = 0.6, r = 0.3$$

- [ ] 双工件联合标定

  Pipeline如上，输入为两个工件的2d匹配点，将两个工件的所有点加入到BA中进行标定优化

- [ ] 多工件移动联合标定

  Pipeline如上，输入为多个移动工件的2d匹配点，将两个工件的所有点加入到BA中进行标定优化

  将顶到头的工件称为初始工件，其他工件的重建点相对于初始工件的重建点有一个RT，将该RT加入到BA中进行优化

测量

2d检测

1. 图像去畸变，利用初始的畸变系数：k1，k2，k3，p1，p2
2. 利用halcon进行模板匹配：
   1. ROI：获取到每个ROI（手工标注的ROI）

      格式：camera-CCDn.json代表CCDn相机view下的ROI

      - name：测点名字
      - lx：左上角点的x
      - ly：左上角点的y
      - rx：右下角点的x
      - ry：右下角点的y

      标注 / 检查ROI软件link：

      Minth ROI Labelme Web

   2. 图像预处理：
      - 将ROI转为灰度图，（高斯滤波，拉普拉斯变换？）
      - 灰度拉伸

        $$I(x,y)=\frac{I(x,y)-I_{min}}{I_{max}-I_{min}}(MAX-MIN)+MIN$$

        灰度拉伸
      - 对比度增强

```
Emphasize(ho_ImageScaleMax, &ho_ImageEmphasize, 7, 7, 1);
```

   3. 模板匹配：
      - 读取模板
      - 将现有的，处理过的ROI与带scale的模板进行匹配，获取到模板的ID（如果分数过低，则去除，<0.49）
      - 对获取到的模板进行仿射变换affine，有SRT（plot为绿色）
      - 对模板轮廓进行平移缩放处理，获得一个环形ROI
      - 对该使用deriche算法进行边缘轮廓亚像素提取
      - 2d的ICP，将模板匹配的结果往边缘检测的结果上ICP，计算transformation
      - 使用ICP再次对模板轮廓进行平移缩放处理，矫正（plot为红色）

三角化

利用优化后的外参，重新三角化全部点

- 2个views的三角化
  - [ ] DLT(Direct Linear Transform)
  - [ ] L1_Angular
  - [ ] LInfinity_Angular
  - [x] Inverse_depth_weighted_midpoint
- N个views的三角化
  - [x] DLT(Direct Linear Transform)

    拿着3d点的坐标XYZ，往各个view下投影，得到2d坐标，列出方程组

    SVD求(超定)方程的解，得到最小二乘解
  - [ ] RANSAC(Random Sample Consensus)
    - 先从所有view中随机选2个view进行三角化，将该重建点往各个view下投，计算重投影误差，如果重投影误差小于一定阈值，则inliers+1，多做几组，从中选出inliers数量最大的那组inliers
    - 拿inliers进行三角化，SVD

计算

- 位置度

  计算公式：

$$\begin{cases} position = 2 * sqrt((x_m - x_N)^2 + (y_m - y_N)^2) & if \quad norm = [0,0,1] \\ position = 2 * sqrt((y_m - y_N)^2 + (z_m - z_N)^2) & if \quad norm = [0,1,0] \end{cases}$$

  - XYZ基准的位置度
  - 小基准的位置度
- 平面度

$$flatness = 正差 + 负差$$

- 轮廓度

$$profile = abs(distance_{measure} - distance_{norminal})$$

  - 平面的轮廓度
  - 大小圆的轮廓度

对标三坐标测量值

1. 三坐标测量值提取
   - excel->json
   - txt->json

   将测点号的position，x，y，z，flatness，profile作为value，将测点号作为key
2. 三坐标重复性分析

多组三坐标测量值进行分析，计算两次的三坐标的差值，得到以下分析结果

- count
- mean
- std
- min
- <25%
- <50%
- <75%
- max
- hist plot

3. 对标

   计算三坐标测量值和我们的测量值的差值

   统计

   - <0.2百分比
   - <0.3百分比
   - <0.4百分比