
Table of Contents

1 简介	1.1
2 上手	1.2
2.1 环境	1.2.1
2.2 配置	1.2.2
2.3 编译	1.2.3
2.4 运行	1.2.4
3 结构	1.3
4 代码	1.4
4.1 VO	1.4.1
4.2 object	1.4.2
4.3 algo	1.4.3
4.4 tool	1.4.4
5 FAQ	1.5
6 附录	1.6

FarStructure简介

FarStructure是一套针对“大间隔”“有序”帧的简易三维重建框架。

大间隔是指相邻两帧位姿差异较大(相邻帧旋转角度 $>15^\circ$, 重合面积 $<80\%$ 时), 当遇到上述大间隔问题时, 常规三维重建算法会失效(目前实际只测量了infiniTAM算法), 故提出了FarStructure算法, 希望能够针对大间隔问题给出解决方案。

有序是指已知各帧的时空顺序, 其中时间顺序是指各帧的拍摄时间先后顺序已知, 空间顺序是指各帧位姿的旋转顺序已知且固定(顺时针或逆时针)

目前FarStructure算法已经基本完成Vision Odometry部分, 且具有以下特点:

1. 针对大间隔

相比于连续帧, 大间隔“帧间特征匹配质量”下降严重。为解决这一问题, 本算法使用了一种基于统计规律的特征匹配过滤算法GMS, 有效的提升了大间隔情况下“帧间特征匹配质量”。

另外, 考虑到大间隔帧间重合度较低导致的错误匹配, 本算法通过适当的图片裁剪技术进行优化, 也取得了不错的效果。

2. 针对纹理缺失

考虑实际场景纹理缺乏导致特征匹配失效的情况, 本算法提出了针对性的解决方案: 首先通过AR交互方式获取先验位姿, 其次利用先验位姿进行重投影获取弱旋转相邻帧, 最后利用ICP等纹理无关技术进行帧间配准, 进而获得相机位姿。

注: 此方案正在完成中

2. 算法模块化

为了方便算法各部分的组合调试, 本算法将整个三维重建(目前仅有视觉里程计)处理流程拆解成若干独立模块, 每个模块均可自由选择具体的实现算法, 例如“ExtractFeature”模块可选择的具体实现算法有: ORB, SIFT, SUFT。这样做的好处是方便进行算法组合测试, 也便于添加新的算法。

具体拆分结构如下: (未来会根据需要继续添加新的模块或向每个模块添加更多算法)

processImage	ExtractFeature	matchFeature	FilterMatches	EstimatePose	OptimizePose
cutImage	ORB	BFM	None	None	None
...	SIFT	flann	GMS	3D3DSVD	sparse3D3D
	SURT	...	RANSAC	3D2DPnp	sparse3D2D
	...		maxDis	3D2DPnpRansac	dense3D3D
		

3. 数据与算法分离

为方便与其他算法快速融合(例如将本算法Vision Odometry融入到infiniTAM算法中),本算法将数据结构与算法分离：本算法的所有的算法接口(主要在algo和object文件夹下)均不依赖本算法的数据结构(主要在object文件夹下), 即只要配置好环境依赖, 所有算法接口可直接调用。

- algo主要内容：核心流程算法

processImage	ExtractFeature	matchFeature	FilterMatches	EstimatePose	OptimizePose
图片预处理算法	特征点提取算法	特征匹配算法	匹配筛选算法	位姿估计算法	位姿优化算法

- tool主要内容：工具算法

math	cloud	image	match
数学类	点云类	图片类	匹配类

- object主要内容：数据结构

Camera	Frame	Feature	Match	Map
管理相机数据	管理每一帧数据	管理每一帧的特征点数据	管理相邻两帧特征匹配结果	管理全局点云

4. CreatFrame机制

大间隔有序帧重建过程中, 位姿计算非常重要, 为了能够准确且定量评估位姿计算结果, 特设立CreatFrame机制, 即第当前帧是由前一帧经过预定义空间变换得到, 而非实际拍摄得到, 这样预定义空间变换即为可作为定量评估依据的理论相对位姿, 进而可准确且定量的评估位姿计算结果, 方便调试及优化算法。

运行

CMake工程

环境

- opencv
 - 版本: stable 4.1.0
- opencv_contrib
 - 版本: stable 4.1.0
- pcl
 - 版本: stable 1.9.1
- Eigen
 - 版本: stable 3.3.7
- g2o
 - 版本: stable 2017-07-30
- Sophu
 - 版本: [v1.0.0](#)

配置

配置文件路径

[工程主目录]/file/parameters.txt

配置文件内容

- vo参数

```
#开始帧
startIndex=0

#结束帧
endIndex=1

#VO运行类型
voType=read
#read:逐帧读取
#creat:生成帧)
```

- test参数

```
# mix测试结果保存位置
mixDir=/Users/longqi/Desktop/project/data/goodScanSlam/FarStructure/tuyang1/mix/
```

- object参数

```
# camera参数
    # 相机参数文件所在文件夹
calib_find_dir=/Users/longqi/Desktop/project/data/goodScanSlam/FarStructure/tuyang1/camera/
    # 相机参数文件名
calib_find_preName=calib
    # 相机参数文件名后缀
calib_find_extName=.txt
```

```
# 相机参数文件格式：内参是否是先rgb后depth
Interior_rgbThenDepth=true
# 相机参数文件格式：外参是否是depth2rgb
Extrinsic_depth2rgb=false
```

```
# frame参数
    # readFream:读取图片时相关参数
        # rgb
            # color图所在文件夹
rgb_read_dir=/Users/longqi/Desktop/project/data/goodScanSlam/FarStructure/tuyang1/frame/
            # color图文件名前半部分(固定部分)
rgb_read_preName=color
            # color图文件名后半部分(变化部分)
rgb_read_formatName=%5i
            # color图文件名后缀(文件格式)
```

```

rgb_read_extName=.png

# depth
# depth图所在文件夹
depth_read_dir=/Users/longqi/Desktop/project/data/goodScanSlam/FarStructure/tuyang1/frame/
# depth图文件名前半部分(固定部分)
depth_read_preName=depth
# depth图文件名后半部分(变化部分)
depth_read_formatName=%5i
# depth图文件名后缀(文件格式)
depth_read_extName=.png

# creatFream:生成图片时候相关参数
# 角度, 单位是°
angle=45
# 半径, 单位mm
r=700

# matches参数
#before filter:筛选前
# matches结果保存位置
matches_save_dir=/Users/longqi/Desktop/project/data/goodScanSlam/FarStructure/tuyang1/match/
# matches结果文件名
matches_save_preName=MatchesBeforeFilter
# matches结果文件后缀
matches_save_extName=.jpg

#after filter:筛选后
# matches结果保存位置
matches_filter_save_dir=/Users/longqi/Desktop/project/data/goodScanSlam/FarStructure/tuyang1/match/
# matches结果文件名
matches_filter_save_preName=MatchesAfterFilter
# matches结果文件后缀
matches_filter_save_extName=.jpg

# map参数
# map结果保存位置
map_save_dir=/Users/longqi/Desktop/project/data/goodScanSlam/FarStructure/tuyang1/map/
# map结果文件名
map_save_preName=map
# map结果文件后缀
map_save_extName=.pcd
# map文件尺度
map_CoordinateSystem_scale=1

```

- step参数

```

# processImage参数
# 是否进行图片剪切
ifCut=true
# 左剪切大小(像素数)
leftCutPixels=650
# 右剪切大小(像素数)
rightCutPixels=650

# extractFeature参数
# "特征提取"实现方法
extractFeatureMethod=ORB
# ORB
# SIFT
# SURT
# "特征提取"数量
featureNums=100000

# MatchFeature参数
# "特征匹配"方法
matchFeatureMethod=BFM
# BFM
# flann

# filterMatches参数
# "匹配过滤"方法

```

```
filterMatchMethod=GMS
## None:不过滤
## GMS
## RANSAC
## maxDis:极大值抑制
# 极大值抑制阈值
maxDisThreshold=4
# estimatePose参数
# 位姿估计
estimatePoseMethod=3D3DSVD
## none:不估计, 输出位姿等于输入位姿
## 3D3DSVD:将三维特征点匹配对带入观测方程, 并利用SVD分解求解出估计位姿
## 3D2DPnp:pnp(opencv接口)
## 3D2DPnpRansac:pnp+ransac(opencv接口)
# OptimizePose参数
# 位姿优化
optimizePoseMethod=dense3D3D
## None:不适用任何方法, 即使用给定的初始位姿作为估计结果
## sparse3D3D:利用稀疏点云(仅特征点)配准误差优化位姿 (使用g2o实现图优化)
## sparse3D2D:利用稀疏点云(仅特征点)重投影误差优化位姿 (使用g2o实现图优化)
## dense3D3D:ICP
```

- 其他

```
# log参数
# 参数文件保存文件夹路径
log_save_dir=/Users/longqi/Desktop/project/data/goodScanSlam/FarStructure/tuyang1/log/
```

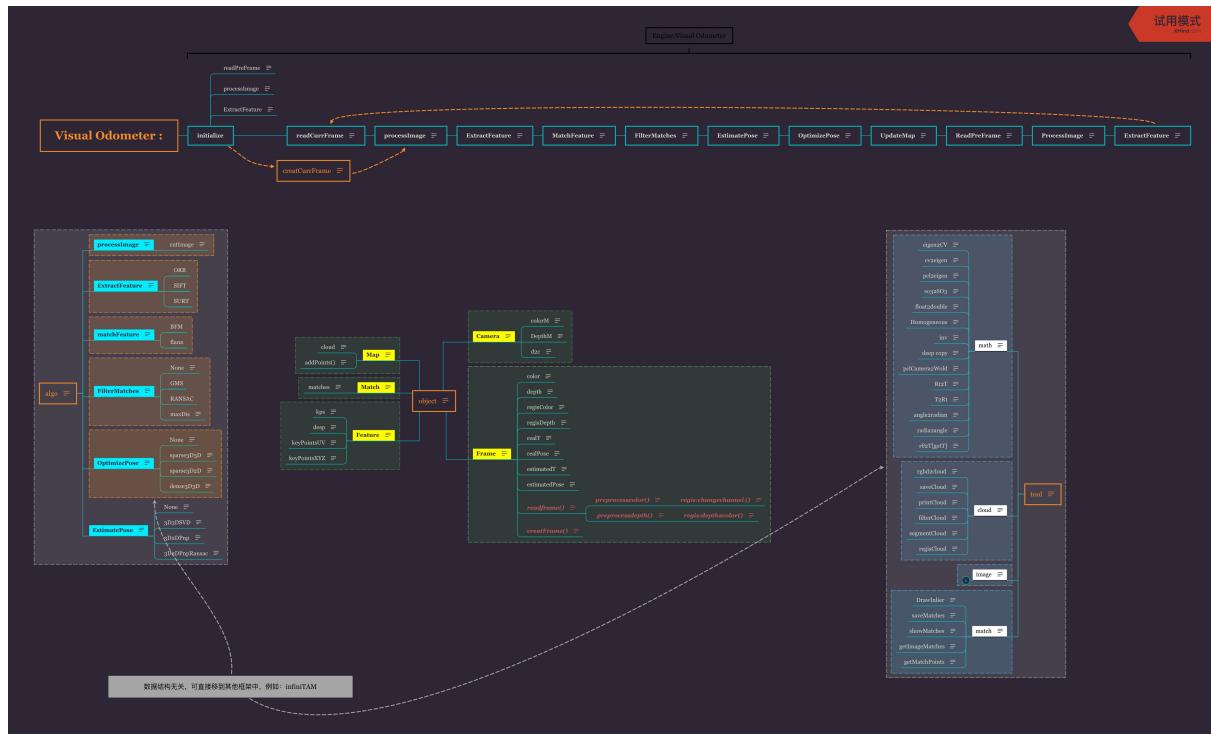
编译

1. \$ cd [工程主目录]
2. \$ mkdir build
3. \$ cd build/
4. \$ cmake ..
5. \$ make -j4

运行

1. \$ cd [工程主目录]
2. \$./apps/vo

结构



代码

代码主要算法与数据结构

Visual Odometer :

initialize

readPreFrame

读取第一帧,作为previous frame

processImage

处理previous frame

ExtractFeature

从previous frame中提取特征点

readCurrFrame

读取新的一帧, 作为current frame

processImage

处理current frame

ExtractFeature

从current frame 提取特征点

MatchFeature

将previous frame提取的特征点与current frame提取的特征点进行特征匹配, 得到match

FilterMatches

对匹配结果(match)进行过滤

EstimatePose

利用匹配结果(match)进行位姿估计

OptimizePose

利用match和原始数据(彩色图&&深度图)优化位姿

UpdateMap

利用计算位姿生成current frame的世界坐标系点云，并将点云融入到地图map中，以完成map的更新

ReadPreFrame

读取第一帧,作为previous frame

ProcessImage

处理previous frame

ExtractFeature

从previous frame中提取特征点

creatCurrFrame

利用previous frame和预先确定的方法生成current frame

object

数据结构

Camera

管理相机数据

- colorM
彩色相机内参
- DepthM
深度相机内参
- d2c
相机外参：深度相机到彩色相机的转换矩阵

Frame

管理每一帧数据

- color
彩色图
- depth
深度图
- regisColor
经过processImage模块处理过后，用于位姿估计和位姿优化的彩色图
- regisDepth
经过processImage模块处理过后，用于位姿估计和位姿优化的深度图
- realT
当前帧相对上一帧的真实转换矩阵：若当前帧是由creatCurrFrame生成的，则可得到具体的realT,否则自动置为单位矩阵
- realPose
当前帧真实位姿：若当前帧是由creatCurrFrame生成的，则可得到具体的realPose,否则自动置为单位矩阵
- estimatedT
当前帧相对上一帧的估计转换矩阵：由EstimatePose和OptimizePose模块计算得出.
- estimatedPose
估计位姿：由EstimatePose和OptimizePose模块计算得出.
- readFrame()
重要接口：从硬盘中读取1张彩色图和1张深度图作为新的一帧

- preprocessColor()

读取时需要对彩色图做一些预处理，例如通道转换

- regis:changeChannel ()

通道转换

- preprocessDepth()

读取时需要对彩色图做一些预处理，例如深度图对齐到彩色图

- regis:depth2color()

深度图对齐到彩色图

- creatFrame()

重要接口：利用previous frame和预先确定的方法生成current frame

Feature

管理每一帧的特征点数据

- kps

特征点(std::vector<_keypoint class="calibre25">)

- desp

描述子

- keyPointsUV

经过FilterMatches模块筛选后的特征点的2D形式(std::vector<(cv::Point2f>)

- keyPointsXYZ

经过FilterMatches模块筛选后的特征点特征点3D形式(std::vector<_point3f class="calibre25">)

Match

管理相邻两帧特征匹配结果

- matches

匹配结果(std::vector<_dmatch class="calibre25">)

Map

管理全局点云

- cloud

点云

- addPoints()

重要接口：向点云中添加特征点

algo

将整个三维重建(目前仅有视觉里程计)处理流程拆解成若干独立模块，每个模块均可自由选择具体的实现算法，例如“ExtractFeature”模块可选择的具体实现算法有：ORB，SIFT，SUFT。这样做的好处是方便进行算法组合测试，也容易添加新的算法。

processImage

第一步：图片预处理算法

- cutImage

剪切图片

ExtractFeature

第二步：特征点提取算法

- ORB
- SIFT
- SURT

matchFeature

第三步：特征匹配算法

- BFM
- flann

FilterMatches

第四步：匹配筛选算法

- None
不筛选
- GMS
- RANSAC
- maxDis

剔除距离高于阈值(可配置)的匹配

OptimizePose

第六：位姿优化算法

- None
不适用任何方法，即使用给定的初始位姿作为估计结果
- sparse3D3D
利用稀疏点云(仅特征点)配准误差优化位姿 (使用g2o实现图优化)

- sparse3D2D
利用稀疏点云(仅特征点)重投影误差优化位姿 (使用g2o实现图优化)
- dense3D3D
ICP

EstimatePose

第五步：位姿估计算法

- None
不使用任何方法，即使用给定的初始位姿作为估计结果
- 3D3DSVD
将三维特征点匹配对带入观测方程，并利用SVD分解求解出估计位姿
- 3D2DPnp
pnp(opencv接口)
- 3D2DPnpRansac
pnp+ransac(opencv接口)

tool

一些工具算法：根据具体需要不断动态添加

math

数学类：

- eigen2CV

Eigen格式转成OpenCV格式

- cv2eigen

OpenCV格式转成Eigen格式

- pcl2eigen

pcl格式转成Eigen格式

- so32SO3

李代数转换成李群

- float2double

Eigen中不支持自动类型提升，所以需要手动转换

- Homogeneous

齐次化与去其次化转换

- inv

相机内参，外参(刚体转换矩阵求逆)求逆

- deep copy

各种格式的数据的深拷贝通过模板重载统一实现

- pclCamera2World

将PCL点云库的点云坐标或3D点坐标从相机系转换成世界系

- Rt2T

由33旋转矩阵和31平移向量生成4*4的刚体变换矩阵

- T2Rt

由44的刚体变换矩阵生成33旋转矩阵和3*1平移向量

- angle2radian

角度转换成弧度，例如180°转换成 π (pi)

- radian2angle

弧度转换成角度，例如 π (pi)转换成180°

- rtheta2T[getT]

由旋转半径r和旋转角度θ生成刚体变换矩阵

cloud

点云类

- rgbd2cloud
由配准好的color&&depth图生成点云
- saveCloud
将点云保存到文件中
- printCloud
逐个打印点云中的3D到log文件
- filterCloud
点云滤波(未完成)
- segmentCloud
点云分割(未完成)
- regisCloud
点云配准(未完成)

image

图片类

- pixel2Pixel
像素配准：输入(u,v,d)&&内参&&变换矩阵，输出对应的(u,v)
- depth2color
深度图配准到彩色图
- rgbd2rgbd
一帧color&&depth配准到另一帧color&&depth
- imageCut
图片剪切
- printColor
逐点打印彩色图到log文件
- printDepth
逐点打印深度图到log文件

match

匹配类

- DrawInlier
若是用GMS做的filterMatches，则在getImageMatches时候，需要调用DrawInlier.
- saveMatches

图片形式保存Matches结果

- `showMatches`

图片形式展示Matches结果

- `getImageMatches`

获取ImageMatches(Matches结果的图片形式)

- `getMatchPoints`

根据匹配结果， 获取成功匹配的特征点(2D形式&&3D形式)

FAQ

SURF和SIFT特征点为什么无法使用

SURF和SIFT因为专利原因，在OpenCV3以后被放置在opencv_contrib，且必须在编译时候打开相应开关，才能使用，所以必须要确保两点

1. 确保安装了opencv_contrib。
2. 确保打开了SURF和SIFT所在模块的编译开关

具体参见：

- OpenCV3.4.X中Nonfree模块的使用-SURF为例
- Mac上编译安装 opencv和opencv_contrib

GMS为什么无法使用

目前必须是ORB特征点+BFS特征匹配才能使用GMS，后期会根据需要解决这个问题

相机参数为什么会读取错误

默认的相机参数读取格式如下：

```

calib.txt — 已编辑
color 相机内参
width height
1280 960
fx 6.07 fy 7.5
61 Cx 21 70 Cy 435

depth 相机内参
width height
1280 960
fx 22 fy 22
68 Cx 43 58 Cy 277

外参: depth2color
0.999952 -0.00748643 -0.00638078 -23.87
0.0083754 R 0.988169 0.153141 0.403457
0.00515881 -0.153187 0.988184 -2.2299 单位:mm
T
0 0

```

常见的相机参数错误如下

- 内参格式错误：先depth相机后color相机。
- 外参格式错误：是color2depth，而非color2depth。
- 外参单位错误：单位不是mm，而是m。

有时候必须要调整相机参数默认读取方式，可以通过配置文件进行修改(详见“上手/配置”章节)

```
# 相机参数文件格式：内参是否是先rgb后depth
Interior_rgbThenDepth=true
# 相机参数文件格式：外参是否是depth2rgb
Extrinsic_depth2rgb=false
```

为什么图片读取不进来

图片无法读取的可能原因如下

- 图片格式不对：默认格式后缀为.ppm和.pgm，若是其他个格式，可直接在配置文件里修改：(详见“上手/配置”章节)

```
# color图文件名后缀(文件格式)
rgb_read_extName=.png
# depth图文件名后缀(文件格式)
depth_read_extName=.png
```

- 图片名称不对：默认图片命名格式如下,若是其他个格式，可直接在配置文件里修改：(详见“上手/配置”章节)

- 深度图：

```
* depth00000.pgm
* depth00001.pgm
* depth00002.pgm
...
```

对应的配置为：

```
# color图文件名前半部分(固定部分)
rgb_read_preName=color
# color图文件名后半部分(变化部分)
rgb_read_formatName=%5i
# color图文件名后缀(文件格式)
rgb_read_extName=.pgm
```

- 彩色图：

```
* color00000.ppm
* color00001.ppm
* color00002.ppm
...
```

对应的配置为：

```
# depth图文件名前半部分(固定部分)
depth_read_preName=depth
# depth图文件名后半部分(变化部分)
depth_read_formatName=%5i
# depth图文件名后缀(文件格式)
depth_read_extName=.ppm
```


附录

- GMS