

Git使用手册（江佩视觉实验室）

Git是非常好的分布式代码协作工作，大家常见的有github、gitlab等。这篇文章就是和大家分享一下git中常用的一些功能。

一、git 建立分支与分支合并

1. 分支创建

你在本地创建一个分支，并将你的代码传到该分支上。但是此时该分支就是 不为他人所见的。

```
#创建一个叫做"feature_x"的分支，并切换过去：
git checkout -b feature_x
# 分支切换，切换回主分支：
git checkout master
# 分支删除，再把新建的分支删掉：
git branch -d feature_x
```

将你的代码传到这个分支上

```
git push origin <branch name>
```

2. 更新与合并

既然我们自己建设的分支不为主分支（master/main）所见，因此我们要将分支合并到主分支之上，才能将代码贡献上去。

```
# 查看你当前所在的分支
git branch

#首先我们用主分支更新本地仓库至最新改动，
git pull
```

以在你的工作目录中 获取（fetch）并 合并（merge）远端的改动。要合并其他分支到你的当前分支（例如 master），执行：

```
git merge <branch>
```

如果主分支上的信息在你下载之后没有更新，那么上述命令就可以直接合并成功。但是如果其他小伙伴上传的代码改变了主分支，那么你在merge的时候，可能出现冲突（conflicts）。这时候就需要你修改这些文件来手动合并这些冲突（conflicts）。

```
改完之后，你需要执行如下命令以将它们标记为合并成功：
git add <filename>
在合并改动之前，你可以使用如下命令预览差异：
git diff <source_branch> <target_branch>
```

3. 标签

为提交代码添加标签是很好的习惯，特别是阶段性的成果。你可以执行如下命令创建一个叫做 1.0.0 的标签：

```
git tag 1.0.0 1b2e1d63ff
1b2e1d63ff 是你想要标记的提交 ID 的前 10 位字符。可以使用下列命令获取提交 ID：
git log
你也可以使用少一点的提交 ID 前几位，只要它的指向具有唯一性。
替换本地改动

假如你操作失误（当然，这最好永远不要发生），你可以使用如下命令替换掉本地改动：
git checkout -- <filename>
此命令会使用 HEAD 中的最新内容替换掉你的工作目录中的文件。已添加到暂存区的改动以及新文件都不会受到影响。
```

假如你想丢弃你在本地的所有改动与提交，可以到服务器上获取最新的版本历史，并将你本地主分支指向它：

```
# 之前把repo上的最新状态更新当前文件夹，你本地的修改将会被直接丢弃
git fetch origin
git reset --hard origin/master
```

提交不成果时，出现错误：*Your branch is ahead of 'origin/master' by 2 commits.*

遇到这种问题，表示在你之前已经有2个commit而没有push到远程分支上，所以需要先git push origin **将本地分支提到远程仓库。
此时也可以使用：

```
#这里的x表示的就是在这之前已经有多少次的提交，
git reset --hard HEAD~x
```

这句命令的意思就是直接回退到x 个commit之前（这两次提交所改动的代码都没有了，慎用）。

4. 实用小贴士

```
内建的图形化 git：
gitk
彩色的 git 输出：
git config color.ui true
显示历史记录时，每个提交的信息只显示一行：
git config format.pretty oneline
交互式添加文件到暂存区：
git add -i
```

二、Git上传大文件

1. 安装 git lfs

在ubuntu系统中，git本身是没有上传大文件的命令的，因此我们安装git lfs 来上传打文件

```
curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh
| sudo bash
sudo apt-get install git-lfs
sudo git lfs install
```

2. 尝试一下

在ORB_SLAM2中Vocabulary文件夹中，有一个大文件“ORBvoc.txt”，我们把这个文件传到云端的repository中。

```
# 这个指令是把大文件添加到 lfs 中
git lfs track "Vocabulary/ORBvoc.txt"

# 然后可以 ll 看一下，当前目录应该会多出一个 .gitattributes文件，使用git add 加入缓存
git add .gitattributes

# 在加入文件夹中的其他的文件
git add Vocabulary/
git commit -m "first commit"
git push origin main
```

上传成功之后，你可以在terminal上看到：

```
Uploading LFS objects: 100% (1/1), 145 MB | 0 B/s, done.
Counting objects: 5, done.
Delta compression using up to 12 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 40.57 MiB | 354.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/xxxx/DensePlanarSLAM.git
   efb4ce6..8bcd1fc  main -> main
```