目　录

1 所需要的库及其依赖关系

1. zlib

2. libpng（依赖于 zlib）

3. libjpeg

4. x264

5. yasm

6. libv4l

7. xvid

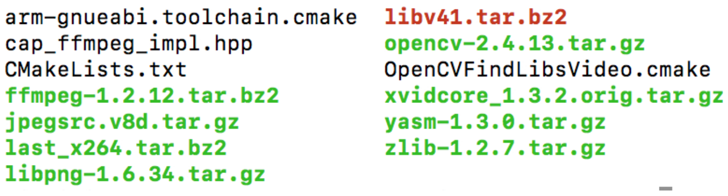
8. ffmpeg（依赖于 x264 与 xvid）

9. opencv-2.4.13（依赖于 ffmpeg 与 zlib）

2 准备工作

1. 找到交叉编译器的路径，如/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-gcc

2. 准备好所有安装包，如图 1所示。



```
arm-gnueabi.toolchain.cmake  libv41.tar.bz2
cap_ffmpeg_impl.hpp          opencv-2.4.13.tar.gz
CMakeLists.txt               OpenCVFindLibsVideo.cmake
ffmpeg-1.2.12.tar.bz2        xvidcore_1.3.2.orig.tar.gz
jpegsrc.v8d.tar.gz           yasm-1.3.0.tar.gz
last_x264.tar.bz2            zlib-1.2.7.tar.gz
libpng-1.6.34.tar.gz
```

图 1: backup 文件夹内容

- 需要用 CMakeLists.txt 覆盖 Opencv 目录下原 CMakeList.txt 文件（4 个平台均如此）。

- cap_ffmpeg_impl.hpp 以及 OpenCVFindLibsVideos.cmake 在安卓的两个平台配置时需要使用(因为编译安卓系统的库时,Opencv 默认会关闭 FFMPEG 这个依赖库的选项，如果手动修改配置文件，将其打开，会出现一些报错。cap_ffmpeg_impl.hpp 是为了解决其中一个报错“找不到头文件 sys/sys***.h”而准备的，该文件在 Opencv 目录下，具体位置请通过 find 命令搜寻）

- OpenCVFindLibsVideos.cmake 是为了打开 FFMPEG 这个依赖库的选项而准备的。
- arm-gnueabi.toolchain.cmake 是为了配置 armv7-linux 平台的环境准备的。

3 各个平台配置

3.1 安装设置

注：以下的所有配置请一段一段地复制到命令行，可能过程中会出现一些小问题。复制前注意检查代码是否需要修改。

3.1.1 armv7-linux 平台

```
CC1='/usr/bin/arm-linux-gnueabihf-gcc'
LD='/usr/bin/arm-linux-gnueabihf-ld'
AR='/usr/bin/arm-linux-gnueabihf-ar'
RANLIB='/usr/bin/arm-linux-gnueabihf-ranlib'
STRIP='/usr/bin/arm-linux-gnueabihf-strip'
PREFIX='/home/liuzili/arm-linux-build' # 安装目录
HOST='arm-linux' # 编译出的库在什么平台上运行
```

3.1.2 aarch64-linux 平台

```
CC1='/opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-gcc'
LD='/opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-ld'
AR='/opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-ar'
RANLIB='/opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-ranlib'
STRIP='/opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-strip'
PREFIX='/home/liuzili/linux-aarch64-build'
HOST='aarch64-linux'
export LDFLAGS="-L$PREFIX/lib"
export CPPFLAGS="-I$PREFIX/include"
YOUR_BUILD_PATH='/home/liuzili/linux-aarch64'
TOOLCHAINS='/opt/toolchains/linux-toolchain-aarch64'
CROSS_PREFIX="$TOOLCHAINS/bin/aarch64-linux-gnu-"
ARCH="aarch64"
```

3.1.3 armv7-android 平台

```
CC1='/opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-gcc'
LD='/opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-ld'
AR='/opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-ar'
RANLIB='/opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-ranlib'
STRIP='/opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-strip'
PREFIX='/home/liuzili/armv7-android-build'
HOST='arm-linux-android'
```

```
TOOLCHAINS='/opt/toolchains/android-toolchain-armv7'
SYSROOT="$TOOLCHAINS/sysroot"
CROSS_PREFIX="$TOOLCHAINS/bin/arm-linux-androideabi-"
EXTRA_CFLAGS="-march=armv7-a -mfloat-abi=softfp -mfpu=neon -mthumb -D__ANDROID__ ...
    -D__ARM_ARCH_7__ -D__ARM_ARCH_7A__ -D__ARM_ARCH_7R__ -D__ARM_ARCH_7M__ ...
    -D__ARM_ARCH_7S__"
EXTRA_LDFLAGS="-nostdlib"
YOUR_BUILD_PATH="/home/liuzili/armv7-android"
```

3.1.4 aarch64-android平台

```
CC1='/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-gcc'
LD='/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-ld'
AR='/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-ar'
RANLIB='/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-ranlib'
STRIP='/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-strip'
PREFIX='/home/liuzili/test-aarch64/3rdpart-aarch64-install'
HOST='aarch64-linux-android'
TOOLCHAINS='/opt/toolchains/android-toolchain-aarch64'
SYSROOT="$TOOLCHAINS/sysroot"
CROSS_PREFIX="$TOOLCHAINS/bin/aarch64-linux-android-"
YOUR_BUILD_PATH='¬/ad-aarch64'
EXTRA_CFLAGS="-march=armv8-a -D__ANDROID__ -D__ARM_ARCH_8__ -D__ARM_ARCH_8A__"
EXTRA_LDFLAGS="-nostdlib"
```

3.2 zlib安装

zlib一般最先编译，因为libpng以及后面的ffmpeg的编译会依赖它。通过prefix指定安装目录，sed语句将Makefile文件中的“CC”, “LDSHARED”等项替换为交叉编译器的对应项。

3.2.1 armv7-linux平台

```
tar -zxvf zlib-1.2.7.tar.gz;
cd zlib-1.2.7/;
./configure --prefix=$PREFIX --shared;
sed -i "s|CC=gcc|CC=$CC1|g" Makefile;
sed -i "s|LDSHARED=gcc|LDSHARED=$CC1|g" Makefile;
sed -i "s|CPP=gcc|CPP=$CC1|g" Makefile;
sed -i "s|AR=ar|AR=$AR|g" Makefile;
sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" Makefile;
make;
make install;
cd ..;
```

3.2.2 aarch64-linux平台

```
tar -zxvf zlib-1.2.7.tar.gz;
```

```
cd zlib-1.2.7/;
./configure --prefix=$PREFIX --shared;
sed -i "s|CC=gcc|CC=$CC1|g" Makefile;
sed -i "s|LDSHARED=gcc|LDSHARED=$CC1|g" Makefile;
sed -i "s|CPP=gcc|CPP=$CC1|g" Makefile;
sed -i "s|AR=ar|AR=$AR|g" Makefile;
sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.2.3 armv7-android 平台

```
tar -zxvf zlib-1.2.7.tar.gz;
cd zlib-1.2.7/;
./configure --prefix=$PREFIX --shared;
sed -i "s|CC=gcc|CC=$CC1|g" Makefile;
sed -i "s|LDSHARED=gcc|LDSHARED=$CC1|g" Makefile;
sed -i "s|CPP=gcc|CPP=$CC1|g" Makefile;
sed -i "s|AR=ar|AR=$AR|g" Makefile;
sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.2.4 aarch64-android 平台

```
tar -zxvf zlib-1.2.7.tar.gz;
cd zlib-1.2.7/;
./configure --prefix=$PREFIX --shared;
sed -i "s|CC=gcc|CC=$CC1|g" Makefile;
sed -i "s|LDSHARED=gcc|LDSHARED=$CC1|g" Makefile;
sed -i "s|CPP=gcc|CPP=$CC1|g" Makefile;
sed -i "s|AR=ar|AR=$AR|g" Makefile;
sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.3 libpng 安装

编译 libpng 时需要设置环境变量指向安装 zlib 的目录，否则会提示找不到 zlib。同样需要修改 Makefile。

3.3.1 armv7-linux 平台

```
export LDFLAGS="-L$PREFIX/lib";
export CPPFLAGS="-I$PREFIX/include";
tar -zxvf libpng-1.6.34.tar.gz;
```

```
cd libpng-1.6.34/;
./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static …
    CC=arm-linux-gnueabihf-gcc;
sed -i "s|CC = gcc|CC = $CC1|g" Makefile;
sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.3.2 aarch64-linux 平台

```
tar -zxvf libpng-1.6.34.tar.gz;
cd libpng-1.6.34/;
export LDFLAGS="-L$PREFIX/lib"
export CPPFLAGS="-I$PREFIX/include"
./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static CC=$CC1;
sed -i "s|CC = gcc|CC = $CC1|g" Makefile;
sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.3.3 armv7-android 平台

```
tar -zxvf libpng-1.6.34.tar.gz;
cd libpng-1.6.34/;
./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static CC=$CC1;
sed -i "s|CC = gcc|CC = $CC1|g" Makefile;
sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.3.4 aarch64-android 平台

```
tar -zxvf libpng-1.6.34.tar.gz;
cd libpng-1.6.34/;
./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static CC=$CC1;
sed -i "s|CC = gcc|CC = $CC1|g" Makefile;
sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.4 libjpeg 安装

3.4.1 armv7-linux 平台

```
tar -zxvf jpegsrc.v8d.tar.gz;
cd jpeg-8d/;
./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static ...
    CC=arm-linux-gnueabihf-gcc;
sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.4.2 aarch64-linux 平台

```
tar -zxvf jpegsrc.v8d.tar.gz;
cd jpeg-8d/;
./configure --host=arm-linux --prefix=$PREFIX --enable-shared --enable-static CC=$CC1;
sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.4.3 armv7-android 平台

```
tar -zxvf jpegsrc.v8d.tar.gz;
cd jpeg-8d/;
./configure --host=arm-linux-android --prefix=$PREFIX --enable-shared ...
    --enable-static CC=$CC1;
sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.4.4 aarch64-android 平台

```
tar -zxvf jpegsrc.v8d.tar.gz;
cd jpeg-8d/;
./configure --host=arm-linux-android --prefix=$PREFIX --enable-shared ...
    --enable-static CC=$CC1;
sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.5 yasm 安装

3.5.1 armv7-linux 平台

```
tar -zxvf yasm-1.3.0.tar.gz;
cd yasm-1.3.0/;
CC=arm-linux-gnueabihf-gcc;
./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static;
sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.5.2 aarch64-linux 平台

```
tar -zxvf yasm-1.3.0.tar.gz;
cd yasm-1.3.0/;
./configure --host=$HOST --prefix=$PREFIX CC=$CC1;
sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.5.3 armv7-android 平台

```
tar -zxvf yasm-1.3.0.tar.gz;
cd yasm-1.3.0/;
./configure --host=arm-linux-android --prefix=$PREFIX CC=$CC1;
sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.5.4 aarch64-android 平台

```
tar -zxvf yasm-1.3.0.tar.gz;
cd yasm-1.3.0/;
./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static CC=$CC1;
sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
sed -i "s|AR = ar|AR = $AR|g" Makefile;
sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
make -j;
make install;
cd ..;
```

3.6 x264 安装

x264 的安装容易出现奇怪的问题，如果下面的命令出现问题，请谷歌上搜寻解决方案。

3.6.1 armv7-linux 平台

```
tar -xjf last_x264.tar.bz2;
cd x264-snapshot-20171201-2245/;
./configure --enable-shared --host=$HOST --disable-asm --prefix=$PREFIX ...
    CC=arm-linux-gnueabihf-gcc;
sed -i "s|CC=gcc|CC=$CC1|g" config.mak;
sed -i "s|AR=gcc-ar|AR=$AR|g" config.mak;
sed -i "s|RANLIB=gcc-ranlib|RANLIB=$RANLIB|g" config.mak;
sed -i "s|LD=gcc -o|LD=$CC1 -o|g" config.mak;
sed -i "s|STRIP=strip|STRIP=$STRIP|g" config.mak;
```

```
make -j;
make install;
cd ..;
```

3.6.2 aarch64-linux 平台

```
YOUR_BUILD_PATH='/home/liuzili/linux-aarch64'
TOOLCHAINS=/opt/toolchains/linux-toolchain-aarch64
CROSS_PREFIX=$TOOLCHAINS/bin/aarch64-linux-gnu-
ARCH=aarch64
tar -xjf last_x264.tar.bz2;
cd x264-snapshot-20171201-2245/;
./configure --cross-prefix=$CROSS_PREFIX \
--host=aarch64-linux \
--enable-pic \
--enable-shared \
--extra-ldflags="-nostdlib" \
--disable-cli \
--prefix=$PREFIX
sed -i "s|CC=gcc|CC=$CC1|g" config.mak;
sed -i "s|AR=gcc-ar|AR=$AR|g" config.mak;
sed -i "s|RANLIB=gcc-ranlib|RANLIB=$RANLIB|g" config.mak;
sed -i "s|LD=gcc -o|LD=$CC1 -o|g" config.mak;
make -j;
make install;
cd ..;
```

3.6.3 armv7-android 平台

```
tar -xjf last_x264.tar.bz2;
cd x264-snapshot-20171201-2245/;
./configure --prefix=$PREFIX \
--host=arm-linux-androideabi \
--sysroot=$SYSROOT \
--cross-prefix=$CROSS_PREFIX \
--extra-cflags="$EXTRA_CFLAGS" \
--extra-ldflags="$EXTRA_LDFLAGS" \
--enable-pic \
--enable-static \
--enable-shared \
--enable-strip \
--disable-cli \
--disable-win32thread \
--disable-avs \
--disable-swscale \
--disable-lavf \
--disable-ffms \
--disable-gpac \
--disable-lsmash
make clean;
make STRIP= -j8 install;
cd ..;
```

3.6.4 aarch64-android 平台

```
tar -xjf last_x264.tar.bz2;
cd x264-snapshot-20171201-2245/;
./configure --prefix=$PREFIX \
 --host=aarch64-linux-android \
 --sysroot=$SYSROOT \
 --cross-prefix=$CROSS_PREFIX \
 --extra-cflags="$EXTRA_CFLAGS" \
 --extra-ldflags="$EXTRA_LDFLAGS" \
 --enable-pic \
 --enable-static \
 --enable-shared \
 --enable-strip \
 --disable-cli \
 --disable-win32thread \
 --disable-avs --disable-swscale --disable-lavf --disable-ffms --disable-gpac ...
     --disable-lsmash
make clean;
make STRIP= -j8 install;
cd ..;
```

3.7 libv4l 安装

这个压缩包中为库文件，库文件直接拷贝到“安装目录/lib”文件夹下即可。armv7-linux 以外的平台不需要安装。

3.7.1 armv7-linux 平台

```
cp libv41.tar.bz2 $PREFIX/lib;
cd $PREFIX/lib;
tar -xjf libv41.tar.bz2;
rm -f libv41.tar.bz2;
cd -;
```

3.8 xvid 安装

注意 xviud 是在 build/generic 目录下进行操作，修改 platform.inc 文件中的内容（CC, AR, RANLIB），修改方式同前面修改 Makefile 一样，这个文件会作用于 Makefile。

3.8.1 armv7-linux 平台

```
tar -zxvf xvidcore_1.3.2.orig.tar.gz;
cd xvidcore-1.3.2/build/generic;
./configure --prefix=$PREFIX --host=$HOST --disable-assembly ...
    CC=arm-linux-gnueabihf-gcc;
sed -i "s|CC=gcc|CC=$CC1|g" platform.inc;
sed -i "s|AR=ar|AR=$AR|g" platform.inc;
sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" platform.inc;
make -j;
make install;
cd ../../..;
```

3.8.2 aarch64-linux 平台

```
tar -zxvf xvidcore_1.3.2.orig.tar.gz;
cd xvidcore-1.3.2/build/generic;
./configure --prefix=$PREFIX --host=arm-linux --disable-assembly CC=$CC1;
sed -i "s|CC=gcc|CC=$CC1|g" platform.inc;
sed -i "s|AR=ar|AR=$AR|g" platform.inc;
sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" platform.inc;
make -j;
make install;
cd ../../..;
```

3.8.3 armv7-android 平台

```
tar -zxvf xvidcore_1.3.2.orig.tar.gz;
cd xvidcore-1.3.2/build/generic;
./configure --prefix=$PREFIX --host=$HOST --disable-assembly CC=$CC1;
sed -i "s|CC=gcc|CC=$CC1|g" platform.inc;
sed -i "s|AR=ar|AR=$AR|g" platform.inc;
sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" platform.inc;
make -j;
make install;
cd ../../..;
```

3.8.4 aarch64-android 平台

```
tar -zxvf xvidcore_1.3.2.orig.tar.gz;
cd xvidcore-1.3.2/build/generic;
./configure --prefix=$PREFIX --host=arm-linux-android --disable-assembly CC=$CC1;
sed -i "s|CC=gcc|CC=$CC1|g" platform.inc;
sed -i "s|AR=ar|AR=$AR|g" platform.inc;
sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" platform.inc;
make -j;
make install;
cd ../../..;
```

3.9 ffmpeg 安装

3.9.1 armv7-linux 平台

ffmpeg 是用于 mpeg4 编解码的。在安卓平台一定要 enable-static，否则 Opencv 无法依赖 ffmpeg。

```
tar -xjf ffmpeg-1.2.12.tar.bz2;
cd ffmpeg-1.2.12/
./configure \
--enable-cross-compile \
--target-os=linux \
--cc=arm-linux-gnueabihf-gcc \
--arch=arm \
--enable-shared \
--disable-static \
--enable-gpl \
--enable-nonfree \
--enable-ffmpeg \
--disable-ffplay \
--enable-ffserver \
--enable-swscale \
--enable-pthreads \
--enable-yasm \
--disable-stripping \
--enable-libx264 \
--enable-libxvid \
--extra-cflags=-I$PREFIX/include \
--extra-ldflags=-L$PREFIX/lib \
--prefix=$PREFIX;
make -j;
make install;
cd ..;
```

3.9.2 aarch64-linux 平台

```
tar -xjf ffmpeg-1.2.12.tar.bz2;
cd ffmpeg-1.2.12/
./configure \
--enable-cross-compile \
--target-os=linux \
--cc=$CC1 \
--arch=arm64 \
--enable-shared \
--disable-static \
--enable-gpl \
--enable-nonfree \
--enable-ffmpeg \
--disable-ffplay \
--enable-ffserver \
--enable-swscale \
--enable-pthreads \
--enable-yasm \
--disable-stripping \
--enable-libx264 \
--enable-libxvid \
--extra-cflags="-I$PREFIX/include" \
--extra-ldflags="-L$PREFIX/lib" \
--prefix=$PREFIX;
make -j;
make install;
cd ..;
```

3.9.3 armv7-android 平台

```
tar -xjf ffmpeg-1.2.12.tar.bz2;
cd ffmpeg-1.2.12/
./configure \
--enable-cross-compile \
--target-os=linux \
--cc=$CC1 \
--arch=arm \
--enable-shared \
--enable-static \
--enable-gpl \
--enable-nonfree \
--enable-ffmpeg \
--disable-ffplay \
--enable-ffserver \
--enable-swscale \
--disable-symver \
--enable-pthreads \
--disable-yasm \
--disable-stripping \
--enable-libx264 \
--enable-libxvid \
--extra-cflags=-I$PREFIX/include \
--extra-ldflags=-L$PREFIX/lib \
--prefix=$PREFIX;
make -j;
make install;
cd ..;
```

3.9.4 aarch64-android 平台

```
tar -xjf ffmpeg-1.2.12.tar.bz2;
cd ffmpeg-1.2.12/
./configure \
--enable-cross-compile \
--target-os=linux \
--cc=$CC1 \
--arch=arm64 \
--enable-shared \
--enable-static \
--enable-gpl \
--enable-nonfree \
--enable-ffmpeg \
--disable-ffplay \
--enable-ffserver \
--enable-swscale \
--disable-symver \
--enable-pthreads \
--disable-yasm \
--disable-stripping \
--enable-libx264 \
--enable-libxvid \
--extra-cflags=-I$PREFIX/include \
--extra-ldflags=-L$PREFIX/lib \
--prefix=$PREFIX;
make -j;
make install;
cd ..;
```

3.10 opencv 安装

在安卓平台请格外注意，需要手动进行一些替换操作。

3.10.1 armv7-linux 平台

```
tar -zxvf opencv-2.4.13.tar.gz;
cd opencv-2.4.13/;
export LD_LIBRARY_PATH=$PREFIX/lib/;
export PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig;
export PKG_CONFIG_LIBDIR=$PKG_CONFIG_LIBDIR:$PREFIX/lib/;
mv CMakeLists.txt CMakeLists_backup.txt;
cp ./../CMakeLists.txt ./;
mkdir -p build;
cd build;
mkdir -p install;
mkdir -p lib;
cp -r $PREFIX/lib/* install/;
cp -r $PREFIX/lib/* lib/;
cp ./../../arm-gnueabi.toolchain.cmake ./;
cmake -DCMAKE_TOOLCHAIN_FILE=arm-gnueabi.toolchain.cmake -DCMAKE_BUILD_TYPE=Release ...
    ..;
#sed -i "s|WITH_LIBV4L:BOOL=OFF|WITH_LIBV4L:BOOL=ON|g" CMakeCache.txt;
make -j;
make install;
cp -r lib/ ¬/lib
```

3.10.2 aarch64-linux 平台

```
tar -zxvf opencv-2.4.13.tar.gz;
cd opencv-2.4.13/;
export LD_LIBRARY_PATH=$PREFIX/lib/;
export PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig;
export PKG_CONFIG_LIBDIR=$PKG_CONFIG_LIBDIR:$PREFIX/lib/;
mv CMakeLists.txt CMakeLists_backup.txt;
cp ./../CMakeLists.txt ./;
mkdir -p build;
cd build;
mkdir -p install;
mkdir -p lib;
cp -r $PREFIX/lib/* install/;
cp -r $PREFIX/lib/* lib/;
cp ./../../arm-gnueabi.toolchain.cmake ./;
sed -i '$a\set(CMAKE_C_COMPILER ...
    /opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-gcc)' ...
    arm-gnueabi.toolchain.cmake
```

```
sed -i '$a\set(CMAKE_CXX_COMPILER …
    /opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-g++)' …
    arm-gnueabi.toolchain.cmake
sed -i '$a\set(CMAKE_FIND_ROOT_PATH "$YOUR_BUILD_PATH/opencv-2.4.13/build")' …
    arm-gnueabi.toolchain.cmake
sed -i '$a\set(link_directories(${PREFIX}/lib))' arm-gnueabi.toolchain.cmake
sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_PROGRAM NEVER)' arm-gnueabi.toolchain.cmake
sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_LIBRAY ONLY)' arm-gnueabi.toolchain.cmake
sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_INCLUDE ONLY)' arm-gnueabi.toolchain.cmake
cmake -DCMAKE_TOOLCHAIN_FILE=arm-gnueabi.toolchain.cmake -DCMAKE_BUILD_TYPE=Release …
    ..;
#sed -i "s|WITH_LIBV4L:BOOL=OFF|WITH_LIBV4L:BOOL=ON|g" CMakeCache.txt;
make -j;
make install;
cp -r lib/ ¬/lib
```

3.10.3 armv7-android 平台

```
tar -zxvf opencv-2.4.13.tar.gz;
cd opencv-2.4.13/;
export LD_LIBRARY_PATH=$PREFIX/lib/;
export PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig;
export PKG_CONFIG_LIBDIR=$PREFIX/lib/;
mv CMakeLists.txt CMakeLists_backup.txt;
cp ./../CMakeLists.txt ./;
mkdir -p build;
cd build;
mkdir -p install;
mkdir -p lib;
cp -r $PREFIX/lib/* install/;
cp -r $PREFIX/lib/* lib/;
cp  ./../platforms/android/android.toolchain.cmake ./;
sed -i '212iset(CMAKE_C_COMPILER …
    /opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-gcc)' …
    android.toolchain.cmake
sed -i '213iset(CMAKE_CXX_COMPILER …
    /opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-g++)' …
    android.toolchain.cmake
sed -i '214iset(CMAKE_FIND_ROOT_PATH "$YOUR_BUILD_PATH/opencv-2.4.13/build")' …
    android.toolchain.cmake
sed -i '215ilink_directories(${PREFIX}/lib)' android.toolchain.cmake
sed -i '216iset(CMAKE_FIND_ROOT_PATH_MODE_PROGRAM NEVER)' android.toolchain.cmake
sed -i '217iset(CMAKE_FIND_ROOT_PATH_MODE_LIBRAY ONLY)' android.toolchain.cmake
sed -i '218iset(CMAKE_FIND_ROOT_PATH_MODE_INCLUDE ONLY)' android.toolchain.cmake

export ANDROID_STANDALONE_TOOLCHAIN="/opt/toolchains/android-toolchain-armv7"
export PKG_CONFIG_PATH=$PREFIX/lib:$PKG_CONFIG_PATH

sed -i 's|OCV_OPTION(WITH_FFMPEG         "Include FFMPEG support" …
                        ON   IF (NOT ANDROID AND NOT IOS))|OCV_OPTION(WITH_FFMPEG …
            "Include FFMPEG support"                        ON   IF (ANDROID AND NOT …
    IOS))|g' ../CMakeLists.txt;
mv ../cmake/OpenCVFindLibsVideo.cmake ../cmake/OpenCVFindLibsVideo_backup.cmake
cp ../../OpenCVFindLibsVideo.cmake ../cmake/OpenCVFindLibsVideo.cmake
sed -i "s|include_directories( |include_directories("${PREFIX}/include" |g" …
    ../cmake/OpenCVFindLibsVideo.cmake
sed -i "s|REPLACEME|${PREFIX}|g" ../cmake/OpenCVFindLibsVideo.cmake
```

```

mv ../modules/highgui/src/cap_ffmpeg_impl.hpp ...
    ../modules/highgui/src/cap_ffmpeg_impl_backup.hpp
cp ../../cap_ffmpeg_impl.hpp ../modules/highgui/src/cap_ffmpeg_impl.hpp
sed -i "s|include_directories( |include_directories("${PREFIX}/include" |g" ...
    android.toolchain.cmake

cmake -DANDROID=1 -DCMAKE_TOOLCHAIN_FILE=android.toolchain.cmake ...
    -DBUILD_SHARED_LIBS=ON -DANDROID_ABI="armeabi-v7a" -DCMAKE_BUILD_TYPE=Release ..

#sed -i "s|WITH_LIBV4L:BOOL=OFF|WITH_LIBV4L:BOOL=ON|g" CMakeCache.txt;
make -j;
make install;
cp -r ./install/sdk/native/libs/armeabi-v7a ¬/lib
```

3.10.4 aarch64-android 平台

```
tar -zxvf opencv-2.4.13.tar.gz;
cd opencv-2.4.13/;
export LD_LIBRARY_PATH=$PREFIX/lib/;
export PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig;
export PKG_CONFIG_LIBDIR=$PKG_CONFIG_LIBDIR:$PREFIX/lib/;
mv CMakeLists.txt CMakeLists_backup.txt;
cp ./../CMakeLists.txt ./;
mkdir -p build;
cd build;
mkdir -p install;
mkdir -p lib;
cp -r $PREFIX/lib/* install/;
cp -r $PREFIX/lib/* lib/;
cp  ./../platforms/android/android.toolchain.cmake ./;
sed -i '$a\set(CMAKE_C_COMPILER ...
    /opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-gcc)' ...
    android.toolchain.cmake
sed -i '$a\set(CMAKE_CXX_COMPILER ...
    /opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-g++)' ...
    android.toolchain.cmake
sed -i '$a\set(CMAKE_FIND_ROOT_PATH "$YOUR_BUILD_PATH/opencv-2.4.13/build")' ...
    android.toolchain.cmake
sed -i '$a\link_directories(${PREFIX}/lib)' android.toolchain.cmake
sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_PROGRAM NEVER)' android.toolchain.cmake
sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_LIBRAY ONLY)' android.toolchain.cmake
sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_INCLUDE ONLY)' android.toolchain.cmake

export ANDROID_STANDALONE_TOOLCHAIN="/opt/toolchains/android-toolchain-aarch64"
sed -i 's|OCV_OPTION(WITH_FFMPEG         "Include FFMPEG support" ...
                         ON   IF (NOT ANDROID AND NOT IOS))|OCV_OPTION(WITH_FFMPEG ...
            "Include FFMPEG support"                        ON   IF (ANDROID AND NOT ...
    IOS))|g' ../CMakeLists.txt;
mv ../cmake/OpenCVFindLibsVideo.cmake ../cmake/OpenCVFindLibsVideo_backup.cmake
cp ../../OpenCVFindLibsVideo.cmake ../cmake/OpenCVFindLibsVideo.cmake
sed -i "s|include_directories( |include_directories("${PREFIX}/include" |g" ...
    ../cmake/OpenCVFindLibsVideo.cmake
sed -i "s|REPLACEME|${PREFIX}|g" ../cmake/OpenCVFindLibsVideo.cmake

mv ../modules/highgui/src/cap_ffmpeg_impl.hpp ...
    ../modules/highgui/src/cap_ffmpeg_impl_backup.hpp
```

```
cp ../../cap_ffmpeg_impl.hpp ../modules/highgui/src/cap_ffmpeg_impl.hpp
sed -i "s|include_directories( |include_directories("${PREFIX}/include" |g" …
    android.toolchain.cmake

cmake -DANDROID=1 -DCMAKE_TOOLCHAIN_FILE=android.toolchain.cmake …
    -DBUILD_SHARED_LIBS=ON DBUILD_ZLIB=ON -DANDROID_ABI="arm64-v8a" …
    -DCMAKE_BUILD_TYPE=Release ..
make -j;
make install;
cp -r ./install/sdk/native/libs/arm64-v8a ¬/lib
```