对于高分辨率遥感卫星数据而言，目前几乎都提供了有理函数模型（RFM）来进行图像校正（SPOT系列提供了有理函数模型之外还提供了严格轨道模型）。对遥感影像进行校正目前最常用的就是使用有理函数模型来进行校正。

有理函数模型的计算公式网上可以搜到，同时我之前的博客中也有比较详细的说明，可以参考这篇博客[http://blog.csdn.net/liminlu0314/article/details/24810593](http://blog.csdn.net/liminlu0314/article/details/24810593" \t "_blank)。

GDAL库从1.3左右就开始提供基于RPC的坐标转换，到目前2.x版本以来，基于RPC的坐标转换一直在持续优化，从1.X版本到2.X版本，计算速度有了较大提升。

下面是使用GDAL基于RPC坐标转换的代码：

#include "gdal\_priv.h"

#include "gdal\_alg.h"

void main()

{

CPLSetConfigOption("GDAL\_FILENAME\_IS\_UTF8", "NO");

CPLSetConfigOption("GDAL\_DATA", "./data");

GDALAllRegister();//注册驱动

const char\* pszTif = "F:\\ZY3\_TLC\_E115.5\_N39.8\_20150110\_L1A0002974893-NAD.tiff";

//打开数据

GDALDataset \*pDS = (GDALDataset\*)GDALOpen(pszTif, GA\_ReadOnly);

//从元数据中获取RPC信息

char\*\* papszRPC = pDS->GetMetadata("RPC");

//将获取的RPC信息构造成结构体

GDALRPCInfo oInfo;

GDALExtractRPCInfo(papszRPC, &oInfo);

GDALClose((GDALDatasetH)pDS);

//设置RPC模型中所需的DEM路径

char\*\* papszTransOption = NULL;

papszTransOption = CSLSetNameValue( papszTransOption, "RPC\_DEM", "E:\\DEM.img" ); //设置DEM

//使用RPC信息，DEM等构造RPC转换参数

void \*pRPCTransform = GDALCreateRPCTransformer(&oInfo, FALSE, 0, papszTransOption);

//定义图像的四个角点行列号坐标

double dX[4] = {0, 24525, 24525, 0};

double dY[4] = {0, 0, 24438, 24438};

double dZ[4] = {0, 0, 0, 0};

int nSuccess[4] = {FALSE};

//输出原始行列号坐标

printf("row col:\n");

for(int i=0; i<4; i++)

printf("%d\tX=%.7f\tY=%.7f\tZ=%.7f\n", i, dX[i], dY[i], dZ[i]);

//调用RPC坐标转换函数进行坐标转换反算（行列号转经纬度）

GDALRPCTransform(pRPCTransform, FALSE, 4, dX, dY, dZ, nSuccess);

//输出转换的经纬度

printf("long lat:\n");

for(int i=0; i<4; i++)

printf("%d\tX=%.7f\tY=%.7f\tZ=%.7f\n", i, dX[i], dY[i], dZ[i]);

//调用RPC坐标转换函数进行坐标正算（经纬度转行列号）

GDALRPCTransform(pRPCTransform, TRUE, 4, dX, dY, dZ, nSuccess);

//输出转换后的行列号坐标

printf("row2 col2:\n");

for(int i=0; i<4; i++)

printf("%d\tX=%.7f\tY=%.7f\tZ=%.7f\n", i, dX[i], dY[i], dZ[i]);

//释放资源

GDALDestroyRPCTransformer(pRPCTransform);

CSLDestroy(papszTransOption);

}