void CPhotoProcessView::Onoldpicture()

{

CString geometricFilepath = \_T("");

Mat src; //原图像

//打开图像文件，获取文件路径名

LPCTSTR szFilter = \_T("ALLSUPORTFILE(\*.\*)|\*.\*||");/\*BMP(\*.bmp)|\*.bmp|\*JPG(\*.jpg)|\*.jpg|\*/

CFileDialog dlgFileOpenImg(TRUE, NULL, NULL, OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT, szFilter, NULL);

//打开图像

if (dlgFileOpenImg.DoModal() == IDOK) {

//读取图像文件名

geometricFilepath = dlgFileOpenImg.GetPathName();

}

else {

return;

}

//为了将CString转到const cv::String

USES\_CONVERSION;

std::string str2(geometricFilepath.GetBuffer());

geometricFilepath.ReleaseBuffer();

src = imread(str2, CV\_LOAD\_IMAGE\_COLOR);

imshow("原图像", src);

int width=src.cols;

int heigh=src.rows;

RNG rng;

Mat img(src.size(),CV\_8UC3);

for (int y=0; y<heigh; y++)

{

uchar\* P0 = src.ptr<uchar>(y);

uchar\* P1 = img.ptr<uchar>(y);

for (int x=0; x<width; x++)

{

float B=P0[3\*x];

float G=P0[3\*x+1];

float R=P0[3\*x+2];

float newB=0.272\*R+0.534\*G+0.131\*B;

float newG=0.349\*R+0.686\*G+0.168\*B;

float newR=0.393\*R+0.769\*G+0.189\*B;

if(newB<0)newB=0;

if(newB>255)newB=255;

if(newG<0)newG=0;

if(newG>255)newG=255;

if(newR<0)newR=0;

if(newR>255)newR=255;

P1[3\*x] = (uchar)newB;

P1[3\*x+1] = (uchar)newG;

P1[3\*x+2] = (uchar)newR;

}

}

imshow("怀旧色",img);

waitKey();

imwrite("怀旧色.jpg",img);

}

void CPhotoProcessView::Onlianhuanhua()

{

CString geometricFilepath = \_T("");

Mat src; //原图像

//打开图像文件，获取文件路径名

LPCTSTR szFilter = \_T("ALLSUPORTFILE(\*.\*)|\*.\*||");/\*BMP(\*.bmp)|\*.bmp|\*JPG(\*.jpg)|\*.jpg|\*/

CFileDialog dlgFileOpenImg(TRUE, NULL, NULL, OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT, szFilter, NULL);

//打开图像

if (dlgFileOpenImg.DoModal() == IDOK) {

//读取图像文件名

geometricFilepath = dlgFileOpenImg.GetPathName();

}

else {

return;

}

//为了将CString转到const cv::String

USES\_CONVERSION;

std::string str2(geometricFilepath.GetBuffer());

geometricFilepath.ReleaseBuffer();

src = imread(str2, CV\_LOAD\_IMAGE\_COLOR);

imshow("原图像", src);

int width=src.cols;

int heigh=src.rows;

RNG rng;

Mat img(src.size(),CV\_8UC3);

for (int y=0; y<heigh; y++)

{

uchar\* P0 = src.ptr<uchar>(y);

uchar\* P1 = img.ptr<uchar>(y);

for (int x=0; x<width; x++)

{

float B=P0[3\*x];

float G=P0[3\*x+1];

float R=P0[3\*x+2];

float newB=abs(B-G+B+R)\*G/256;

float newG=abs(B-G+B+R)\*R/256;

float newR=abs(G-B+G+R)\*R/256;

if(newB<0)newB=0;

if(newB>255)newB=255;

if(newG<0)newG=0;

if(newG>255)newG=255;

if(newR<0)newR=0;

if(newR>255)newR=255;

P1[3\*x] = (uchar)newB;

P1[3\*x+1] = (uchar)newG;

P1[3\*x+2] = (uchar)newR;

}

}

Mat gray;

cvtColor(img,gray,CV\_BGR2GRAY);

normalize(gray,gray,255,0,CV\_MINMAX);

imshow("连环画",gray);

waitKey();

imwrite("连环画.jpg",gray);

}

void CPhotoProcessView::Onrongzhu()

{

CString geometricFilepath = \_T("");

Mat src; //原图像

//打开图像文件，获取文件路径名

LPCTSTR szFilter = \_T("ALLSUPORTFILE(\*.\*)|\*.\*||");/\*BMP(\*.bmp)|\*.bmp|\*JPG(\*.jpg)|\*.jpg|\*/

CFileDialog dlgFileOpenImg(TRUE, NULL, NULL, OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT, szFilter, NULL);

//打开图像

if (dlgFileOpenImg.DoModal() == IDOK) {

//读取图像文件名

geometricFilepath = dlgFileOpenImg.GetPathName();

}

else {

return;

}

//为了将CString转到const cv::String

USES\_CONVERSION;

std::string str2(geometricFilepath.GetBuffer());

geometricFilepath.ReleaseBuffer();

src = imread(str2, CV\_LOAD\_IMAGE\_COLOR);

imshow("src",src);

int width=src.cols;

int heigh=src.rows;

Mat gray0,gray1;

//去色

cvtColor(src,gray0,CV\_BGR2GRAY);

//反色

addWeighted(gray0,-1,NULL,0,255,gray1);

//高斯模糊,高斯核的Size与最后的效果有关

GaussianBlur(gray1,gray1,Size(11,11),0);

//融合：颜色减淡

Mat img(gray1.size(),CV\_8UC1);

for (int y=0; y<heigh; y++)

{

uchar\* P0 = gray0.ptr<uchar>(y);

uchar\* P1 = gray1.ptr<uchar>(y);

uchar\* P = img.ptr<uchar>(y);

for (int x=0; x<width; x++)

{

int tmp0=P0[x];

int tmp1=P1[x];

P[x] =(uchar) min((tmp0+(tmp0\*tmp1)/(256-tmp1)),255);

}

}

imshow("素描",img);

waitKey();

imwrite("素描.jpg",img);

}

void CPhotoProcessView::Onbingdong()

{

CString geometricFilepath = \_T("");

Mat src; //原图像

//打开图像文件，获取文件路径名

LPCTSTR szFilter = \_T("ALLSUPORTFILE(\*.\*)|\*.\*||");/\*BMP(\*.bmp)|\*.bmp|\*JPG(\*.jpg)|\*.jpg|\*/

CFileDialog dlgFileOpenImg(TRUE, NULL, NULL, OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT, szFilter, NULL);

//打开图像

if (dlgFileOpenImg.DoModal() == IDOK) {

//读取图像文件名

geometricFilepath = dlgFileOpenImg.GetPathName();

}

else {

return;

}

//为了将CString转到const cv::String

USES\_CONVERSION;

std::string str2(geometricFilepath.GetBuffer());

geometricFilepath.ReleaseBuffer();

src = imread(str2, CV\_LOAD\_IMAGE\_COLOR);

imshow("src",src);

Mat img;

src.copyTo(img);

int width=src.cols;

int heigh=src.rows;

Mat dst(img.size(),CV\_8UC3);

for (int y=0;y<heigh;y++)

{

uchar\* imgP=img.ptr<uchar>(y);

uchar\* dstP=dst.ptr<uchar>(y);

for (int x=0;x<width;x++)

{

float b0=imgP[3\*x];

float g0=imgP[3\*x+1];

float r0=imgP[3\*x+2];

float b = (b0-g0-r0)\*3/2;

float g = (g0-b0-r0)\*3/2;

float r = (r0-g0-b0)\*3/2;

r = (r>255 ? 255 : (r<0? -r : r));

g = (g>255 ? 255 : (g<0? -g : g));

b = (b>255 ? 255 : (b<0? -b : b));

dstP[3\*x] = (uchar)b;

dstP[3\*x+1] = (uchar)g;

dstP[3\*x+2] = (uchar)r;

}

}

imshow("冰冻",dst);

imwrite("冰冻.jpg",dst);

}

void CPhotoProcessView::muKeFilter(Mat &srcImage)

{

Mat dstImage = srcImage;

cvtColor(dstImage,dstImage,CV\_BGR2GRAY);

threshold(dstImage,dstImage,127,255,THRESH\_BINARY);

imshow("木刻滤镜",dstImage);

imwrite("木刻.jpg",dstImage);

}

void CPhotoProcessView::Onmuke()

{

CString geometricFilepath = \_T("");

Mat src; //原图像

//打开图像文件，获取文件路径名

LPCTSTR szFilter = \_T("ALLSUPORTFILE(\*.\*)|\*.\*||");/\*BMP(\*.bmp)|\*.bmp|\*JPG(\*.jpg)|\*.jpg|\*/

CFileDialog dlgFileOpenImg(TRUE, NULL, NULL, OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT, szFilter, NULL);

//打开图像

if (dlgFileOpenImg.DoModal() == IDOK) {

//读取图像文件名

geometricFilepath = dlgFileOpenImg.GetPathName();

}

else {

return;

}

//为了将CString转到const cv::String

USES\_CONVERSION;

std::string str2(geometricFilepath.GetBuffer());

geometricFilepath.ReleaseBuffer();

src = imread(str2, CV\_LOAD\_IMAGE\_COLOR);

imshow("src",src);

if(!src.data || src.empty()){

cout<<"读入图片错误！"<<endl;

}

muKeFilter(src);

waitKey(0);

}

float mSize = 0.5;

void CPhotoProcessView::Onyuhua()

{

CString geometricFilepath = \_T("");

Mat src; //原图像

//打开图像文件，获取文件路径名

LPCTSTR szFilter = \_T("ALLSUPORTFILE(\*.\*)|\*.\*||");/\*BMP(\*.bmp)|\*.bmp|\*JPG(\*.jpg)|\*.jpg|\*/

CFileDialog dlgFileOpenImg(TRUE, NULL, NULL, OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT, szFilter, NULL);

//打开图像

if (dlgFileOpenImg.DoModal() == IDOK) {

//读取图像文件名

geometricFilepath = dlgFileOpenImg.GetPathName();

}

else {

return;

}

//为了将CString转到const cv::String

USES\_CONVERSION;

std::string str2(geometricFilepath.GetBuffer());

geometricFilepath.ReleaseBuffer();

src = imread(str2, CV\_LOAD\_IMAGE\_COLOR);

imshow("src",src);

int width=src.cols;

int heigh=src.rows;

int centerX=width>>1;

int centerY=heigh>>1;

int maxV=centerX\*centerX+centerY\*centerY;

int minV=(int)(maxV\*(1-mSize));

int diff= maxV -minV;

float ratio = width >heigh ? (float)heigh/(float)width : (float)width/(float)heigh;

Mat img;

src.copyTo(img);

Scalar avg=mean(src);

Mat dst(img.size(),CV\_8UC3);

Mat mask1u[3];

float tmp,r;

for (int y=0;y<heigh;y++)

{

uchar\* imgP=img.ptr<uchar>(y);

uchar\* dstP=dst.ptr<uchar>(y);

for (int x=0;x<width;x++)

{

int b=imgP[3\*x];

int g=imgP[3\*x+1];

int r=imgP[3\*x+2];

float dx=centerX-x;

float dy=centerY-y;

if(width > heigh)

dx=(dx\*ratio);

else

dy=(dy\*ratio);

int dstSq = dx\*dx + dy\*dy;

float v = ((float) dstSq / diff)\*255;

r = (int)(r +v);

g = (int)(g +v);

b = (int)(b +v);

r = (r>255 ? 255 : (r<0? 0 : r));

g = (g>255 ? 255 : (g<0? 0 : g));

b = (b>255 ? 255 : (b<0? 0 : b));

dstP[3\*x] = (uchar)b;

dstP[3\*x+1] = (uchar)g;

dstP[3\*x+2] = (uchar)r;

}

}

imshow("羽化",dst);

waitKey();

imwrite("羽化.jpg",dst);

}