

hope_v

于 2016-04-06 16:47:05 发布

6781

收藏 11

文章标签：

opencv

mfc

显示图片-显示视频

OpenCV

专栏收录该内容

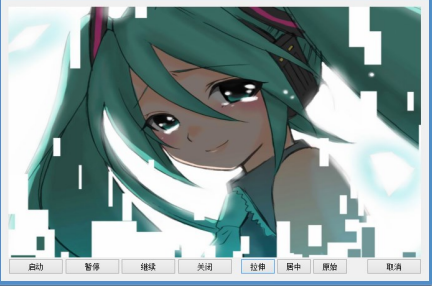
1 订阅

1 篇文章

订阅专栏


先看图：
拉伸显示

CvImgCtrl Demo



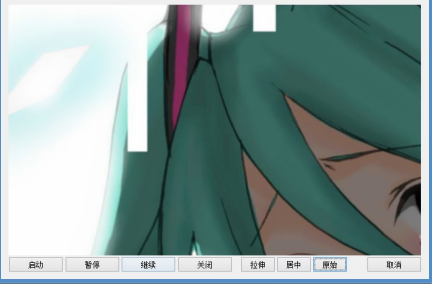
居中显示

CvImgCtrl Demo




原始大小显示

CvImgCtrl Demo



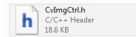
显示视频：

CvImgCtrl Demo



实现图片显示，只需要两句话：
//链接到控件
m_show.LinkDlgItem(IDC_Show_this);
//显示图像
m_show.SetImage(IStream("G:/image/miku.jpg"));

只需要包含一个头文件：



调整显示方式：

```
void CdemoDlg::OnBnClickedTensile()
{
    //设置为拉伸显示
    m_show.SetResizeType(CvImgCtrl::ResizeType_Tensile);
}

void CdemoDlg::OnBnClickedCenter()
{
    //设置为居中显示
    m_show.SetResizeType(CvImgCtrl::ResizeType_CenterResize);
}

void CdemoDlg::OnBnClickedRaw()
{
    //设置为原始大小显示
    m_show.SetResizeType(CvImgCtrl::ResizeType_Raw);
}
```

下面是CvImgCtrl.h的源代码，复制保存下来即可用，或者也可以下载demo，不要积分
控件采用的是双缓冲绘图，显示效率很高，不会闪烁，支持视频，图片的显示

```
1  //*****
2  /* OpenCV NVC图像显示控件 v1.0
3  /* 该控件很容易集成到MFC中显示，具体用法见demo。
4  /*
5  /* Wsh: 512690069
6  /* 2016年4月6日 15:28:58
7  //*****
8  #pragma once
9
10 #include <cv.h>
11 #include <highgui.h>
12 #include <afxwin.h>
13
14 //自定义消息基础值
15 #define MSG_BASE (WM_USER + 0x40)
16 #define MSG_CLICK (MSG_BASE + 0) //鼠标左键被按下的消息
17 #define MSG_MOVE (MSG_BASE + 1) //左键拖拽起来
18 #define MSG_SCROLL (MSG_BASE + 2) //右键按下
```

分类专栏	
Life	1篇
Matlab	
OpenCV & Matlab	
Visual C++	2篇
Linux	
ARM	
C#	
Machine Learning	
Computer Vision	
E Language	
Mathematics	
Algorithm	
Optimization Accelerate	
Robust	
Halcon	
Android	3篇
Cocos2dx	1篇
iOS	1篇

```
19 #define MsgMove (MsgBase + 3) //鼠标移动
20
21 #define MsgMouseDown (MsgBase + 4) //中键按下
22 #define MsgMButtonDown (MsgBase + 5) //中键抬起
23
24 #define MsgKeyDown (MsgBase + 6) //按下某个键盘按键
25 #define MsgDragFile (MsgBase + 7) //拖动文件到控件上去的时候
26
27 #define MsgVideoStart (MsgBase + 8) //视频开始的时候
28 #define MsgVideoOver (MsgBase + 9) //视频结束了
29
30
31 //得到一帧视频画面了，具体看demo
32 //1. 在dlg的头文件中写: afx_msg LRESULT onFrame(WPARAM wp, LPARAM lp);声明
33 //2. 在dlg的BEGIN_MESSAGE_MAP里面写: ON_MESSAGE(MsgVideoFrame, onFrame)
34 //3. 在dlg的cpp中写:
35 /*
36 LRESULT CdemoDlg::onFrame(WPARAM wp, LPARAM lp){
37     Mat frame((IplImage*)lp, false);
38     flip(frame, frame, -1);
39     //这里就是在处理每一帧的图像了，类似的MsgVideoPreDisplay消息也是这么来的
40     return 0;
41 }
42 */
43 #define MsgVideoFrame (MsgBase + 0xA)
44 #define MsgVideoPreDisplay (MsgBase + 0x08) //视频一帧画面准备显示了，他会在MsgVideoFrame之后
45
46 //控件被单击消息
47 #define MsgClick (MsgBase + 0x0C)
48
49 //内部消息，内部使用，不需要理会
50 #define InnerMsgBase (MsgBase + 0xF0) //基本信息起点
51 #define InnerMsgVideoFrame (InnerMsgBase + 0) //接受到一帧画面
52
53 //图像显示控件
54 class CImgCtrl : public CWnd
55 {
56     DECLARE_DYNAMIC(CvImgCtrl)
57
58 public:
59     //视图缩放类型
60     enum ResizeType
61     {
62         ResizeType_Raw, //原始大小，以左上角为锚，图像多大就多大，不能缩放
63         ResizeType_CenterResize, //居中缩放，图像宽高按比例，画面居中
64         ResizeType_Tensile //拉伸，宽高拉伸到控件一样大
65     };
66
67     //视频播放的状态
68     enum VideoState{
69         VideoState_Stop, //视频停止了
70         VideoState_Playing, //视频播放中
71         VideoState_Pause //视频暂停了
72     };
73
74 public:
75     CvImgCtrl();
76     virtual ~CvImgCtrl();
77
78     //链接到对话框上的控件，链接过后，就被这个类接管了，他会负责将画面绘制到控件上去
79     //Id: 控件的id
80     //parent: 父控件的指针，比如在dialog里面，给this就行了
81     BOOL linkDlgItem(UINT idd, CWnd* parent);
82
83     //创建一个控件，如果需要动态创建，就调用这个创建函数，而不是CWnd的创建函数
84     //pParent: 父类的指针，一般给Dialog的this指针就好了
85     //rcPos: 控件所在位置，可以直接给Rect()也行，然后在OnSize里边调整宽高，或者给指定值也行
86     //autoDelete: 如果CvImgCtrl* m_show, 这种，指针，采用new的方式得到的控件，而不是实体变量，那么当这个参数给true的时候，
87     // 会自己回收内存 (delete this)，也就是你就不用手动去delete他了
88     //resizeType: 缩放方式，当显示图像和窗口大小不一致时，该如何缩放，有3种方式，看枚举类InnerMsgBase+resizeType的备注
89     //defaultColor: 默认为黑色，即背景默认为黑色，比如是中显示的时候，四周会留空，或是什么颜色，或者给创建时设置什么颜色等
90     bool create(CWnd* pParent, CRect rcPos, bool autoDelete = false, ResizeType resizeType = ResizeType_Raw, Cv
91
92 //开始视频，这个控件内置了VideoCapture，所以这里可以直接调用openVideo来打开视频
93 //file: 视频文件路径，直接是VideoCapture.open，如果失败格式不支持，那就去看VideoCapture
94 //waitTime: 延迟时间，每一帧画面需要延迟的时间，单位毫秒，如果不延迟，画面会很快，当然你可以改成从VideoCapture中获取
95 bool openVideo(const char* file, unsigned int waitTime = 30);
96
97 //开始视频摄像头，这个控件内置了VideoCapture，所以这里可以直接调用openVideo来打开视频
98 //cameraIndex: 摄像头索引，一样是VideoCapture的open函数，open函数从-1会弹出选择视频对话框，0-n是你电脑个摄像头序号
99 //waitTime: 延迟时间，每一帧画面需要延迟的时间，单位毫秒，如果不延迟，画面会很快，当然你可以改成从VideoCapture中获取
100 bool openVideo(int cameraIndex = 0, unsigned int waitTime = 30);
101
102 //暂停视频
103 void pauseVideo();
104
105 //继续视频
106 void resumeVideo();
107
108 //关闭视频
109 void closeVideo();
110
111 //设置图像，设置后控件会刷新显示，会执行onUpdateImage，允许你设置任意图像，内部会自动调整
112 //如果设置的图像是相同大小、相同通道的图像，则只会进行copy，所以效率是杠杠的
113 void setImage(const cv::Mat img){setImage(&IplImage(img));}
114
115 //设置图像Ipl方式，设置后控件会刷新显示，会执行onUpdateImage
116 void setImage(const IplImage* img);
117
118 //获取当前图像
119 cv::Mat getImageMat(){return cv::Mat(m_img, false);}
120
121 //获取当前图像
122 IplImage* getImage(){return m_img;}
123
124 //更新显示图像，如果你通过getImageMat得到的Mat或者IplImage，你修改了他，那么请使用onUpdateImage来刷新画面应用到控件上
125 void onUpdateImage();
126
127 //获取显示图像，显示图像跟控件一样大小的那个图像，随着控件大小改变而改变
128 IplImage* getDisplayImage(){return m_imgResize;}
129
130 //获取显示图像Mat
131 cv::Mat getDisplayImageMat(){return cv::Mat(m_imgResize, false);}
132
133 //更新显示图像，当你通过getDisplayImage获取到的图像，并修改了他，那么请使用这个函数来刷新应用到画面上
134 void onUpdateDisplayImage();
135
136 //设置缩放方式，共3种方式，见ResizeType的定义上写的备注，设置后立马生效并刷新
137 void setResizeType(ResizeType rt){m_rt = rt; onUpdateImage();}
138
139 //设置数据，存一个int，以备特殊时候使用，比如要存什么Id之类的，指针之类的
140 void setData(int data){m_saveData = data;}
141 int getData(){return m_saveData;}
142
143 //获取视频播放数据
144 cv::VideoCapture* getVideoCapture(){return &m_videoCapture;}
145
146 //返回当前播放视频状态
147 VideoState getVideoState(){return m_videoState;}
148
149 private:
150     //设置图像，只是设置图像，不会显示到控件上去
151     void _onlySetImage(const IplImage* img);
152
153     //启动视频，device_or_file，如果给是-1或者0-1024以内的数字，则认为是camera
154     //否则认为是文件路径，waitTime则是帧延迟时间
155     bool _starVideo(const void* device_or_file, unsigned int waitTime = 30);
156
157     //重画Image到displayImage (即resizeImage)
158     void redrawResize();
159
160     //清空显示的那个图像
161     void voidResize();
162
163     //通知父类，跟this是this
164     LRESULT notifyParent(UINT msg, LPARAM lp = 0);
165
166 private:
167     cv::VideoCapture m_videoCapture;
168     VideoState m_videoState;
169     int m_videoWaitTime;
170     unsigned int m_lastDownTime;
171     int m_saveData;
172     IplImage* m_img;
173     IplImage* m_imgResize;
174     BITMAPINFO m_bmpHeader;
175     bool m_autoDelete;
176     CvScalar m_defaultColor;
177     ResizeType m_rt;
178     CRITICAL_SECTION m_resizeCS;
179
180 protected:
181     DECLARE_MESSAGE_MAP()
182
183 public:
```

```

183  afx_msg void OnSize(UINT nType, int cx, int cy);
184  afx_msg void OnDestroy();
185  afx_msg BOOL OnEraseBkgnd(CDC* pDC);
186  afx_msg void OnPaint();
187  afx_msg void OnNcDestroy();
188  afx_msg void OnMouseMove(UINT nFlags, CPoint point);
189  afx_msg void OnMouseDown(UINT nFlags, CPoint point);
190  afx_msg void OnMouseUp(UINT nFlags, CPoint point);
191  afx_msg void OnButtonDown(UINT nFlags, CPoint point);
192  afx_msg LRESULT OnNcHitTest(CPoint point);
193  afx_msg void OnMButtonDown(UINT nFlags, CPoint point);
194  afx_msg void OnMButtonUp(UINT nFlags, CPoint point);
195  afx_msg void OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);
196  afx_msg void OnDropFiles(HDROP hDropInfo);
197
198  afx_msg LRESULT OnVideoFrame(WPARAM wp, LPARAM lp);
199 };
200
201
202 #define BEGIN_MESSAGE_MAP_EX(class_name) \
203     PTH_WARNING_DISABLE \
204     inline const AFX_MSGMAP* class_name::GetMessageMap() const \
205 {return GetThisMessageMap();} \
206     inline const AFX_MSGMAP* PASCAL class_name::GetThisMessageMap() \
207 {typedef class_name ThisClass; \
208     typedef Qnd TheBaseClass; \
209     static const AFX_MSGMAP_ENTRY _messageEntries[] = {
210
211 #define IMPLEMENT_DYNAMIC_EX(class_name, super_class) \
212 AFX_COMAT_CONST CRuntimeClass class_name:class_name { \
213     #class_name, sizeof(class_name), 0xFFFF, NULL, \
214     ((CRuntimeClass*)&super_class::class_name), NULL, NULL }; \
215 inline CRuntimeClass* class_name::GetRuntimeClass() const \
216 { return RUNTIME_CLASS(class_name); }
217
218 IMPLEMENT_DYNAMIC_EX(CvImgCtrl, Qnd)
219
220 inline static IplImage imgOfROI(const IplImage* img, CvRect roi, bool invalidROI=false)
221 {
222     IplImage ret = *img;
223     int imgWidth = img->roi != 0 && invalidROI ? img->roi->width : img->width;
224     int imgHeight = img->roi != 0 && invalidROI ? img->roi->height : img->height;
225     ret.roi = 0;
226
227     if(roi.x >= imgWidth || roi.y >= imgHeight){
228         ret.width = 0;
229         ret.height = 0;
230         ret.imageData = 0;
231         return ret;
232     }
233
234     //区域裁剪
235     if(roi.x < 0){
236         roi.width += roi.x;
237         roi.x = 0;
238     }
239
240     if(roi.y < 0){
241         roi.height += roi.y;
242         roi.y = 0;
243     }
244
245     if(roi.x + roi.width > imgWidth)    roi.width = imgWidth - roi.x;
246     if(roi.y + roi.height > imgHeight)  roi.height = imgHeight - roi.y;
247
248     CV_Assert(roi.width > 0 && roi.height > 0);
249     ret.width = roi.width;
250     ret.height = roi.height;
251     ret.imageSize = ret.height * ret.widthStep;
252     ret.imageDataOrigin = 0;
253
254     if (!invalidROI && img->roi != 0)
255         ret.imageData = img->imageData + (img->roi->xOffset + roi.x) * img->nChannels * (img->depth & ~IPL_DEPTH_8U)
256     else
257         ret.imageData = img->imageData + roi.x * img->nChannels * (img->depth & ~IPL_DEPTH_SIGN) / 8 + roi.y *
258
259     return ret;
260 }
261
262 inline static void doEvent(unsigned int waitMS = -1){
263     MSG msg;
264     unsigned int time = GetTickCount();
265
266     while(GetTickCount() - time < waitMS){
267         while(GetTickCount() - time < waitMS && PeekMessage(&msg, 0, 0, 0, PM_REMOVE)){
268             if(msg.message == WM_QUIT)
269                 break;
270
271             TranslateMessage(&msg);
272             DispatchMessage(&msg);
273         }
274     }
275 }
276
277 #define min(a,b) (((a) < (b)) ? (a) : (b))
278
279 inline CvImgCtrl::CvImgCtrl(){
280     memset(&m_bmpHeader, 0, sizeof(m_bmpHeader));
281     m_bmpHeader.bmiHeader.biSize = sizeof(m_bmpHeader);
282     m_bmpHeader.bmiHeader.biXpelsPerMeter = 72;
283     m_bmpHeader.bmiHeader.biYpelsPerMeter = 72;
284     m_bmpHeader.bmiHeader.biPlanes = 1;
285     m_imgResize = 0;
286     m_img = 0;
287     m_rt = ResizeType_Tensile;
288     m_autoDelete = false;
289     m_videoState = VideoState_Stop;
290     m_defaultColor = cvScalarAll(255);
291     InitializeCriticalSection(&m_resizeCS);
292 }
293
294 inline CvImgCtrl::~CvImgCtrl(){
295     DeleteCriticalSection(&m_resizeCS);
296 }
297
298 BEGIN_MESSAGE_MAP_EX(CvImgCtrl)
299     ON_WM_SIZE()
300     ON_WM_DESTROY()
301     ON_WM_ERASEBKGD()
302     ON_WM_PAINT()
303     ON_WM_NCDESTROY()
304     ON_WM_MOUSEMOVE()
305     ON_WM_LBUTTONDOWN()
306     ON_WM_LBUTTONUP()
307     ON_WM_NCHITTEST()
308     ON_WM_RBUTTONDOWN()
309     ON_WM_RBUTTONUP()
310     ON_WM_MBUTTONDOWN()
311     ON_WM_MBUTTONUP()
312     ON_WM_KEYDOWN()
313     ON_WM_DROPFILES()
314     ON_MESSAGE(InnerMsgVideoFrame, OnVideoFrame)
315 END_MESSAGE_MAP()
316
317 inline bool CvImgCtrl::openVideo(const char* file, unsigned int waitTime){
318     return _startVideo(file, waitTime);
319 }
320
321 inline void CvImgCtrl::pauseVideo(){
322     if(m_videoState == VideoState_Playing)
323         m_videoState = VideoState_Pause;
324 }
325
326 inline bool CvImgCtrl::openVideo(int cameraIndex, unsigned int waitTime){
327     return _startVideo((const void*)cameraIndex, waitTime);
328 }
329
330 inline void CvImgCtrl::resumeVideo(){
331     if(m_videoState == VideoState_Pause){
332         m_videoState = VideoState_Playing;
333         PostMessage(InnerMsgVideoFrame, 0, 0);
334     }
335 }
336
337 inline void CvImgCtrl::closeVideo(){
338     m_videoState = VideoState_Stop;
339     if(m_videoCapture.isOpened())
340         m_videoCapture.release();
341 }
342
343 inline bool CvImgCtrl::startVideo(const void* device_or_file, unsigned int waitTime){
344     if(!::IsWindow(m_hWnd)) return false;
345
346     int dof = (int)device_or_file;
347     if(m_videoCapture.isOpened())

```

```

347     m_videoCapture.release();
348
349     bool success = false;
350     const int max_count_of_device_index = 1024;
351     const char* file = (const char*)device_or_file;
352     const int device_index = (int)device_or_file;
353     m_videoWaitTime = waitTime;
354
355     if(dof == -1 || dof >= 0 && dof <= max_count_of_device_index)
356         success = m_videoCapture.open(device_index);
357     else
358         success = m_videoCapture.open(file);
359
360     m_videoState = success ? VideoState_Playing : VideoState_Stop;
361     if(success) {
362         cv::Mat frame;
363         m_videoCapture >> frame;
364         if(!frame.empty()){
365             _onlySetImage(&IpImage(frame));
366             notifyParent(MsgVideoStart, (LPARAM)device_or_file);
367             notifyParent(MsgVideoFrame, (LPARAM)m_img);
368
369             if (m_imgResize != 0){
370                 redrawResize();
371                 notifyParent(MsgVideoPreDisplay, (LPARAM)m_imgResize);
372
373                 this->onUpdateDisplayImage();
374             }
375
376             if(m_videoState == VideoState_Playing)
377                 PostMessage(InnerMsgVideoFrame, 0, 0);
378         }else{
379             success = false;
380         }
381     }
382     m_videoState = success ? VideoState_Playing : VideoState_Stop;
383     return success;
384 }
385
386 inline HRESULT CvImgCtrl::notifyParent(UINT msg, LPARAM lp){
387     return ::SendMessage(GetParent()->m_hWnd, msg, (LPARAM)this, lp);
388 }
389
390 inline HRESULT CvImgCtrl::OnVideoFrame(WPARAM wp, LPARAM lp){
391     if(m_videoState == VideoState_Playing){
392         bool grab = m_videoCapture.grab();
393         bool retrieve = grab && m_videoCapture.retrieve(cv::Mat(m_img, false));
394
395         if(retrieve){
396             notifyParent(MsgVideoFrame, (LPARAM)m_img);
397
398             if (m_imgResize != 0){
399                 redrawResize();
400                 notifyParent(MsgVideoPreDisplay, (LPARAM)m_imgResize);
401
402                 this->onUpdateDisplayImage();
403             }
404
405             if(m_videoState == VideoState_Playing){
406                 doEvent(m_videoWaitTime);
407                 PostMessage(InnerMsgVideoFrame, wp, lp);
408             }
409         }else{
410             notifyParent(MsgVideoOver);
411             m_videoState = VideoState_Stop;
412         }
413     }
414     return 0;
415 }
416
417 inline bool CvImgCtrl::create(
418     CWnd* pParent,
419     CRect rcPos,
420     bool autoDelete /*= false*/,
421     ResizeType rt /*= _ResizeType_Non*/,
422     CvScalar defaultColor /*= cvScalarAll(255)*/){
423
424     m_defaultColor = defaultColor;
425     m_autoDelete = autoDelete;
426     m_rt = rt;
427     return CWnd::Create(_T("Static"), 0, WS_OVERLAPPED | WS_CHILD | WS_VISIBLE, rcPos, pParent, 0, 0) != 0;
428 }
429
430 inline void CvImgCtrl::redrawResize(){
431     if(m_imgResize == 0 || m_img == 0 || m_img->width * m_img->height == 0)
432         return;
433
434     EnterCriticalSection(&m_resizeCS);
435     switch(m_rt) {
436     case ResizeType_Tensile: {
437         if(m_img->nChannels != 3){
438             IpImage* re = cvCreateImage(cvGetSize(m_img), 8, 3);
439             cvCvtColor(m_img, re, CV_GRAY2BGR);
440             cvResize(re, m_imgResize);
441             cvReleaseImage(&re);
442         }
443         else
444             cvResize(m_img, m_imgResize);
445     }
446     break;
447
448     case ResizeType_Raw: {
449         void*resize();
450         CRect roi = {0, 0, min(m_img->width, m_imgResize->width), min(m_img->height, m_imgResize->height)};
451         if(m_img->nChannels != 3)
452             cvCvtColor(&imgOfROI(m_img, roi), &imgOfROI(m_imgResize, roi), CV_GRAY2BGR);
453         else
454             cvCopy(&imgOfROI(m_img, roi), &imgOfROI(m_imgResize, roi));
455     }
456     break;
457
458     case ResizeType_CenterResize: {
459         void*resize();
460         int nw = m_imgResize->width;
461         int nh = int(m_imgResize->width / (double)m_img->width * m_img->height);
462
463         if (nh > m_imgResize->height) {
464             nh = m_imgResize->height;
465             nw = int(m_imgResize->height / (double)m_img->height * m_img->width);
466         }
467
468         cvSetImageROI(m_imgResize, cvRect((m_imgResize->width - nw) / 2, (m_imgResize->height - nh) / 2, nw
469         if(m_img->nChannels != 3){
470             IpImage* re = cvCreateImage(cvGetSize(m_img), 8, 3);
471             cvCvtColor(m_img, re, CV_GRAY2BGR);
472             cvResize(re, m_imgResize);
473             cvReleaseImage(&re);
474         }
475         else
476             cvResize(m_img, m_imgResize);
477         cvResetImageROI(m_imgResize);
478     }
479     break;
480 }
481 LeaveCriticalSection(&m_resizeCS);
482 }
483
484 inline void CvImgCtrl::onUpdateImage(){
485     if (m_imgResize != 0 && !IsWindow(m_hWnd)){
486         redrawResize();
487         this->onUpdateDisplayImage();
488     }
489 }
490
491 inline void CvImgCtrl::void*resize(){
492     cvSet(m_imgResize, m_defaultColor);
493 }
494
495 inline BOOL CvImgCtrl::linkDlgItem(UINT idd, CWnd* parent){
496     BOOL r = CWnd::SubclassDlgItem(idd, parent);
497     if(r){
498         CRect rc;
499         GetWindowRect(&rc);
500
501         OnSize(0, rc.Width(), rc.Height());
502     }
503     return r;
504 }
505
506 inline void CvImgCtrl::OnSize(UINT nType, int cx, int cy){
507     if (cx < 1 || cy < 1){
508         CWnd::OnSize(nType, cx, cy);
509         return;
510     }

```

```
512 EnterCriticalSection(&m_resizeCS);
513 cvReleaseImage(&m_imgResize);
514 m_imgResize = cvCreateImage(cvSize(cx, cy), 8, 3);
515 LeaveCriticalSection(&m_resizeCS);
516
517 if(m_img != 0)
518     redrawResize();
519 else
520     voidResize();
521
522 m_bmpHeader.bmiHeader.biBitCount = m_imgResize->nChannels * m_imgResize->depth;
523 m_bmpHeader.bmiHeader.biHeight = -m_imgResize->height;
524 m_bmpHeader.bmiHeader.biWidth = m_imgResize->width;
525 m_bmpHeader.bmiHeader.biSizeImage = m_imgResize->imageSize;
526 this->onUpdateDisplayImage();
527 }
528
529 inline void CvImgCtrl::_onlySetImage(const IplImage* img){
530     if (img == 0) {
531         if (m_img != 0)
532             cvReleaseImage(&m_img);
533
534         if (m_imgResize != 0)
535             voidResize();
536
537         return;
538     }
539
540     if (m_img == 0){
541         m_img = (IplImage*)cvClone(img);
542     }
543     else{
544         if (
545             m_img->width != img->width ||
546             m_img->height != img->height ||
547             m_img->nChannels != img->nChannels ||
548             m_img->depth != img->depth){
549
550             cvReleaseImage(&m_img);
551             m_img = (IplImage*)cvClone(img);
552         }
553         else
554             memcpy(m_img->imageData, img->imageData, min(img->imageSize, m_img->imageSize));
555         //cvCopy(img, m_img);
556     }
557 }
558
559 inline void CvImgCtrl::setImage(const IplImage* img){
560     _onlySetImage(img);
561     onPateImage();
562 }
563
564 inline void CvImgCtrl::OnDestroy(){
565     CWnd::OnDestroy();
566     if (m_img != 0)
567         cvReleaseImage(&m_img);
568
569     if (m_imgResize != 0)
570         cvReleaseImage(&m_imgResize);
571 }
572
573 inline BOOL CvImgCtrl::OnEraseBknd(CDC* pDC){
574     return TRUE;
575 }
576
577 inline void CvImgCtrl::onUpdateDisplayImage(){
578     Invalidate();
579     UpdateWindow();
580 }
581
582 inline void CvImgCtrl::OnPaint(){
583     if(m_imgResize != 0){
584         PAINTSTRUCT ps;
585         HDC hDC = ::BeginPaint(m_hWnd, &ps);
586         CRect rcPaint = ps.rcPaint;
587         StretchDibits(hDC, 0, 0, m_imgResize->width, m_imgResize->height, 0, 0, m_imgResize->width, m_imgResize->
588             ::EndPoint(m_hWnd, &ps);
589     }
590     else
591         CWnd::OnPaint();
592 }
593
594 inline void CvImgCtrl::OnNcDestroy(){
595     CWnd::OnNcDestroy();
596
597     if (m_autoDelete)
598         delete this;
599 }
600
601 inline void CvImgCtrl::OnMouseMove(UINT nFlags, CPoint point){
602     notifyParent(MsgMove, (LPARAM)&point);
603 }
604
605 inline void CvImgCtrl::OnLButtonDown(UINT nFlags, CPoint point){
606     m_lastdownTime = ::GetTickCount();
607     notifyParent(MsgLButtonDown, (LPARAM)&point);
608 }
609
610 inline void CvImgCtrl::OnLButtonUp(UINT nFlags, CPoint point){
611     notifyParent(MsgLBUp, (LPARAM)&point);
612
613     if(GetTickCount() - m_lastdownTime <= 300)
614         notifyParent(MsgClick, (LPARAM)&point);
615 }
616
617 inline void CvImgCtrl::OnRButtonDown(UINT nFlags, CPoint point){
618     notifyParent(MsgRButtonDown, (LPARAM)&point);
619 }
620
621 inline LRESULT CvImgCtrl::OnNcHitTest(CPoint point){
622     return HTCLIENT;
623 }
624
625 inline void CvImgCtrl::OnMButtonDown(UINT nFlags, CPoint point){
626     notifyParent(MsgMButtonDown, (LPARAM)&point);
627 }
628
629 inline void CvImgCtrl::OnMButtonUp(UINT nFlags, CPoint point){
630     notifyParent(MsgMButtonUp, (LPARAM)&point);
631 }
632
633 inline void CvImgCtrl::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags){
634     notifyParent(MsgKeyDown, nChar);
635 }
636
637 inline void CvImgCtrl::OnDropFiles(HDROP hDropInfo){
638     CWnd::OnDropFiles(hDropInfo);
639     notifyParent(MsgDragFile, (LPARAM)hDropInfo);
640 }
```

hope_v

码龄8年

暂无认证

7

26万+

170万+

2万+

周排名

总排名

访问

等级

338

22

11

9

23

积分

粉丝

获赞

评论

收藏

私信

关注

搜博主文章

热门文章

win32打印控制、API打印操作

8778

OpenCV系列1-简单高效：MFC显示图片、播放视频

8781

MFC用OpenCV打开摄像头显示视频或图像(Picture Control控件)
MFC创建对话框，用OpenCV读图，读取图像或者打开摄像头或者打开视频。保存视频帧，为图像

MFC 获取摄像头 实时 显示并图像处理 (光斑自动对焦)
第一次写CSDN论文，给自己的毕业做个纪念。对于这个常年用C语言的渣渣，突然接触需要C++的MFC和Openov，就发成毕业论文难了，好在有CSDN...

OpenCV学习记录(一)——在OpenCV在MFC控件上显示图片和视频
IcvLoadImage函数，不管图片路径换成绝对路径还是绝对路径，通过image_watch查看，IplImage*的变量还是空的。后来想着，要不要用mimage*读图放到mat...

OpenCV学习笔记 利用MFC的Picture控件显示图像+播放视频...
在《OpenCV教程-基础篇》的2.8节中，所创建的MFC图像显示是直接放在对话框面板的左上角的，感觉不大美观。《MFC中快速应用OpenCV》则是介绍用S...

MFC显示Mat图片 入门指南
在MFC框架上显示图片可以通过Cvimage类里的DrawPicToHDC(IplImage*img,UINT ID)方法方便的显示出来，当然这里使用的是最...

MFC读取图像并显示
old CoderDg - OnBnClickedButton1() { // TODO: 在此添加控件通知处理程序代码 //添加代码 // TODO: 在此添加控件通知处理程序代码 //打...

opencv2在MFC对话框控件中显示图片播放视频_沐阳2100的...
利用opencv在MFC对话框相关控件上显示图片或者播放视频的方法很多,如使用opencv的Cvimage类,将Mat数据转换到CImage,使用MFC绘图函数等,但个...

mfc集成opencv实现监控、拍照、录像、录像播放(保姆级)...
将opencv\build\x64\vc15\bin目录下的opencv_videolo_ffmpeg454_64.dll,opencv_world454d.dll文件都拷贝到www所在目录。拷贝好的项目目录如下:3.4...

MFC入门之七——图像显示
用static text显示图像，并设置鼠标点击下一张 效果图：点击上一张按钮，显示上一张 点击下一张按钮，显示下一张 点击自动按钮，自动显示文件内图像...

关于opencv在MFC上显示图像的具体操作步骤及结果.rar
此为Word文件，详细讲述了如何从0开始创建一个简单的opencv和MFC的图像处理框架，具有打开文件去读取图像，进行简单图像处理，保存处理后...

OpenCV+MFC显示图像/视频_caiqi1123的博客
但它的显示是使用的OpenCV库里面的简单的GUI用来测试验证可以,如果要做成可视化工程,还是得借助专门的GUI开发工具了。本文简述使用MFC显示图...

OpenCV环境配置Visual Studio 2010、2012、2013、2015 3827

Cocoa2dx 3.2移植到Android 完全可行的方法 1182

使用ViewPager，设置为wrap_content后没有效果，以及多个view高度不同时ViewPager高度自适应 1107

最新评论

今天申请的帐号，纪念咯
hope_v 回复 Amelia0911: ...当成真的他才是真的
今天申请的帐号，纪念咯
Amelia0911: 说的跟真的似的

OpenCV系列1-简单高效：MFC显示图片...
8057 回复 chialander: 把头文件CvimgCtrl.h 添加到demoDlg.cpp中，不要在.h中，
OpenCV系列1-简单高效：MFC显示图片...
chialander 回复 再见雄虫: 请问解决了么，我也遇到同样问题了(๑_๑)_b

OpenCV系列1-简单高效：MFC显示图片...
再见雄虫: IMPLEMENT_DYNAMIC_EXCOWingCtrl, CWnd这一直有问题，为什么？

您愿意向朋友推荐“博客详情”吗？

强烈推荐 不推荐 一般般 推荐 强烈推荐

最新文章

iOS自学之路0 - 开篇

Cocoa2dx 3.2移植到Android 完全可行的方法

OpenCV环境配置Visual Studio 2010、2012、2013、2015

2016年 1篇 2015年 4篇
2014年 3篇

OpenCV窗口绑定到MFC中用以显示视频或图片数据_Captain...

当使用MFC在OpenCV显示图片数据时,很多采用将Mat类型的图片数据转成CvImage类型,然后在MFC的窗口中进行显示,但是将面临需要额外的头文件Cv...

MFC+OPENCV显示视频每一帧

MFC 图片控件只能显示一幅图片

OpenCV在MFC中显示图片_youngyang525的博客

在MFC控件中显示opencv库中的图像或者视频用的CvImage...

mfc+最大最小化窗口+控件变化+图片（视频帧）显示在控件上

数字图像处理实验1——用MFC编程显示图像

OpenCV-基于MFC的视频播放器器和图片读取器_张倬胜的博客

OpenCV应用MFC上画图片显示的控件代码案例，应用，就一个h文件就可以干起来，该代码能很轻易的实现图片和视频的显示，高效易用，简单，参...

【待测】Opencv 实现简单播放器_最新发布

MFC OpenCV VS2019显示视频

简单高效：OpenCV MFC下显示图片视频demo控件

OpenCV应用MFC上画图片显示的控件代码案例，应用，就一个h文件就可以干起来，该代码能很轻易的实现图片和视频的显示，高效易用，简单，参...

【待测】Opencv 实现简单播放器_最新发布

MFC OpenCV VS2019显示视频

opencv +MFC实现视频播放、暂停、视频标注、跟踪

最近在搞视频标注、跟踪这一块，参考了好多资料，先把代码和效果图贴出来。 #include "CvImage.h" #include "opencv2/opencv.hpp" #include <win...

“相关推荐”对你有帮助么？

非常有帮助 没帮助 一般 有帮助 非常有帮助

©2022 CSDN 皮肤主题：大白 设计师：CSDN官方博客 返回首页

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

hope_v 关注

10 4 11 专栏目录

