

# RGB 컬러 공간 분리 - 메뉴 기반 구현

<http://deios.kr>

이번에는 본격적으로 OpenCV를 사용하여 영상처리를 해 보겠습니다. 그 첫 단계로 메뉴기반 정지영상 처리를 해 볼 것이며, cvSplit함수를 사용하여 RGB 컬러공간을 분리하는 예제를 구현해 보겠습니다.

먼저 OpenCV 2.1 C API Reference에서 cvSplit함수를 찾아보면 다음과 같습니다.

## cvSplit

Divides multi-channel array into several single-channel arrays or extracts a single channel from the array.

```
void cvSplit(const CvArr* src, CvArr* dst0, CvArr* dst1, CvArr* dst2, CvArr* dst3);
```

**src** Source array

**dst0** Destination channel 0

**dst1** Destination channel 1

**dst2** Destination channel 2

**dst3** Destination channel 3

The function divides a multi-channel array into separate single-channel arrays. Two modes are available for the operation. If the source array has N channels then if the first N destination channels are not NULL, they all are extracted from the source array; if only a single destination channel of the first N is not NULL, this particular channel is extracted; otherwise an error is raised. The rest of the destination channels (beyond the first N) must always be NULL. For IplImage [cvCopy](#) with COI set can be also used to extract a single channel from the image.

cvSplit함수는 다중 채널(Multi-channel)배열을 여러 개의 단일 채널(Single-channel) 배열로 분할하거나, 배열에서 단일 채널을 추출 할 수 있음을 알 수 있습니다.

또한, 입력이 IplImage인 경우에는 cvCopy함수를 사용해서도 같은 결과를 얻을 수 있음을 알 수 있습니다.

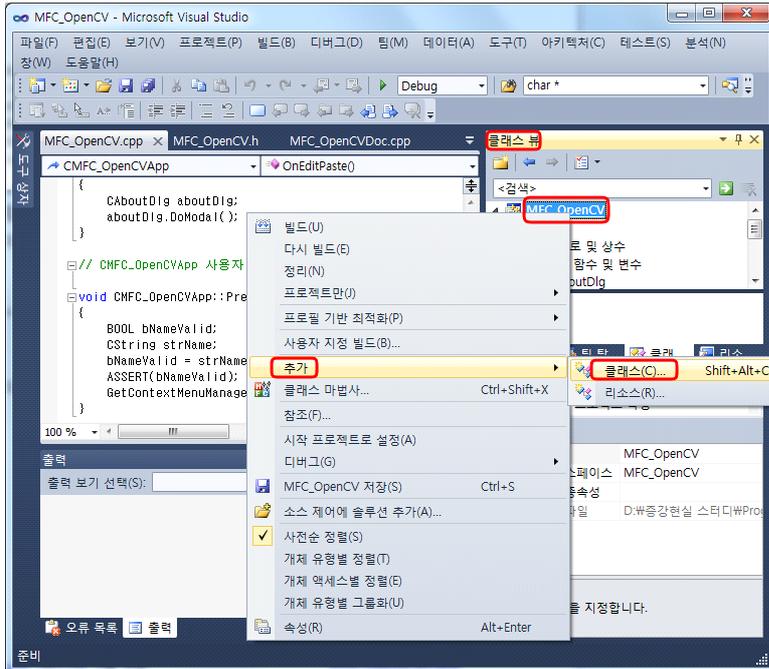
이미 “영상처리 프로그래밍 by Visual C++”에서 해 본 것처럼 별도의 Class를 만들고 이를 호출하는 방식으로 예제를 구현해 보겠습니다.

먼저 해당 기능을 구현할 Class를 만듭니다. 저는 CDeiosColor라고 하겠습니다.

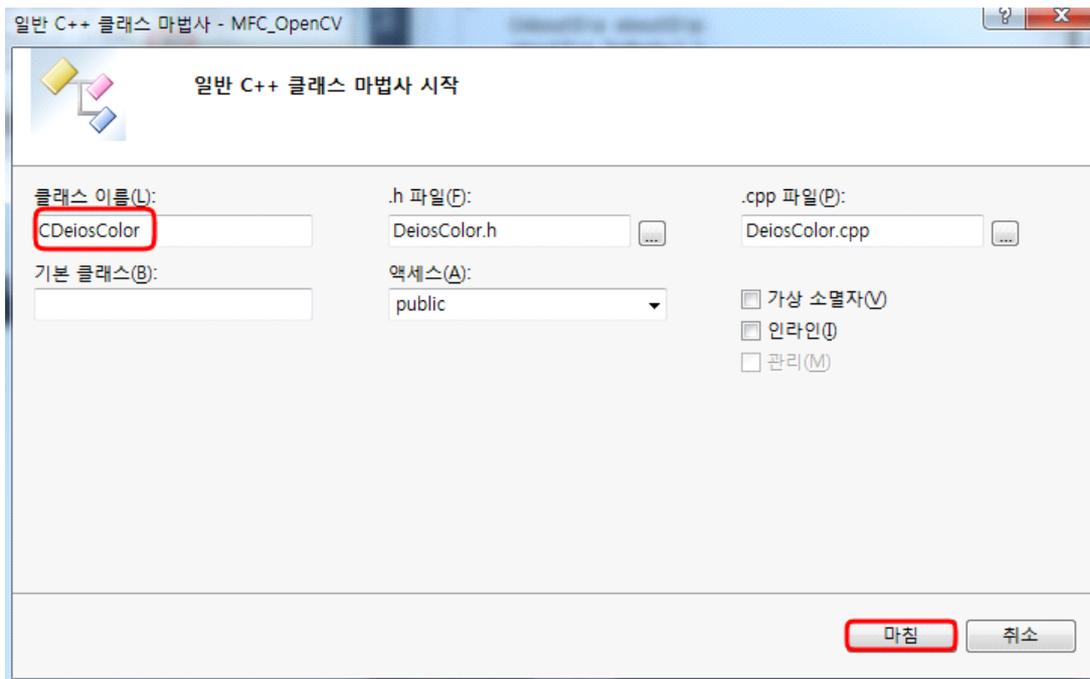
MFC를 공부할 때 헝가리언 표기법(Hungarian notation)을 알아둔다면 많은 도움이 됩니다. MS가 .Net에서 헝가리언 표기법을 포기했다고는 하지만 많은 이름들이 헝가리언 표기법을 근사적으로 따릅니다. 헝가리언 표기법을 반드시 따를 필요는 없지만 대략적으로 봐 둔다면 MFC프로그래밍 뿐만 아니라 각종 라이브러리 사용 시에 많은 도움이 됩니다. 자신만의 코딩 스타일을 만들어서 일관적으로 사용한다면 과거에 만들었던 코드를 분석하거나 디버깅 할 때에 많은 도움을 받게 될 것입니다.

참고로 Class 이름(DeiosColor) 앞의 C는 Class임을 알려주는 MS의 코딩 스타일입니다.

클래스 뷰에서 마우스 오른쪽 버튼을 클릭하고, [추가] → [클래스]를 선택하여 클래스 추가 마법사를 실행합니다.



[C++ 클래스]를 선택하여 일반 Class를 만든 후 클래스 이름을 지정해 주고 [마침]을 클릭합니다.



다음으로 멤버 함수를 추가하겠습니다. 멤버 함수는 DSplitRGB이며 함수 원형은 다음과 같습니다.

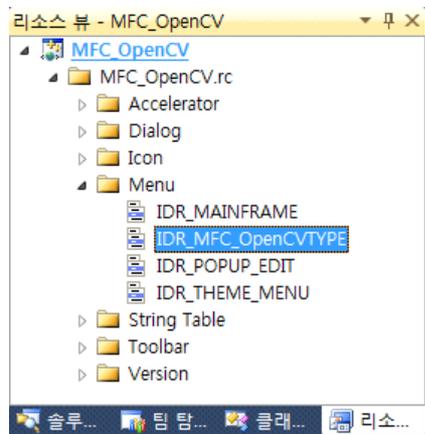
```
void DSplitRGB(IplImage *color_image, IplImage **red, IplImage **green, IplImage **blue)
```

헤더 파일(DeiosColor.h)에 cv.h를 include 한 후, 소스 파일(DeiosColor.cpp)에 다음과 같이 코딩합니다.

```
void CDeiosColor::DSplitRGB(IplImage * color_image, IplImage ** red, IplImage ** green,
IplImage ** blue)
{
    *red = cvCreateImage(cvGetSize(color_image), IPL_DEPTH_8U,1);
    *green = cvCreateImage(cvGetSize(color_image), IPL_DEPTH_8U,1);
    *blue = cvCreateImage(cvGetSize(color_image), IPL_DEPTH_8U,1);

    cvSplit(color_image, *red, *green, *blue, NULL);
}
```

이번에는 새로운 메뉴를 등록해 보겠습니다. MFC 리소스 뷰의 메뉴 항목을 보면 다음과 같은 4개의 리소스가 보입니다.



이 중 IDR\_MAINFRAME은 문서가 하나도 생성되지 않은 시점의 메뉴를, IDR\_MFC\_OpenCVTYPE은 한 개 이상의 문서가 생성된 시점의 메뉴를, IDR\_POPUP\_EDIT은 팝업 메뉴를 의미합니다. 우리는 한 개의 문서(사진)을 필요로 하기 때문에 IDR\_MFC\_OpenCVTYPE 리소스를 편집해야 합니다.

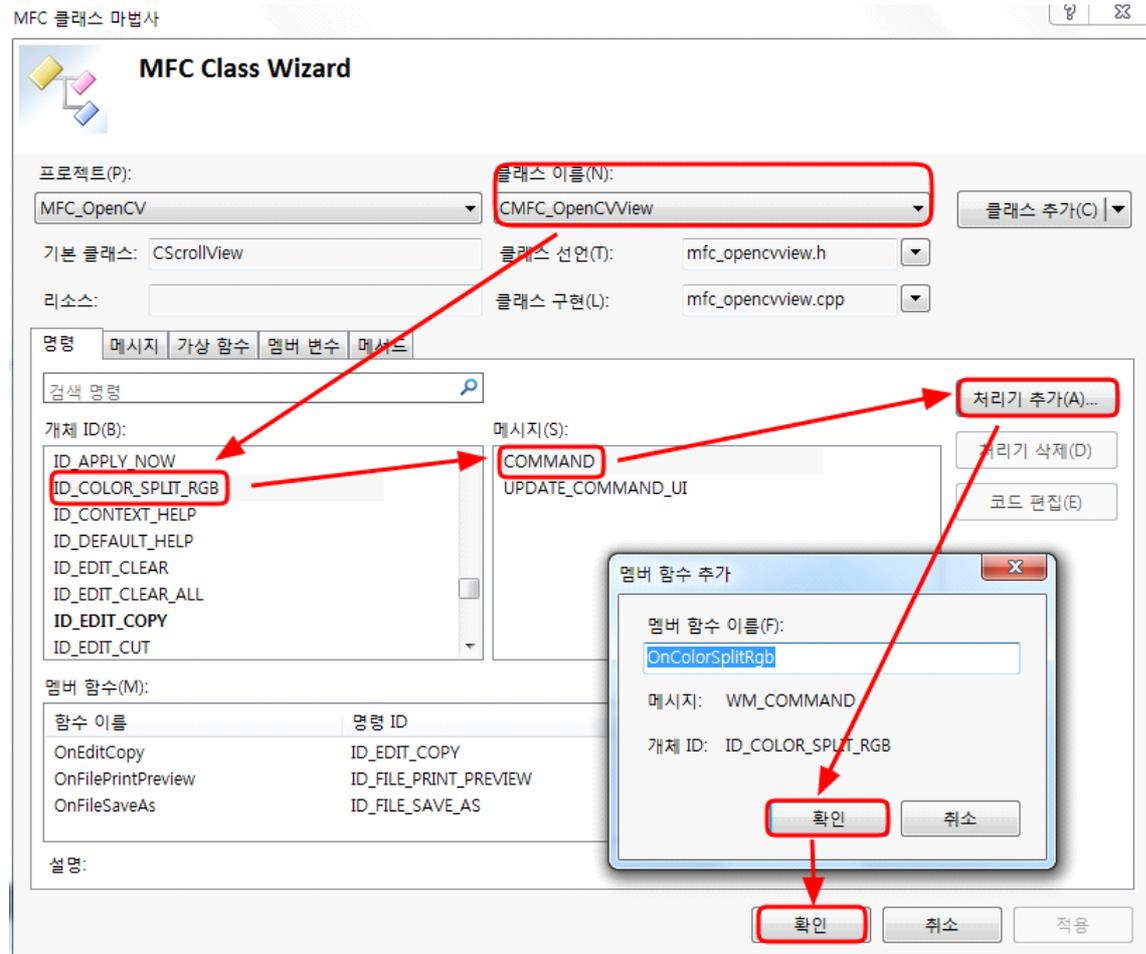
다음과 같은 모양으로 메뉴를 편집하고 다음과 같은 속성을 줍니다.



(Name)	메뉴 편집기
Break	None
Caption	Split RGB
Checked	True
Enabled	True
Grayed	False
Help	False
ID	ID_COLOR_SPLIT_RGB
Popup	False
Prompt	RGB 컬러공간을 분리합니다.
Right Justify	False
Right Order	False
Separator	False

이번에는 Split RGB메뉴를 클릭했을 때 발생할 이벤트를 처리하는 이벤트 처리기를 등록해 보겠습니다.

[Ctrl + Shift + X]키를 누르면 [Class Wizard]가 열립니다.



"DeiosColor.h"파일을 include한 후 다음과 같이 코딩합니다.

```
void CMFC_OpenCVView::OnColorSplitRgb()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    CMFC_OpenCVDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);

    IplImage *color_image = pDoc->m_CvvImage.GetImage();

    if( color_image->nChannels != 3 ){
        AfxMessageBox( TEXT("컬러 영상이 아닙니다."), MB_ICONSTOP );
        return;
    }

    IplImage *red, *green, *blue;
```

```

CDeiosColor cColor;
cColor.DSplitRGB( color_image, &red, &green, &blue );

pDoc->CopyClipBoard( red );
((CMFC_OpenCVApp *)AfxGetApp()->OnEditPaste();
CMFC_OpenCVDoc *pDoc1 = (CMFC_OpenCVDoc
*)((CMDIFrameWnd*)AfxGetMainWnd()->GetActiveFrame()->GetActiveDocument());
pDoc1->SetTitle(TEXT("[SplitRGB0] 변환 결과 - 빨강 채널"));

pDoc->CopyClipBoard( green );
((CMFC_OpenCVApp *)AfxGetApp()->OnEditPaste();
CMFC_OpenCVDoc *pDoc2 = (CMFC_OpenCVDoc
*)((CMDIFrameWnd*)AfxGetMainWnd()->GetActiveFrame()->GetActiveDocument());
pDoc2->SetTitle(TEXT("[SplitRGB0] 변환 결과 - 녹색 채널"));

pDoc->CopyClipBoard( blue );
((CMFC_OpenCVApp *)AfxGetApp()->OnEditPaste();
CMFC_OpenCVDoc *pDoc3 = (CMFC_OpenCVDoc
*)((CMDIFrameWnd*)AfxGetMainWnd()->GetActiveFrame()->GetActiveDocument());
pDoc3->SetTitle(TEXT("[SplitRGB0] 변환 결과 - 파랑 채널"));

cvReleaseImage( &red );
cvReleaseImage( &green );
cvReleaseImage( &blue );

Invalidate( FALSE );
}

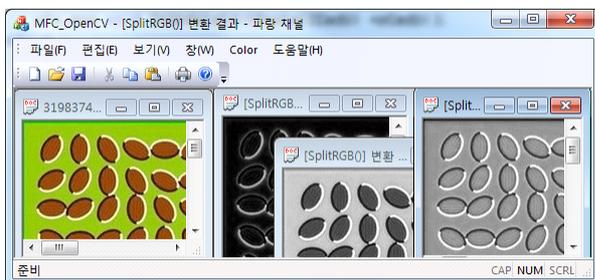
```

마지막으로 MFC\_OpenCV.h에서 OnEditPaste를 public으로 바꿔줍니다.

```

afx_msg void OnAppAbout( );
DECLARE_MESSAGE_MAP( )
afx_msg void OnFileOpen( );
afx_msg void OnFileCloseAll( );
IplImage * CreateFromHandle( HANDLE hDIB);
afx_msg void OnUpdateEditPaste( CCmdUI *pCmdUI);
public :
afx_msg void OnEditPaste( );

```



## Reference

정성환외 1명/컴퓨터 비전 실무 프로그래밍/홍릉과학출판사/2007/274p~287p

최호성/Visual C++ 2008 MFC 윈도우 프로그래밍/프리렉/2009/669p~674p

OpenCV 2.1.0 The Reference Manual/127p