

VJEŽBA 1: UVOD U OPENCV I VISUAL C++

I. Cili vježbe: *Upoznati se s bibliotekom OpenCV. Naučiti kako koristiti Microsoft Visual C++ i Intel OpenCV za izgradnju aplikacije. Napisati jednostavni program za određivanje rubova na slici pomoću Canny-evog detektora rubova.*

II. Opis vježbe:

OpenCV (Intel® **Open** Source Computer Vision Library) je “open source” biblioteka ponuđena od strane Intela. OpenCV sadrži kolekciju C funkcija i C++ klasa koje se često primjenjuju u obradi slika i algoritmima za računalni vid.

C++ trenutno slovi kao najrašireniji programski jezik. Svoju je titulu stekao ne samo svojim mogućnostima, već i sirovom snagom te fleksibilnošću koje ga krase. Najrašireniji alat za razvoj aplikacija korištenjem C++-a je upravo Visual Studio. Visual Studio razvojno sučelje sadrži u sebi niz programskih jezika (Visual Basic, C++, C#, J#...).

Ova vježba predstavlja uvod u korištenje MS Visual C++ i Intel OpenCV u izgradnji aplikacije. Budući da je ovo početnički vodič, detaljno su opisani svi potrebni koraci za dobivanje prikazanih rezultata.

III. Priprema za vježbu:

Ponoviti osnove iz C++, kao što su programska struktura, tipovi podataka, osnovne strukture podataka (polje, strukture, itd.), pokazivači (*engl.* pointers) itd. Stranica <http://www.cplusplus.com/doc/tutorial/> se može koristiti kao pomoć. Također je potrebno upoznati se s integriranim razvojnim okruženjem MS Visual C++ 2008, proći kroz izbornike i osnove korištenja.

Pogledati dokumentaciju za OpenCV2.0 koja je dostupna na web-stranici <http://opencv.willowgarage.com/documentation/index.html> ili skinuti pdf datoteku [opencv.pdf](#). Posebno pročitati dokumentaciju vezanu za CXCORE, CvReference te HighGUI. Također downloadirati zip datoteku [OpenCV20.zip](#) u kojem se nalazi lib, dll i h datoteke koje će se koristiti na vježbama.

Neke od metoda i struktura OpenCV-a korisne za rad na vježbi:

Metode:

```
cvLoadImage; cvNamedWindow; cvShowImage; cvCvtColor; cvCanny; cvCreateImage;  
cvReleaseImage; cvDestroyAllWindows; cvSetImageROI; cvCopy; cvResetImageROI;  
cvSetCaptureProperty; cvCreateCameraCapture.
```

Strukture:

```
IplImage; CvRect.
```

Linkovi:

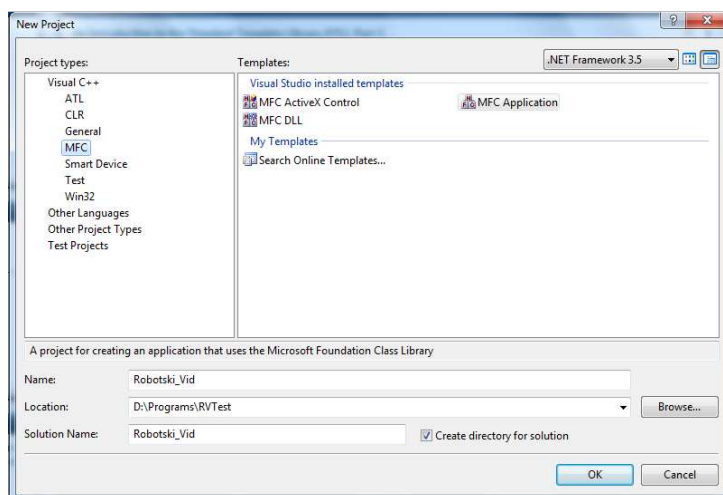
opencv.pdf - <http://moodle.etfos.hr/mod/resource/view.php?id=10677>
OpenCV20.zip - <http://moodle.etfos.hr/mod/resource/view.php?id=10678>
Listing 1.4. - <http://moodle.etfos.hr/mod/resource/view.php?id=10683>

IV. Rad na vježbi:

1. Kreiranje aplikacije zasnovane na dijalogu (Dialog Based Application)

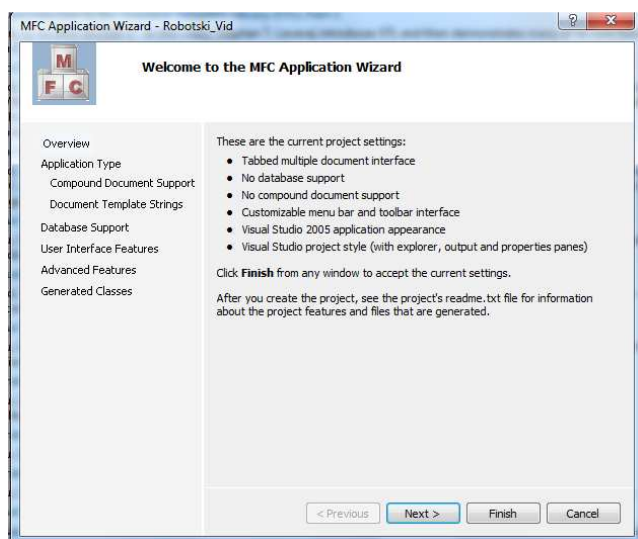
Slijedi prikaz izrade jednostavne aplikacije zasnovane na dijalogu. Ove se vrste aplikacije mogu lako kreirati pomoću čarobnjaka MFC aplikacija.

Pokrenuti Microsoft Visual C++ 2008 te iz izbornika kliknuti na **File|New|Projects** opciju, odabrati tip projekta **MFC** te odabrati template **MFC Application** i upisati ime projekta (ovdje Robotski_Vid) te folder gdje će se nalaziti taj projekt (Slika 1.1.)

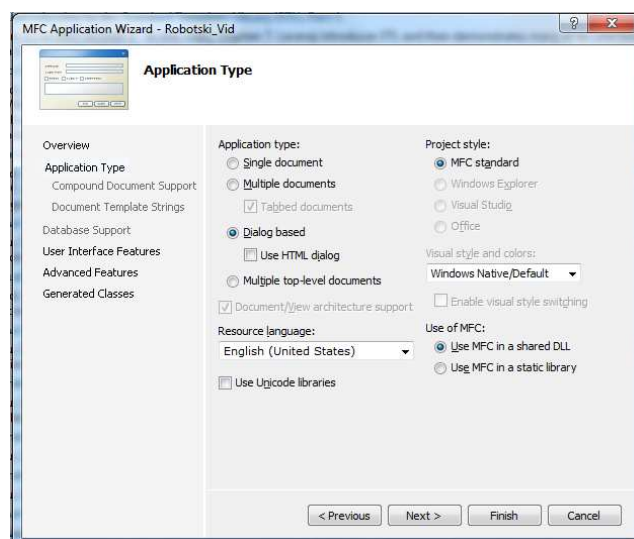


Sl. 1.1. Izbor tipa projekta (MFC aplikacija) i definiranje imena projekta

Nakon klika na **OK**, pokrenut će se MFC čarobnjak (MFC Application Wizard) koji nudi različite mogućnosti izbora vrste aplikacije. U ovom trenutku treba jednostavno kliknuti na **Next** (Slika 1.2.), odabrati **Dialog based** opciju te svakako isključiti opciju **Use Unicode libraries** (Slika 1.3.) te kliknuti na **Finish**. Ostale standardne postavke mogu ostati kao što jesu (ostale opcije se mogu kasnije istražiti)

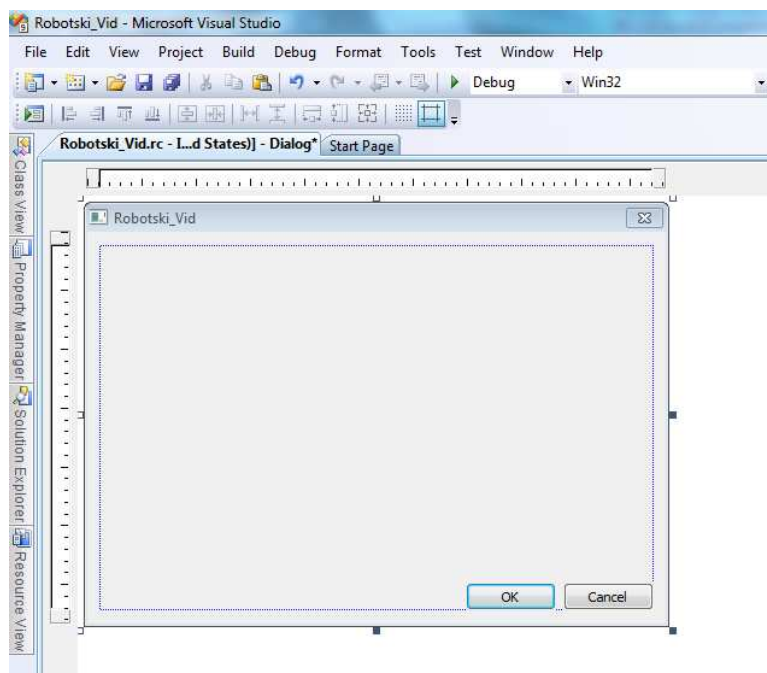


Sl. 1.2. MFC čarobnjak (MFC Application Wizard)



Sl.1.3. Definiranje tipa aplikacije

Kreirat će se jednostavna dijalog aplikacija s dva gumba “**Ok**” i “**Cancel**” (Slika 1.4.). Klasa s imenom koja završava s “**DLG**” (ovdje CRobotski_VidDlg) će sadržavati funkcije koje upravljaju dijalogom.



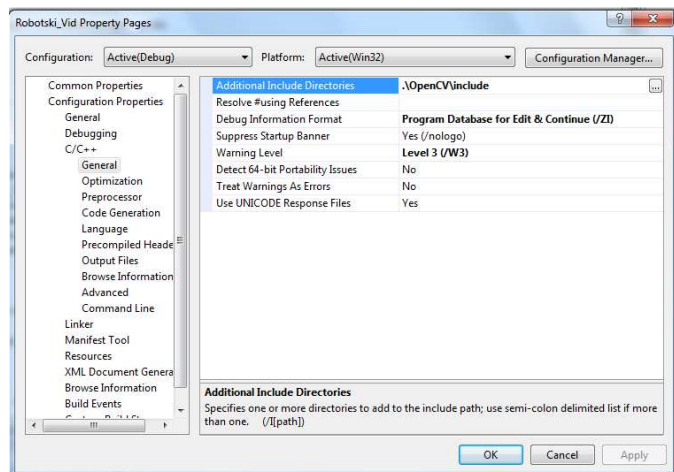
Sl. 1.4. Jednostavna dijalog aplikacija

Budući da će se koristiti biblioteke iz OpenCV-a u svim vježbama, potrebno je podesiti odgovarajuće postavke u MS Visual C++ projektu kojima se biblioteke povezuju s aplikacijom. Najprije raspakirati zip datoteku OpenCV20.zip (koju ste prethodno downloadirali) u direktorij gdje se nalazi novokreirani projekt. Kliknuti na **Project|Properties**, odabrati opciju **Configuration Properties|C/C++|General** te pod kategorijom **Additional Include Directories** dodati (Slika 1.5.):

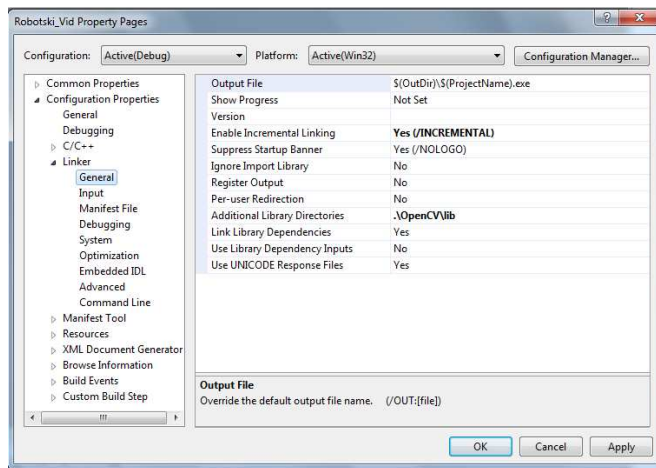
- `.\OpenCV20\include` ;

Odabrati opciju **Configuration Properties|Linker|General** te pod kategorijom **Additional Library Directories** dodati (Slika 1.6.):

- `.\OpenCV20\lib`.



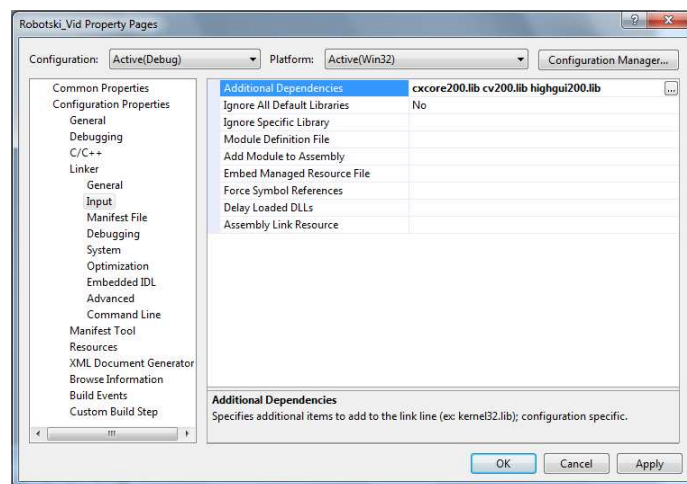
Sl. 1.5. Definiranje "Include" datoteke



Sl. 1.6. Definiranje "Library" datoteke

S tim postavkama se trebaju sada samo navesti imena biblioteka modula koja su potrebna. Odabrati opciju **Configuration Properties|Linker|Input** te pod kategorijom **Additional Dependencies**, unijeti imena tri glavne komponente OpenCV biblioteke (Slika 1.7.):

- **cxcore200.lib** (sadrži uglavnom osnovne strukture podataka);
- **cv200.lib** (sadrži funkcije za obradu slika i računalni vid);
- **highgui200.lib** (osnovna biblioteka za prikaz i spremanje slike).



Sl. 1.7. Definiranje postavki projekta

Ukoliko se želi napisati program koji će koristiti OpenCV klase i funkcije, najprije se moraju uključiti datoteke zaglavlja **highgui.h** i **cv.h** u datoteci **xxxDlg.h**. Primijetiti da je u datoteci **highgui.h** već uključen **cxcore.h**.

2. Učitavanje i prikaz slike

Prva zadaća je otvaranje i prikazivanje slike. Da bi se to učinilo, prvo treba dodati gumb koji će omogućiti odabir datoteke koja sadrži slike.

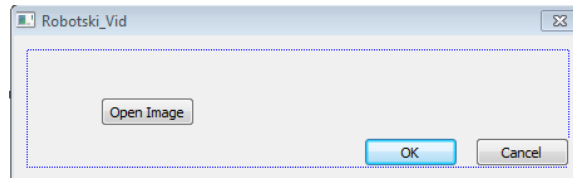
a) Ići pod **Tools** i povući gumb na dijalog (*engl. form*).

b) Promijeniti tekst na gumbu tako da piše **“Open Image”**.

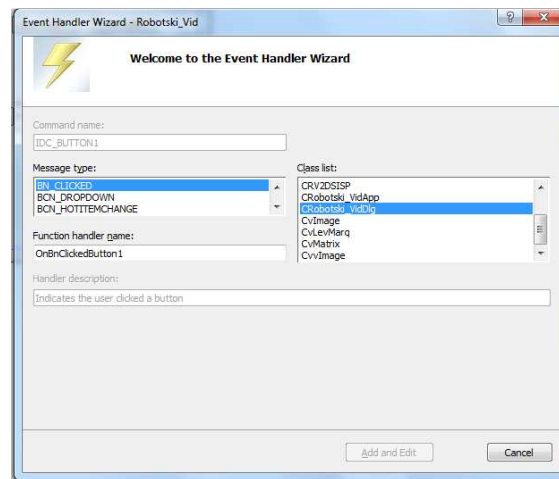
(Desni klik na gumb, pa odabrati **Properties** i promijeniti **Caption** (Slika 1.8.)).

c) Definirati upravitelja događaja (*engl. event handler*) kada se klikne na gumb.

(Desni klik na gumb pa odabrati **Add Event Handler**. Zadati ime pripadne funkcije ili upravitelja događaja, npr. OnImageOpen (Slika 1.9)).



Sl. 1.8. Dijalog aplikacija



Sl. 1.9. Čarobnjak upravitelja događaja (“Event Handler wizard”)

Klikom na gumb **“Add and Edit”** odmah će se uskočiti u Robotski_VidDlg.cpp unutar funkcije OnImageOpen ().

d) Treba upisati kôd (Listing 1.1.) u member funkciju OnImageOpen ().

```
void CRobotski_VidDlg::OnImageOpen()
{
    CFileDialog dlg(TRUE, _T("*.bmp"), "", OFN_FILEMUSTEXIST|OFN_PATHMUSTEXIST|OFN_HIDEREADONLY,
        "Image files (*.bmp; *.jpg; *.pgm; *.png) |*.bmp;*.jpg;*.pgm;*.png |All Files (*.*)|*.*||", NULL);

    dlg.m_ofn.lpstrTitle = _T("Select Image");

    if (dlg.DoModal() == IDOK)
    {
        CString path = dlg.GetPathName(); //Sadrži odabranu datoteke
        IplImage *image; //Pokazivač na sliku
        image = cvLoadImage(path); //Učitavanje slike
        cvNamedWindow("Original image"); //Stvaranje prozora u kojem će slika biti prikazana
        cvShowImage("Original image", image); //Prikaz slike
        cvReleaseImage(&image);
    }
}
```

Listing 1.1. Definiranje upravitelja događaja za gumb **“Open Image”**

Imena OpenCV funkcije počinju sa “cv”. IplImage je struktura kojom se u OpenCV prikazuju slike.

- e) Nadalje, pri izlasku iz programa, mora se pozvati funkcija `cvDestroyAllWindows()` koja će zatvoriti sve prozore napravljene pomoću **highgui**. Taj poziv treba biti vezan uz događaje kada se kliknu gumbi **OK** i **Cancel** (Listing 1.2.).

```
void CRobotski_VidDlg::OnBtnClickOk()  
{  
    cvDestroyAllWindows();  
    OnOK ();  
}  
  
void CRV1Dlg::OnBtnClickCancel()  
{  
    cvDestroyAllWindows();  
    OnCancel ();  
}
```

Listing 1.2. Definiranje upravitelja događaja za gumbove **OK** i **Cancel**

- f) Pokrenuti aplikaciju tipkom **F5** u MS Visual C++ 2008 i provjeriti kako program radi.

3. Obrada slike

```
void CRobotski_VidDlg::OnImageOpen()
{
    CFileDialog dlg(TRUE, _T("*.bmp"), "", OFN_FILEMUSTEXIST|OFN_PATHMUSTEXIST|OFN_HIDEREADONLY,
        Image files (*.bmp; *.jpg; *.pgm; *.png) | *.bmp;*.jpg;*.pgm;*.png | All Files (*.*)|*.*||", NULL);

    dlg.m_ofn.lpstrTitle = _T("Select Image");

    if (dlg.DoModal() == IDOK)
    {
        CString path = dlg.GetPathName();           //Sadrži odabranu datoteke
        IplImage *pImageIn;                         //Pokazivač na sliku
        pImageIn = cvLoadImage(path);               //Učitavanje slike
        cvNamedWindow("Original image");           //Stvaranje prozora u kojem će slika biti prikazana
        cvShowImage("Original image", pImageIn);   //Prikaz slike
        IplImage *pImageOut = cvCreateImage(cvSize(pImageIn->width,pImageIn->height),IPL_DEPTH_8U,1);
        cvCvtColor(pImageIn, pImageOut, CV_BGR2GRAY );
        cvCanny(pImageOut, pImageOut, 50, 200, 3 );
        cvNamedWindow ("Processed image");
        cvShowImage("Processed image", pImageOut);
        cvReleaseImage(&pImageIn);
        cvReleaseImage(&pImageOut);
    }
}
```

Listing 1.3. Primjer kôda za obradu slike

- a) Analizirati Listing 1.3. i komentirati označene linije.
- b) Dodati gumb **“Process Image”** koji vrši procesiranje odabrane slike. Na taj način odvajati događaj izbora i prikaza slike od događaja procesiranja slike.

4. Korištenje događaja vezanih uz miša

Listing 1.4. daje primjer jednostavnog programa koji crta pomoću klikova mišem. Kada se mišem klikne na prozor, stvara se pravokutnik koji se proteže sve dok se ne otpusti miš. To zahtjeva definiranje povratnog poziva (*engl.* callback) za miš. Glavna funkcija koristi “while” petlju koja neprestano precrtava pravokutnik kako se kreira. Novost je definiranje povratnog poziva za miš. Funkcija povratnog poziva pridružuje se mišu pomoću funkcije `cvSetMouseCallback()`. Funkcija povratnog poziva miša je proizvoljno nazvana funkcija oblika `my_mouse_callback(int event, int x, int y, int flags, void * param)`, gdje su `x` i `y` koordinate miša, a “event” je šifra koja predstavlja akciju miša koja se dogodila. Povratni poziv pomoću “switch” naredbe određuje koje se akcije moraju izvršiti ovisno o događaju, npr. kada se miš pomakne, klikne ili otpusti.

Treba analizirati kôd i primijeniti ga u aplikaciji tako da korisnik pomoću miša može označiti dio slike (slika dobivena kada se klikne na “**Open Image**”) i tako označeni dio odmah prikazati u novootvorenom prozoru (za pomoć pogledati Listing 1.5). (*Napomena:* Akcija je moguća samo ako se klikne na gumb “**Crop Image**”, a pomoću **Esc** se izađe iz tog moda.)

```
void crop_image(IplImage* img, CvRect rect)
{
    cvSetImageROI(img, rect);
    IplImage *img2 = cvCreateImage(cvGetSize(img),
                                   img->depth,
                                   img->nChannels);
    cvCopy(img, img2, NULL);

    cvResetImageROI(img);

    cvNamedWindow ("Cropped Image", CV_WINDOW_AUTOSIZE );
    cvShowImage("Cropped Image", img2);
    cvReleaseImage(&img2);
}
```

Listing 1.5. Primjer programa za izdvanja dijela slike (*eng.* crop)

5. Prikaz videa iz kamere spojene na računalo

```
CvCapture *capture;= cvCreateCameraCapture(0);

//capture = cvCaptureFromCAM( CV_CAP_ANY );

cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_WIDTH, 640);
cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT, 480);
cvSetCaptureProperty(capture, CV_CAP_PROP_FPS, 30);

assert( capture != NULL );

IplImage *img;

cvNamedWindow( "CaptureFromCam", CV_WINDOW_AUTOSIZE );

while((img=cvQueryFrame(capture)) != NULL)
{
    cvShowImage( "CaptureFromCam", img );
    if( cvWaitKey(33) == 27 )
    {
        cvDestroyWindow( "CaptureFromCam" );
        break;
    }
}
```

Listing 1.6. Primjer programa za prikaz videa snimljenog kamerom

- a) Analizirati i komentirati Listing 1.6.
- b) Dodati gumb “**View video**” koji prikazuje video snimljen web kamerom spojenom na računalo.