



KoNaR

KOŁO NAUKOWE ROBOTYKÓW

**Krótki wstęp do biblioteki OpenCV.
Jak wykorzystać kamerkę internetową
do przetwarzania obrazów.**

Wrocław, 13.10.2008r.

Autor:

Filip Romanowski (140401@student.pwr.wroc.pl)

1.	Wstęp.....	3
2.	Instalacja kamerki internetowej.....	3
3.	Instalacja biblioteki OpenCV.....	3
4.	Pierwszy program.....	3
5.	Przykładowa aplikacja.....	5
6.	Linki.....	6

1. Wstęp.

Biblioteka OpenCV, czyli Open Source Computer Vision Library została napisana w języku C/C++ i jest jedną z najlepszych darmowych bibliotek do przetwarzania obrazów. Posiada ona ogromne możliwości. Jest ona zoptymalizowana pod kątem operacji w czasie rzeczywistym. Posiada własny system zarządzania okienkami, niezależny od systemu operacyjnego. Biblioteka posiada wiele przydatnych funkcji, pozwalających m.in. na:

- przechwytywanie strumieni obrazów z kamer;
- pobieranie z tychże strumieni obrazów pojedynczych klatek;
- zapis /odczyt strumieni obrazów a także pojedynczych klatek na/z dysku;
- wykonywanie różnych operacji na obrazach (operacje logiczne, spłot, zmiana przestrzeni kolorów, np. BGR → HSV);
- wyświetlanie wyników transformacji w oknach;
- rysowanie w oknach zarówno figur geometrycznych jak i dowolnych kształtów;
- reagowanie na zdarzenia od klawiatury oraz myszy.

Biblioteka ta w połączeniu z tanią kamerką internetową pozwala na ciekawe eksperymenty nie tylko z obrazami zapisanymi na dysku, ale także na o wiele ciekawsze operacje na obrazie pobieranym w czasie rzeczywistym z kamerki internetowej.

2. Instalacja kamerki internetowej.

Do eksperymentów wybrano kamerkę internetową firmy AVer Media, model TC-300, ze względu na niską cenę (zakupiono używany egzemplarz za 10zł na jednym z popularnych serwisów aukcyjnych). Można jednak śmiało powiedzieć, że każda kamerka internetowa z interfejsem USB będzie się nadawała.

Po zainstalowaniu sterowników (czasami jest spory problem ze znalezieniem sterowników jeśli sprzedający nie dostarczył ich wraz z urządzeniem; należy wziąć to pod uwagę), można sprawdzić czy kamerka działa poprawnie. Aby to zrobić instalujemy program do monitoringu pomieszczeń Ferro CCTV 2.2, którego demo (2.2 MB) możemy bezpłatnie pobrać z internetu. Po uruchomieniu programu i odpowiednim skonfigurowaniu, w oknie powinien pojawić się obraz z naszej kamery.

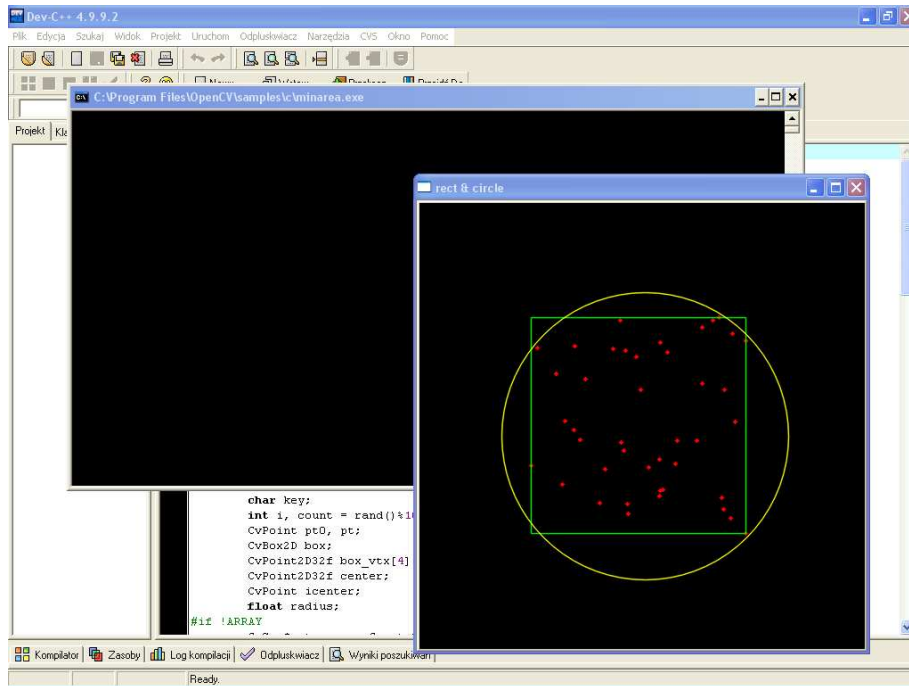
3. Instalacja biblioteki OpenCV.

Cały proces instalacji biblioteki zarówno w systemie Windows jak i Linux został dokładnie opisany w [3]. Jest tam pokazane jak rozpocząć tworzenie projektów przy użyciu różnych środowisk (tj. Visual C++, C++ Builder, Eclipse itd.). Niestety w momencie tworzenia tego raportu strona [3] była nie w pełni dostępna. Dla potrzeb niniejszego raportu wykorzystano DevCpp (jak połączyć DevCpp z OpenCV opisano w [9]).

4. Pierwszy program.

Jako że raport ten ma na celu pokazanie możliwości biblioteki OpenCV, pominięto budowę interfejsu graficznego, pozostając przy prostym interfejsie tekstowym (konsola). Sprawę odpowiedniego interfejsu, który z pewnością ułatwia pracę z programem, pozostawiono czytelnikowi.

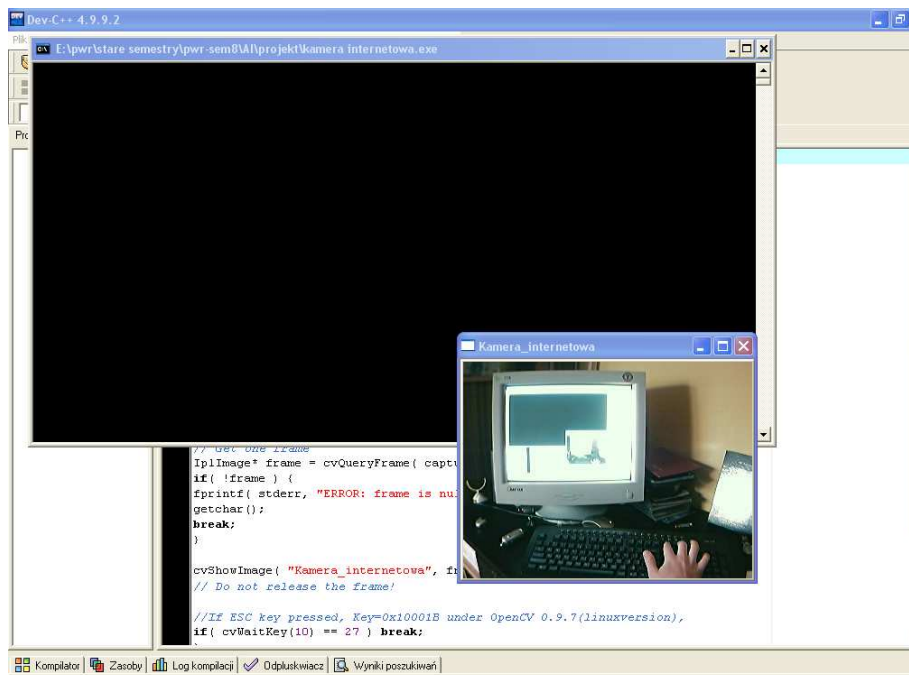
Po poprawnym zainstalowaniu biblioteki OpenCV, oraz skonfigurowaniu środowiska programistycznego DevCpp, można sprawdzić poprawność wykonanych czynności kompilując przykładowy program. W programie DevCpp otwieramy jeden z przykładowych programów jakie zawiera instalacja biblioteki OpenCV (np. program minarea, znajdujący się w folderze C:\Program Files\OpenCV\samples\c. Po skompilowaniu i uruchomieniu programu ekran komputera przedstawia się jak na ryc.1.



Ryc. 1 Uruchomiony program minarea

W oknie tym widać w tle edytor środowiska DevCpp, ponad tym znajduje się konsola tekstowa, a na samym wierzchu jest okno zarządzane przez bibliotekę OpenCV w którym możemy wyświetlać porządane obrazy.

Nadszedł teraz moment kiedy można wreszcie skorzystać z kamery internetowej. Program przedstawiony na listingu 1 jest najprostszym chyba sposobem wyświetlenia tego co udostępnia nam kamera internetowa w oknie tworzonym przez bibliotekę OpenCV. Przed uruchomieniem programu należy oczywiście pamiętać o włączeniu kamery. Wynik tego programu widać na ryc. 2 zamieszczonej poniżej.



Ryc. 2 Ekran z uruchomionym programem z listingu 1

Listing 1

```
#include "cv.h"
#include "highgui.h"
#include <stdio.h>

// A Simple Camera Capture Framework
int main() {

// Zainicjowanie przechwylenia strumienia video
CvCapture* capture = cvCaptureFromCAM( CV_CAP_ANY );
if( !capture ) {
fprintf( stderr, "ERROR: capture is NULL \n" );
getchar();
return -1;
}

// Stworzenie okna w którym przechwycone obrazy będą wyświetlane
cvNamedWindow( "Kamera_internetowa", CV_WINDOW_AUTOSIZE );

// Wyświetlenie w powyższym oknie przechwyconej klatki strumienia
// wszystko odbywa się w nieskończonej petli while
while( 1 ) {
// Pobierz jedna klatkę ze strumienia
IplImage* frame = cvQueryFrame( capture );
if( !frame ) {
fprintf( stderr, "ERROR: frame is null...\n" );
getchar();
break;
}

// Wyświetl pobraną klatkę
cvShowImage( "Kamera_internetowa", frame );

// Oczekiwanie przez 10 ms na wciśnięcie klawisza ESC (kod ASCII 27)
// jeśli klawisz zostanie naciśnięty, wtedy program wyskakuje
// z nieskończonej petli i kończy działanie
if( cvWaitKey(10) == 27 ) break;
}

// Zwalnia strumień video pochodzący z kamery
cvReleaseCapture( &capture );
// Niszczy okno
cvDestroyWindow( "Kamera_internetowa" );
return 0;
}
```

Wiadomo już jak wyświetlić obraz z kamerki na ekranie. Aby przetworzyć obraz polecam sięgnąć do zamieszczonych na końcu raportu linków gdzie zamieszczone są przykładowe programy, które w ciekawy sposób pokazują techniki przetwarzania obrazów.

5. Przykładowa aplikacja.

Przykładem wykorzystania biblioteki OpenCV jest aplikacja opisana w raporcie [\[8\]](#) „Rozpoznawanie niektórych znaków polskiego jednorecznego alfabetu palcowego”, powstałym na zaliczenie przedmiotu „Metody i algorytmy sztucznej inteligencji”, prowadzonego przez dra inż. Witolda Paluszyńskiego w semestrze letnim roku akademickiego 2007/2008 na Politechnice Wrocławskiej. Był to pierwszy program napisany przez autora przy użyciu biblioteki OpenCv, dlatego nie wszystko działało tak jak powinno, niemniej efekt był zadowalający.

6. Linki.

- [1] <http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/index.html>
- [2] <http://www.site.uottawa.ca/~laganier/tutorial/opencv+directshow/>
- [3] opencvlibrary.sourceforge.net
- [4] <http://www.xpercept.com/opencv.htm>
- [5] http://www.seas.upenn.edu/~bensapp/opencvdocs/ref/opencvref_cv.htm
- [6] http://www710.univ-lyon1.fr/~bouakaz/OpenCV-0.9.5/docs/ref/OpenCVRef_ImageProcessing.htm
- [7] http://vision.cis.udel.edu/opencv/ref/OpenCVRef_highgui.htm
- [8] http://sequoia.ict.pwr.wroc.pl/~witold/aiarr/2008_projekty/palcowy/#8
- [9] <http://www.cypax.net/tutorials/opencv/index?language=en>