Instrukcja	**Zaawansowane przetwarzanie obrazów**
5	**Temat**: Rozpoznawanie wzorców w obrazie Przygotował: mgr inż. Tomasz Michno

1 Wstęp

Biblioteka OpenCV posiada zaimplementowane funkcje służące do rozpoznawania wzorców. Najprostszą z nich jest cvMatchTemplate(), niestety posiada ona poważną wadę którą jest czas wykonania. Z tego względu warto skorzystać z funkcji FastMatchTemplate dostępnej dla języka C++ pod adresem `http://opencv.willowgarage.com/wiki/FastMatchTemplate`. Jest ona bardzo szybka oraz pozwala wykrywać wiele wystąpień dane wzorca w jednym wykonaniu. Wymaga dodania pliku nagłówkowego vector.

```
bool FastMatchTemplate( const IplImage& source,
const IplImage& target,
vector<CvPoint>* foundPointsList,
vector<double>* confidencesList,
int matchPercentage = 70,
bool findMultipleTargets = true,
int numMaxima = 5,
int numDownPyrs = 2,
int searchExpansion = 15 );
```

Najważniejsze parametry:

source, target - obrazek źródłowy oraz wzorzec

foundPointsList - wektor zawierający znalezione punkty, w których znajduje się wzorzec

confidencesList - wektor zawierający stopnień pewności dla każdego znalezionego wystąpienia (0-100)

matchPercentage - minimalny poziom rozpoznania wzorca

findMultipleTargets - wartość true oznacza wyszukiwanie wielu wystąpień

```
1 IplImage* marker = cvLoadImage("marker.png");
2 vector<CvPoint> foundPointsList;
3 vector<double>  confidencesList;
4 if( !FastMatchTemplate( *image,
5                          *marker,
6                          &foundPointsList,
```

```
                                &confidencesList, 50, true, 15 ) ){
  printf( "\nERROR: Fast match template failed.\n" );
  return 3;
}
for(int i=0; i<foundPointsList.size(); i++){
    cout<<"x="<<foundPointsList[i].x<<", y="<<foundPointsList[i].
        y<<endl;
    cvCircle(image, foundPointsList[i], 64, CV_RGB(255, 0, 0 ) );
}
```

Listing 1: Rozpoznawanie wzorców

W Listingu 1 używana jest również funkcja cvCircle rysująca okrąg:

`void cvCircle(CvArr* img, CvPoint center, int radius, CvScalar color, int thickness=1, int lineType=8, int shift=0)`
gdzie:
img - obrazek na którym zostanie narysowany okrąg
center - środek okręgu
radius - promień okręgu
color - kolor linii (można użyć funkcji CV_RGB(r,g,b))
thickness - grubość linii (jeśli wartość ujemna, okrąg jest wypełniany)
lineType - typ linii (dozwolone typy: 8, 4, CV_AA)
shift - liczba bitów ułamkowych w punkcie środka okręgu i promieniu

2 Zadanie

Napisz program, który będzie rozpoznawał dowolną liczbę 4 wzorców (uproszczony rysunek) na obrazie wczytanym z dysku. W miejscu rozpoznania powinien być rysowany obrazek w pełnej jakości (nieuproszczony), za pomocą cvGet2D() i cvSet2D().