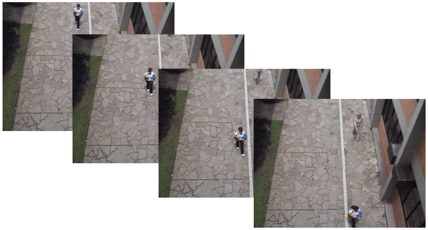
Tracking with OpenCV

Basic tracking using Background subtraction

Nestor Salamon
N.ZiliottoSalamon@tudelft.nl

Tracking

Tracking: a line of travel or motion; a course or route followed;



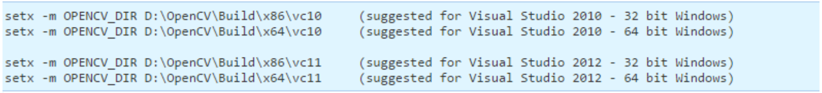
OpenCV

- OpenCV (Open Source Computer Vision Library) is an open source **computer vision** and machine learning software library
- More than 2500 optimized algorithms
- C++, C, Python, Java and MATLAB
- Windows, Linux, Android and Mac OS

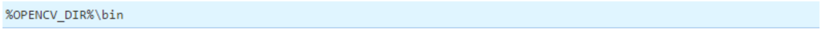
Download OpenCV: `http://opencv.org/downloads.html`

Environment Setup

Following OpenCV installation tutorial..



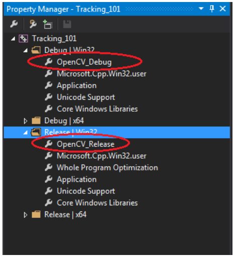
```
setx -m OPENCV_DIR D:\OpenCV\Build\x86\vc10     (suggested for Visual Studio 2010 - 32 bit Windows)
setx -m OPENCV_DIR D:\OpenCV\Build\x64\vc10     (suggested for Visual Studio 2010 - 64 bit Windows)

setx -m OPENCV_DIR D:\OpenCV\Build\x86\vc11     (suggested for Visual Studio 2012 - 32 bit Windows)
setx -m OPENCV_DIR D:\OpenCV\Build\x64\vc11     (suggested for Visual Studio 2012 - 64 bit Windows)
```



```
%OPENCV_DIR%\bin
```

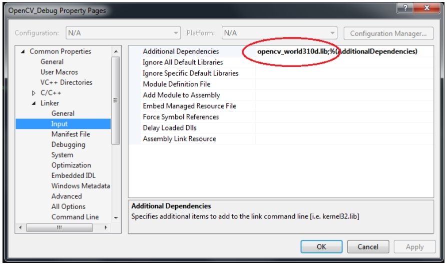
Environment Setup

OpenCV installation tutorial? Do link *opencv_world*310(*d*).*lib*



Property Manager - Tracking_101

- Tracking_101
  - Debug | Win32
    - OpenCV_Debug
    - Microsoft.Cpp.Win32.user
    - Application
    - Unicode Support
    - Core Windows Libraries
  - Debug | x64
  - Release | Win32
    - OpenCV_Release
    - Microsoft.Cpp.Win32.user
    - Whole Program Optimization
    - Application
    - Unicode Support
    - Core Windows Libraries
  - Release | x64



OpenCV_Debug Property Pages

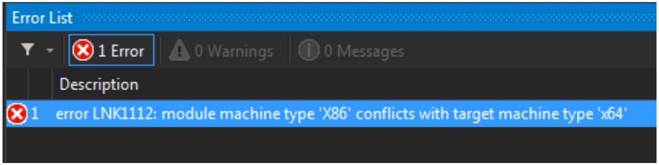
Configuration: N/A Platform: N/A

Additional Dependencies	opencv_world310d.lib;%(AdditionalDependencies)
Ignore All Default Libraries	
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

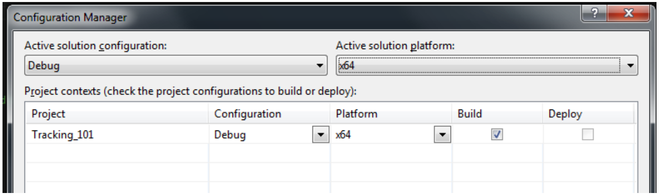
Additional Dependencies
Specifies additional items to add to the link command line [i.e. kernel32.lib]

Environment Setup

OpenCV installation tutorial? Win32 Projects in x64



1 error LNK1112: module machine type 'X86' conflicts with target machine type 'x64'



Active solution configuration: Debug

Active solution platform: x64

Project contexts (check the project configurations to build or deploy):

Project	Configuration	Platform	Build	Deploy
Tracking_101	Debug	x64	✓	

Ready, go!

```
findContours(img, cont, h, CV_RETR, CV_CHAIN, Point(0,0));
```

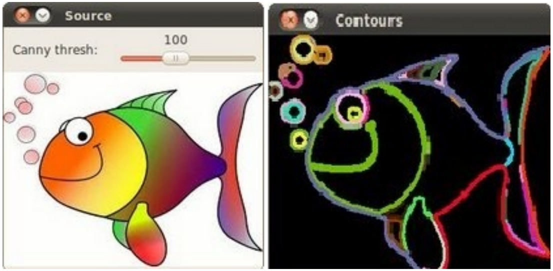
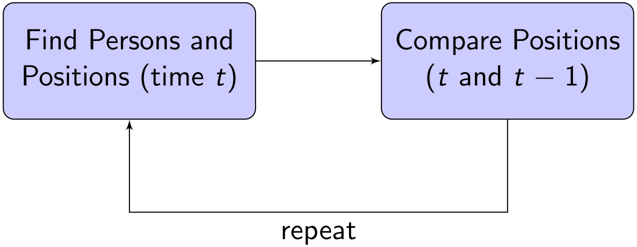


Figure: Simple result with one function. Image from: `http://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/find_contours/find_contours.html`

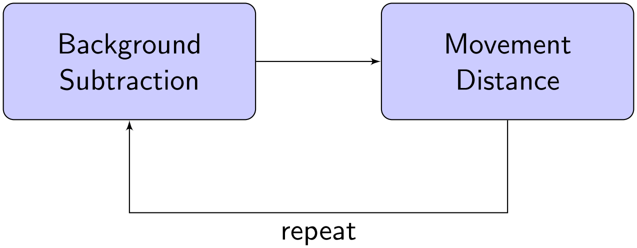
Tracking Pipeline

For each frame of the video sequence:


Find Persons and Positions (time t)
Compare Positions (t and $t-1$)
repeat

Tracking Pipeline

For each frame of the video sequence:



Read from images

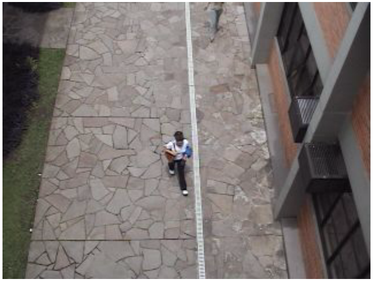
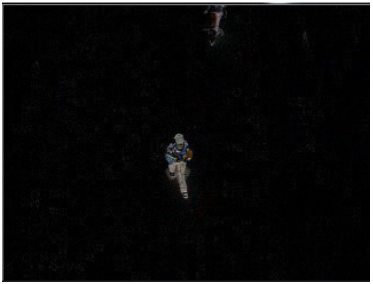
```
vector<String> files;
vector<Mat> imgs;
glob(path, files, true);
for (size_t k = 0; k < files.size(); k++)
{
  Mat im = imread(files[k]);
  //do something
}
```

Live/Video capture

```
VideoCapture cap(0);
for(;;)
{
  Mat frame;
  cap >> frame;
  imshow("frame", frame);
  if(waitKey(30) >= 0) break;
}
```

Background Subtraction

Separates the Region of Interest from the background

Background Estimation

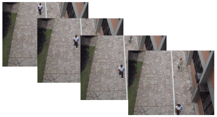
Static scene, with no people walking by: take the first frame.

```
Mat bg = imgs[0];
```

Background Estimation

People moving all around the video sequence: frame average

```
Mat acum;
for (size_t k = 0; k < files.size(); k++)
{
  acum = acum + imread(files[k]);
}
Mat bg = acum / files.size();
```

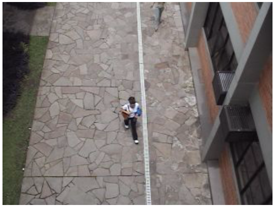


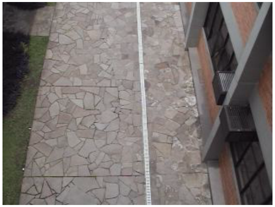
Background Subtraction

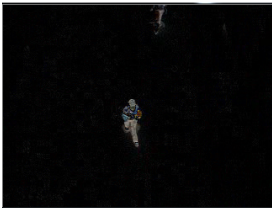
Take the diference between the current frame and the background

```
Mat bg = imgs[0];
for (int i = 1; i < imgs.size(); i++)
{
  Mat im = imgs[i];
  Mat diff = abs(im - bg);
  //...
}
```

Background Subtraction





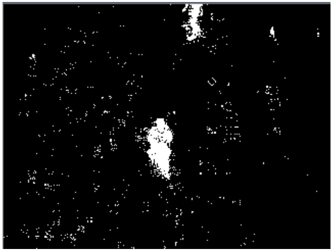


Background Subtraction

Highlighting the differences

```
int threshold = 15;
cvtColor(diff, bin, CV_BGR2GRAY);
threshold(bin, bin, threshold, 255, 0);
```

Background Subtraction

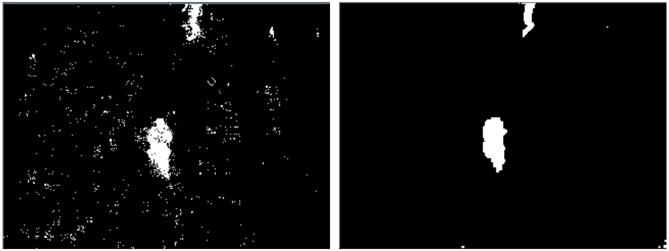


Morphological Operations

Applying a Structuring element to transform (and clean) images

```
Mat element = getStructuringElement(...);
erode(bin, bin, element);
dilate(bin, bin, element);
```

Morphological Operations

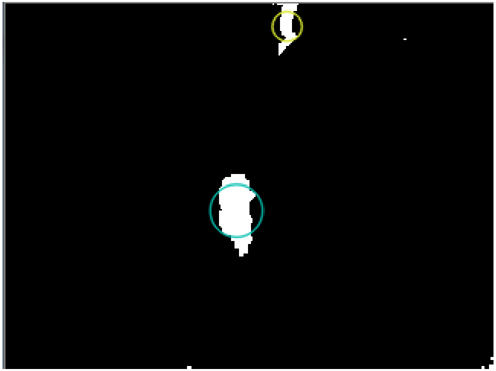


Blob Detection

Detecting connected components - or the regions of interest

```
vector<KeyPoint> keypoints;
Ptr<FeatureDetector> detector =
  SimpleBlobDetector::create();
detector->detect(bin, keypoints);
```

Blob Detection



Blob Detection

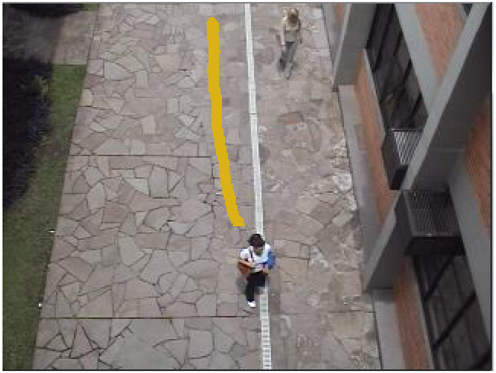
```
SimpleBlobDetector::Params params;
params.minDistBetweenBlobs = 1.0f;
params.filterByInertia = false;
params.filterByCircularity = false;
params.filterByConvexity = false;
params.filterByArea = true;
params.minArea = 200.0;
params.maxArea = 4000.0;
detector = SimpleBlobDetector::create(params);
```

Tracking

Link the person/object position in the current frame t to the previous one $t-1$.

```
Point last_p;
KeyPoint kp = keypoints[0];
p = Point(kp.pt.x, kp.pt.y);
if (distance(p, last_p) < max_dist)
{
  line(fin, p, last_p, color, 1, 8,0);
  last_p = p;
}
```

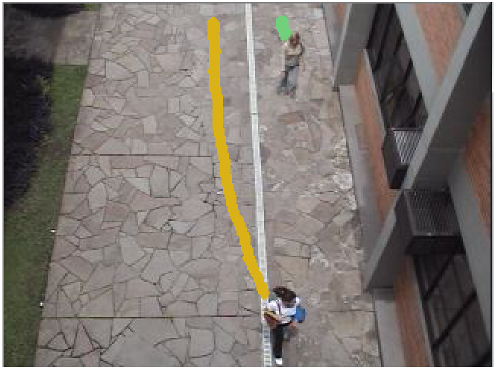
Tracking



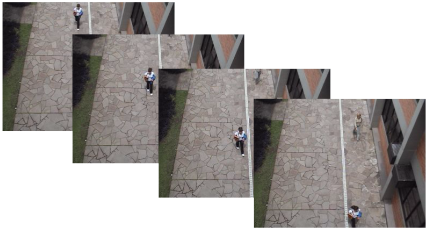
Tracking

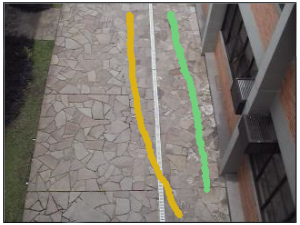
```
Vector<Point> last_ps;
for (int k = 0; k < keypoints.size(); k++)
{
  for (int z = 0; z < last_ps.size(); z++)
  {
    if (distance(p, last_ps[z]) < max_dist)
    {
      line(fin, p, last_ps[z], color, 1, 8,0);
      last_ps[z] = p;
      tracked = true;
    }
  }
  if (!tracked) last_ps.push_back(p);
}
```

Tracking



Results





Final thoughts

- Few lines of code, people detection and tracking
- Performance: can be applied in real-time
- Others approaches: KNN (find people) and Template Matching (tracking), Image Moments

Download code:
`https://github.com/salamon/TrackingBasics`

Questions?

Download code:
`https://github.com/salamon/TrackingBasics`

Tracking with OpenCV

Basic tracking using Background subtraction

Nestor Salamon
N.ZiliottoSalamon@tudelft.nl