



## Segmentation for fixe Camera added to the AVED project

Segmentation added the 6-5-2006 on nanomia /home/aved/jerome/

Usage: `runclip -s ExtractForegroundBW-0:40 my_clip.mov` OR  
`runclip -s ExtractForegroundBW-my_background.ppm my_clip.mov`

### Project reference: AVED-06.05.2006

Here are steps and files considered when the segmentation has been added to the AVED project: Refer to Nicholas Howe publication (*Better Foreground segmentation through Graph Cuts*) for details on the algorithm.

ExtractForegroundBW segmentation takes for parameter a frame range of empty frames or a picture which represent the background. This algorithm has been added to the system in the Eye In The See project context, this algorithm allows to provide a better segmented scene to the Saliency algorithm.

#### 1. ExtractForeground files

The algorithm provided by Nicholas Howe ([nhowe@email.smith.edu](mailto:nhowe@email.smith.edu)) is located on Nanomia /home/aved/jerome/matlab/FGsegBW/. To integrate it to the AVED system, the `extractForeground.m` file has been modified. The folder (with modified files) is on Nanomia:/home/aved/jerome/matlab/graphCut/.

A version of this algorithm for color videos is located on Nanomia:/home/aved/jerome/matlab/FGsegColor/.

#### 2. Compute the mean of a background through several empty frames: `getBackgroundMean.m`

This function allows to compute the background mean of a scene through several frames. This function is called when the ExtractForegroundBW segmentation is called by the user with the frame range option. If a picture is provided, this function is ignored.

```
function getBackgroundMean (folder, input, background)
```

```
% is frames parameters a path to an image or an intervall of frames ?
```

```
fid = fopen(background);
```

```
if fid == -1; % it's an intervall format: bb-ee
```

```
    numframes = strsplit(':', background);
```

```
    first = str2int(numframes{1});
```

```
    last = str2int(numframes{2});
```

```

% Find input files
nb = 0;
fid = -1;
while fid == -1 & nb < 10
    path = [folder input];
    for t = 0: nb;
        path = [path '0'];
    end
    path = [path int2str(first) '.ppm'];
    fid = fopen(path);
    if fid == -1;
        sprintf('%s', path);
    end
    nb = nb + 1;
end

% No files found
if fid == -1
    sprintf('getBackgroundMean::Input files not found')
else
% Files found > compute average
init = 1;% result need to be initialised ?
frames = 0;% number of frames averaged
for i = first : last
    frames = frames + 1;
    path = [folder input];
    [x, y] = size(int2str(i));
    [x1, y1] = size(int2str(first));
    for t = 0: nb-y+y1-1;
        path = [path '0'];
    end

    path = [path int2str(i) '.ppm'];
    img = rgb2gray(imread(path));
    if init == 1
        init = 0;
        [x, y] = size(img);
        backgroundmean = zeros(x, y); % initialisation of the result
    end
    backgroundmean = (double(backgroundmean) + double(img));
end
backgroundmean = (double(backgroundmean)./frames);
% double > uint8 (image format)
backgroundmean = uint8(backgroundmean-1);
end
else
backgroundmean = imread(background);
end
imwrite(backgroundmean, 'background.ppm', 'PPM');
end
end

```

### 3. Add Matlab functions to the shared library: libJeromeSegmentAlgorithms.so

The 2 matlab functions have been added to the createSharedLib script as followings:

```

mcc -W lib:libJeromeSegmentAlgorithms -g applyRectangleMask
    applyRectangleImageMask
    homomorphicCanny
    adaptiveThresholding
    backgroundCanny
    graphCut/extractForegroundBW
    graphCut/getBackgroundMean

```

Nb: the library has to be on /mbarivision/lib directory.

#### 4. File mbarivision/mbarivision.C

```
// if a fixedCamera segmentation algorithm has been asked: process in first the Mean of the background
if(!strcmp("ExtractForegroundBW", detectionParms.itsSegmentAlgorithm.c_str())) {
    mxArray *path = NULL;
    mxArray *input = NULL;
    mxArray *bg = NULL;
    path = mxCreateString("./");
    input = mxCreateString("f");
    bg = mxCreateString(detectionParms.itsSegmentSEType.c_str());
    mlfGetBackgroundMean(path, input, bg); // function write the background: background.ppm
    mxArray *background = NULL;
    background = mxCreateString("background.ppm");
}

for (int curFrame = ifs->minframe(); curFrame <= ifs->maxframe(); ++curFrame)
{
    ...

    else if(!strcmp("ExtractForegroundBW", detectionParms.itsSegmentAlgorithm.c_str())) {
        mxArray *background = NULL;
        background = mxCreateString("background.ppm");
        mlfExtractForegroundBW(inputfile, background);
    }

    ....
}
```