

Potts Model Grid

load modules

```
In [4]: import opengm
import vigra
import numpy
import time
import sys
```

load data

```
In [5]: fname = "135069.jpg"
img = vigra.readImage(fname)
img = numpy.sum(img,axis=2)
img = vigra.resize(img,[s/1 for s in img.shape])
noise = numpy.random.random(img.size).reshape(img.shape)*255
print noise.shape
img += noise
img -= img.min()
img /= img.max()
print "shape", img.shape
vigra.imshow(img)
vigra.show()
```

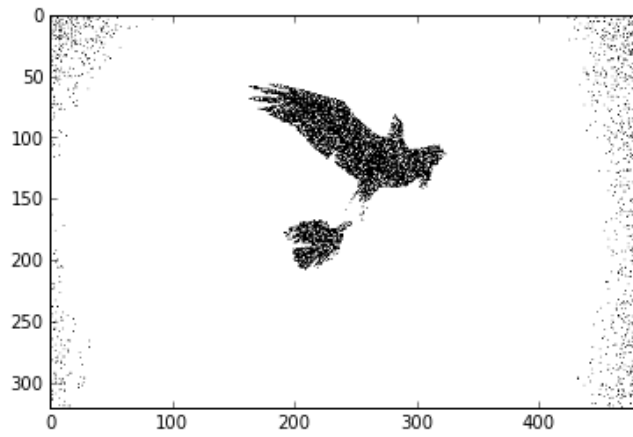
(481, 321)

shape (481, 321)



thresholding naive

```
In [6]: threshold = 0.24  
labelsNaive = img > threshold  
vigra.imshow(labelsNaive)  
vigra.show()
```



```
thresholding with potts model
```

```
In [7]: nVar = img.size
nLabelsPerVar = 2
variableSpace = numpy.ones(nVar)*nLabelsPerVar
gm = opengm.gm(variableSpace)

t0 = time.time()
# add unaries
for x in range(img.shape[0]):
    for y in range(img.shape[1]):

        energy0 = img[x,y] - threshold
        energy1 = threshold - img[x,y]
        unaryFunction = numpy.array([energy0,energy1])

        # add unary function to graphical model
        functionId = gm.addFunction(unaryFunction)

        # add unary factor to graphical model
        variableIndex = y + x*img.shape[1]

        gm.addFactor(functionId,variableIndex)

# add 2. order regularizer
# ``Potts``- regularizer
beta = 0.1
pottsFunction = numpy.zeros([2,2])
pottsFunction[0,1]=beta
pottsFunction[1,0]=beta

# add 2. order function to graphical model
# but only ONCE
pottsFunctionId = gm.addFunction(pottsFunction)

for x in range(img.shape[0]):
    for y in range(img.shape[1]):

        # add "horizontal" second order factor
        if x+1 < img.shape[0]:
            variableIndex0 = y + x*img.shape[1]
            variableIndex1 = y + (x+1)*img.shape[1]

            gm.addFactor(pottsFunctionId,[variableIndex0,variableIndex1])

        # add "vertical" facator
        if y+1 < img.shape[1]:
            variableIndex0 = y + x*img.shape[1]
            variableIndex1 = (y+1) + x*img.shape[1]
            # add "horizontal" second order factor
            gm.addFactor(pottsFunctionId,[variableIndex0,variableIndex1])

t1 = time.time()
print "build model in", t1-t0, "sek"
```

build model in 6.49472093582 sek

Alternative

```
In [9]: gm=None

t0=time.time()

unaries=numpy.ones(img.shape+(2,))
unaries[:, :, 0] = img-threshold
unaries[:, :, 1] = -img+threshold
potts=opengm.PottsFunction([2,2],0.0,beta)
gm=opengm.grid2d2Order(unaries=unaries,regularizer=potts)

t1=time.time()
print "build model in", t1-t0, "sek"
```

Inference

```
In [7]: graphCut = opengm.inference.GraphCut(gm=gm)
graphCut.infer()
labels = graphCut.arg()
labels = labels.reshape(img.shape)

vigra.imshow(vigra.taggedView(labels,'xy'))
vigra.show()
```

