Assignment 2

# 

1. See P1

$ ./P1.exe ../../Texts/Novels/DostoevskyKaramazov.txt 0.5

the, 15173

and, 11436

to, 9528

he, 8149

i, 7995

of, 7281

a, 6804

you, 6305

that, 6177

it, 5852

in, 5514

was, 4776

his, 4345

s, 3480

for, 3471

him, 3377

but, 3267

at, 3004

with, 2986

not, 2907

had, 2854

is, 2433

on, 2334

me, 2292

all, 2196

as, 2132

t, 2108

have, 1983

her, 1885

be, 1874

she, 1784

what, 1743

so, 1705

my, 1691

one, 1603

from, 1486

there, 1357

they, 1310

alyosha, 1243

this, 1241

are, 1222

by, 1205

no, 1205

will, 1168

if, 1162

been, 1105

would, 1102

up, 1091

your, 1064

only, 1055

were, 1000

said, 995

them, 987

out, 959

an, 943

now, 927

mitya, 917

man, 908

who, 898

do, 866

50.0261 %

60 words are necessary for understanding 50% of the text, representing 0.5% of the language.

$ ./P1.exe ../../Texts/Novels/DrSeuss.txt 0.5

the, 89

and, 66

i, 62

of, 45

a, 41

king, 32

that, 31

he, 23

to, 22

yertle, 19

all, 19

in, 19

turtle, 18

m, 16

turtles, 15

you, 15

one, 15

s, 12

they, 12

my, 12

was, 11

up, 11

them, 11

throne, 10

it, 10

going, 10

his, 10

but, 9

on, 9

see, 9

t, 9

down, 8

here, 8

south, 8

zax, 8

ll, 8

50.5176 %

36 words are necessary for understanding 50% of the text, representing 8.53081% of the language. (Some of these are not real words.)

# 

1. See computeEngGrad.m
2. % s =

%

% 4.5304e+06



1. See computeEngColor.m

% s =

%

% -58999407



1. See computeEng.m
2. See removeSeamV.m
3. See addSeamV.m
4. See seamV\_DP.m
5. See bestSeamV.m
6. See reduceWidth.m
7. See reduceHeight.m
8. See increaseWidth.m
9. See increaseHeight.m
10. See intelligentResize.m

Cat:

% totalCost =

%

% -1.0109e+06



Face:

% totalCost =

%

% -4.0257e+05





% sigma2 =

%

% 2.0833

%

%

% W =

%

% 1.0000 4.0000 0.7866

% 2.0000 5.0000 0.7866

% 3.0000 6.0000 0.3829

% 4.0000 7.0000 0.1153

% 5.0000 8.0000 0.7866

% 6.0000 9.0000 1.0000

% 1.0000 2.0000 0.7866

% 4.0000 5.0000 0.7866

% 7.0000 8.0000 0.7866

% 2.0000 3.0000 0.3829

% 5.0000 6.0000 0.7866

% 8.0000 9.0000 0.7866

% 4.0000 1.0000 0.7866

% 5.0000 2.0000 0.7866

% 6.0000 3.0000 0.3829

% 7.0000 4.0000 0.1153

% 8.0000 5.0000 0.7866

% 9.0000 6.0000 1.0000

% 2.0000 1.0000 0.7866

% 5.0000 4.0000 0.7866

% 8.0000 7.0000 0.7866

% 3.0000 2.0000 0.3829

% 6.0000 5.0000 0.7866

% 9.0000 8.0000 0.7866

%

%

% segm =

%

% 3 x 3 logical array

%

% 0 0 0

% 0 0 0

% 0 1 1

%

%

% e2 =

%

% 1.5524

After a while of experimentation, my favorite modification thus far is what I call “*intelligentRemove.m*.” Here, instead of intelligently finding seams of low energy to use to preserve the foreground content of the picture, this program does the opposite—it intelligently finds seams of high energy to preserve the background of the image. I chose to implement this by simply inverting the *computeEng* function by multiplying it by -1.

Before:



After:

