ssh -X [xb52@ecelinux-10.ece.cornell.edu](mailto:xb52@ecelinux-10.ece.cornell.edu)

. ~/e4

**Objective**

The objective of this lab is to gain experience with image segmentation through both *thresholding* and *region growing* techniques and to become familiar with the programming tools for both global and point operations.

### 2. Peakiness Detection (vtpeak)

The command vtpeak implements an automatic thresholding algorithm that detects the two main peaks in the histogram and selects the threshold from the minimum histogram value between these peaks.

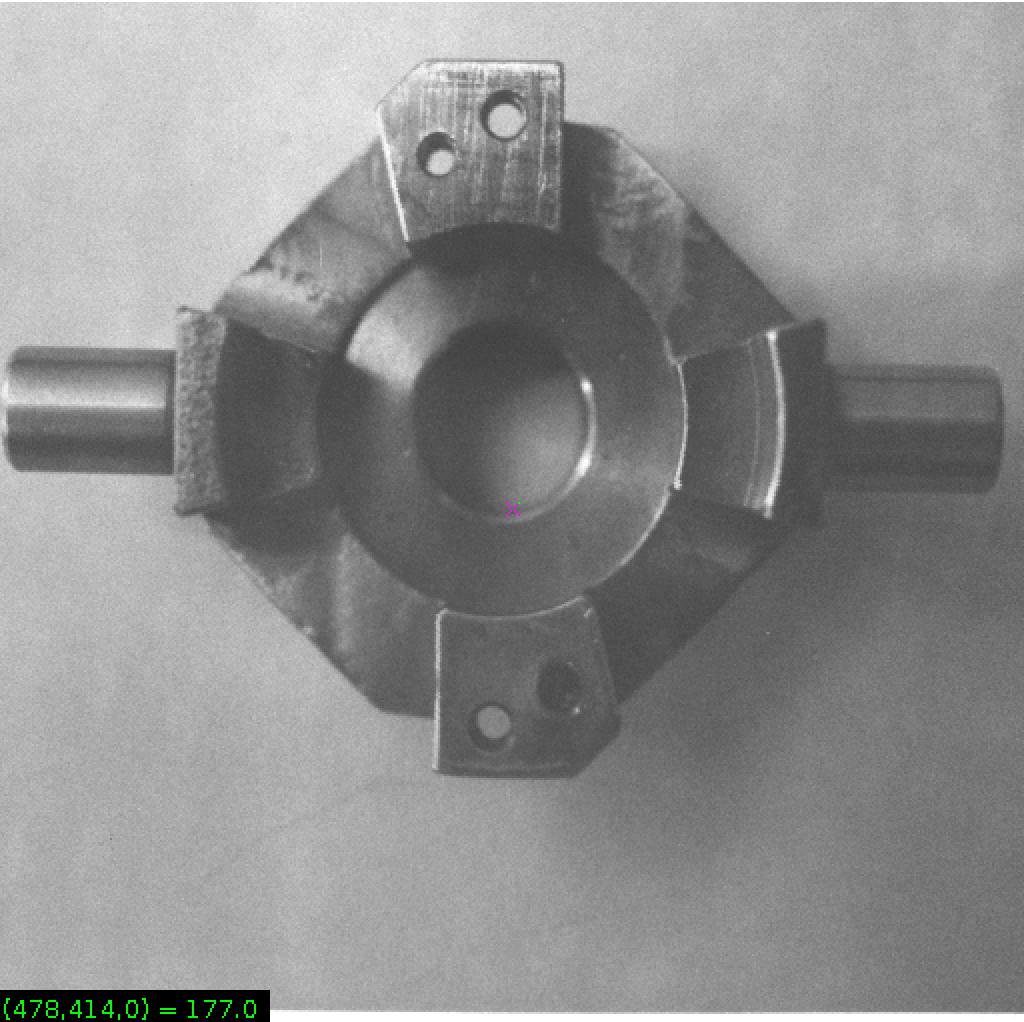
1. Compile vtpeak.c using vcc.

vcc vtpeak.c -o vtpeak

1. Test vtpeak on the image mp.vx

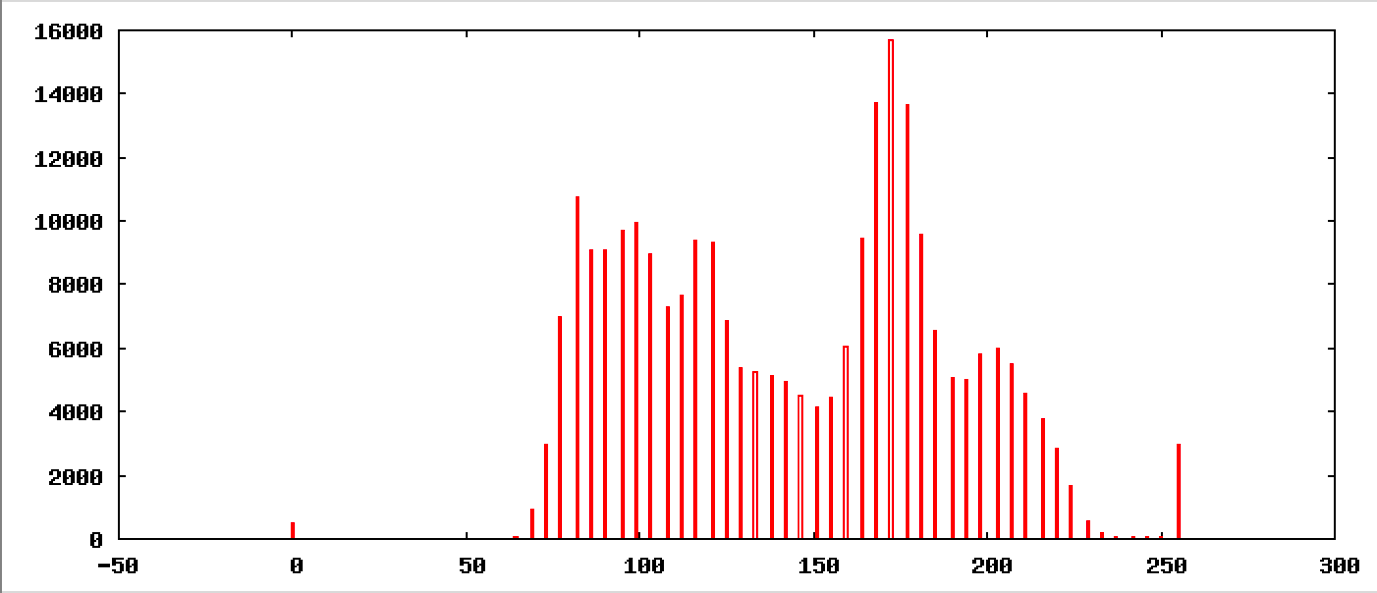
vtpeak mp.vx of=vtpeakmp.vx

view mp.vx



Display histogram of mp.vx

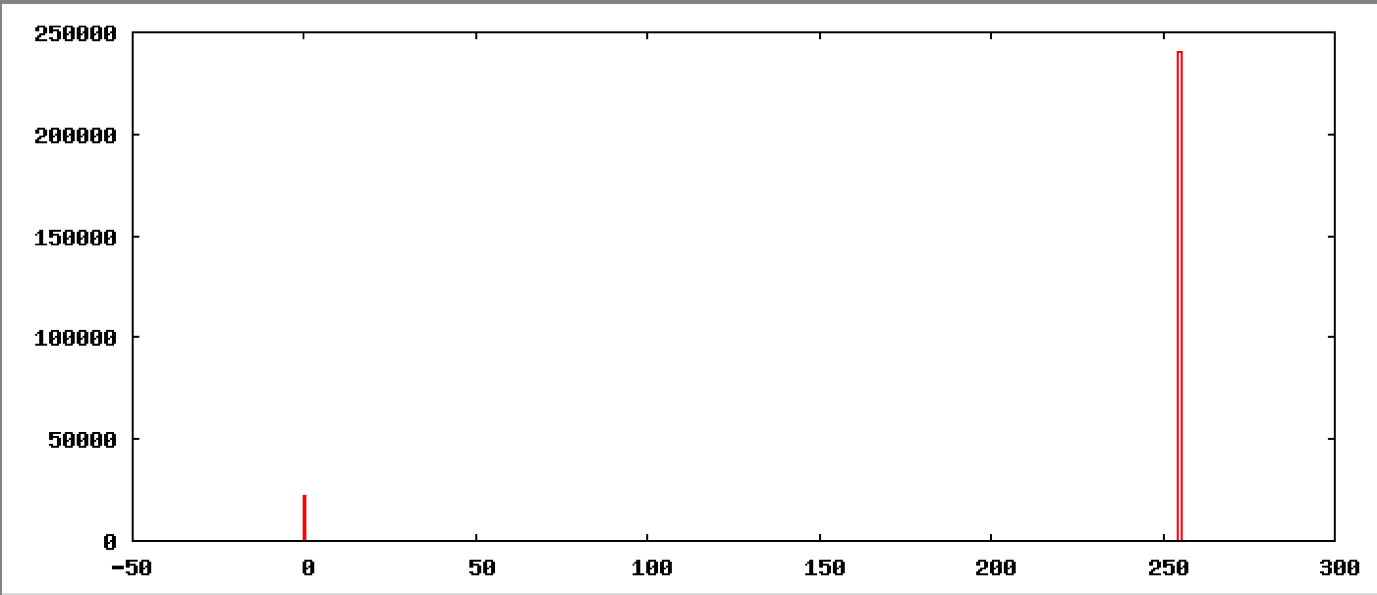
vplot -h mp.vx



View vtpeakmp.vx



Display the histogram of vtpeakmp.vx



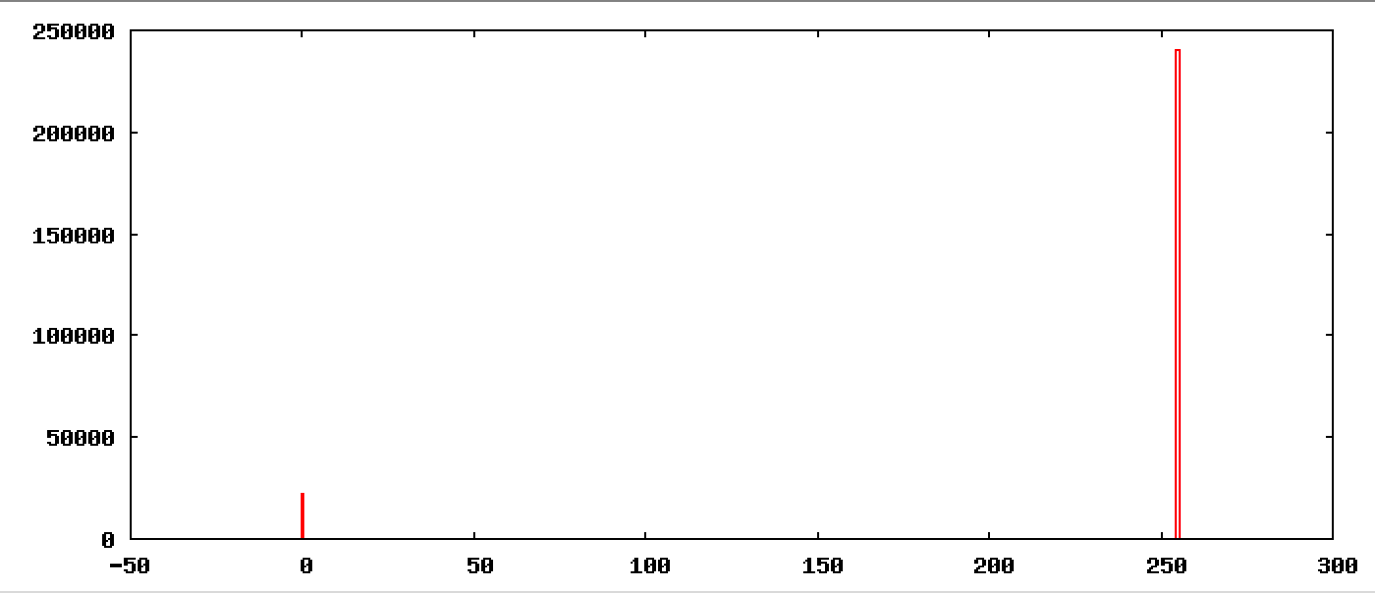
Compute and display a thresholded version of the image using vtpeak

vtpeak d=10 -v mp.vx of=mpp.vx

maxbin = 172 nxtbin = 82 thresh = 83

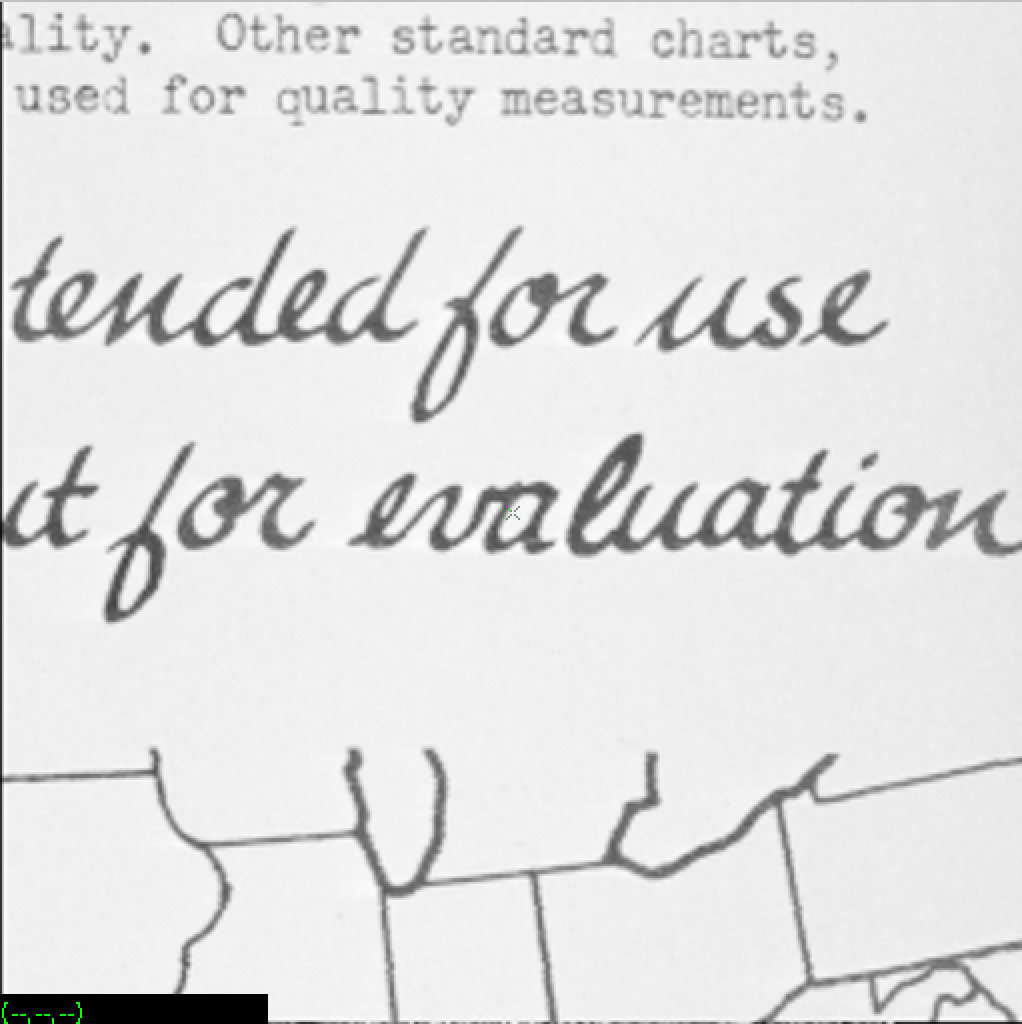
view mpp.vx

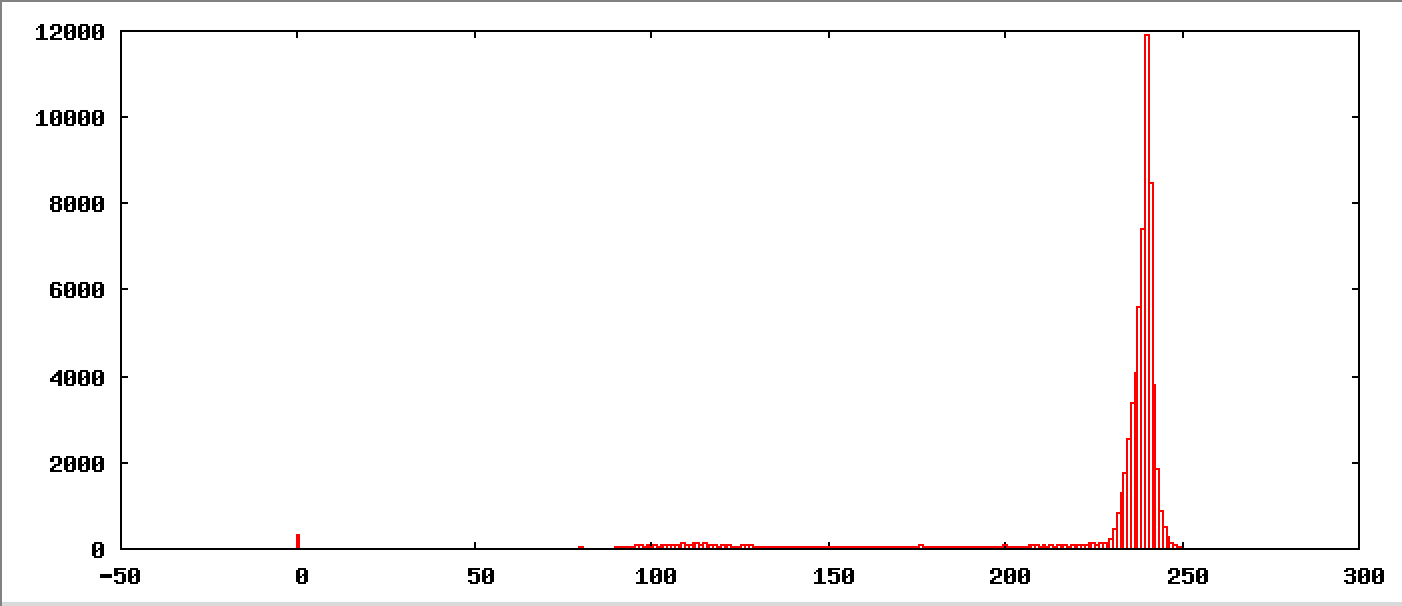


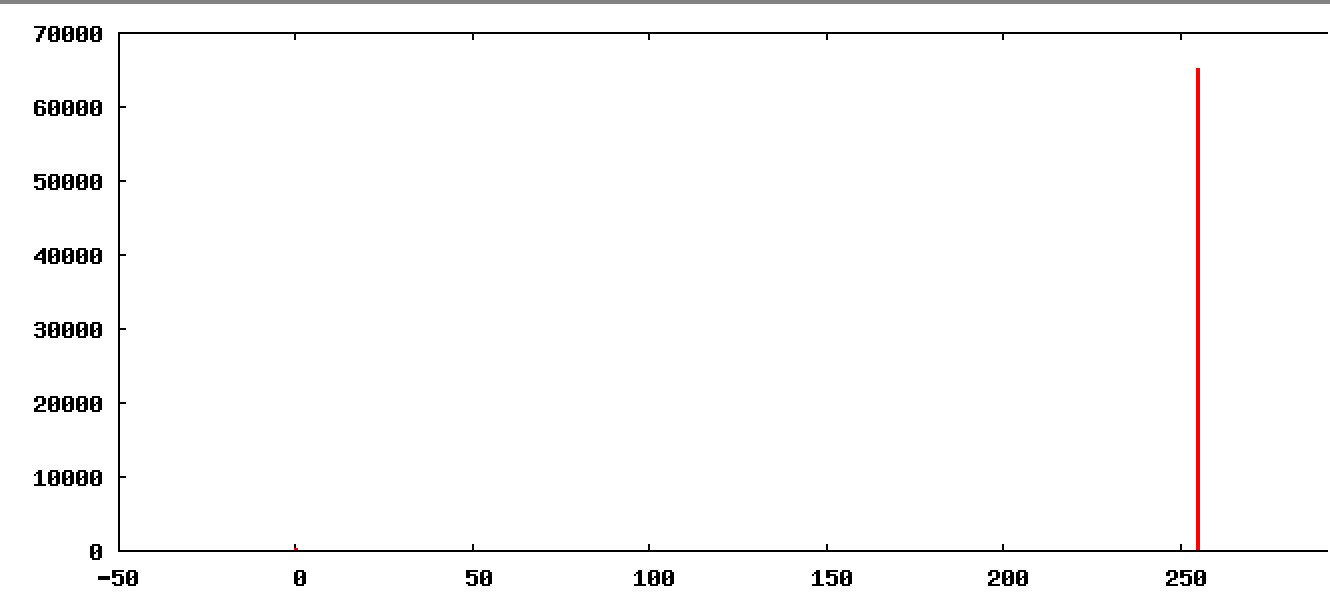
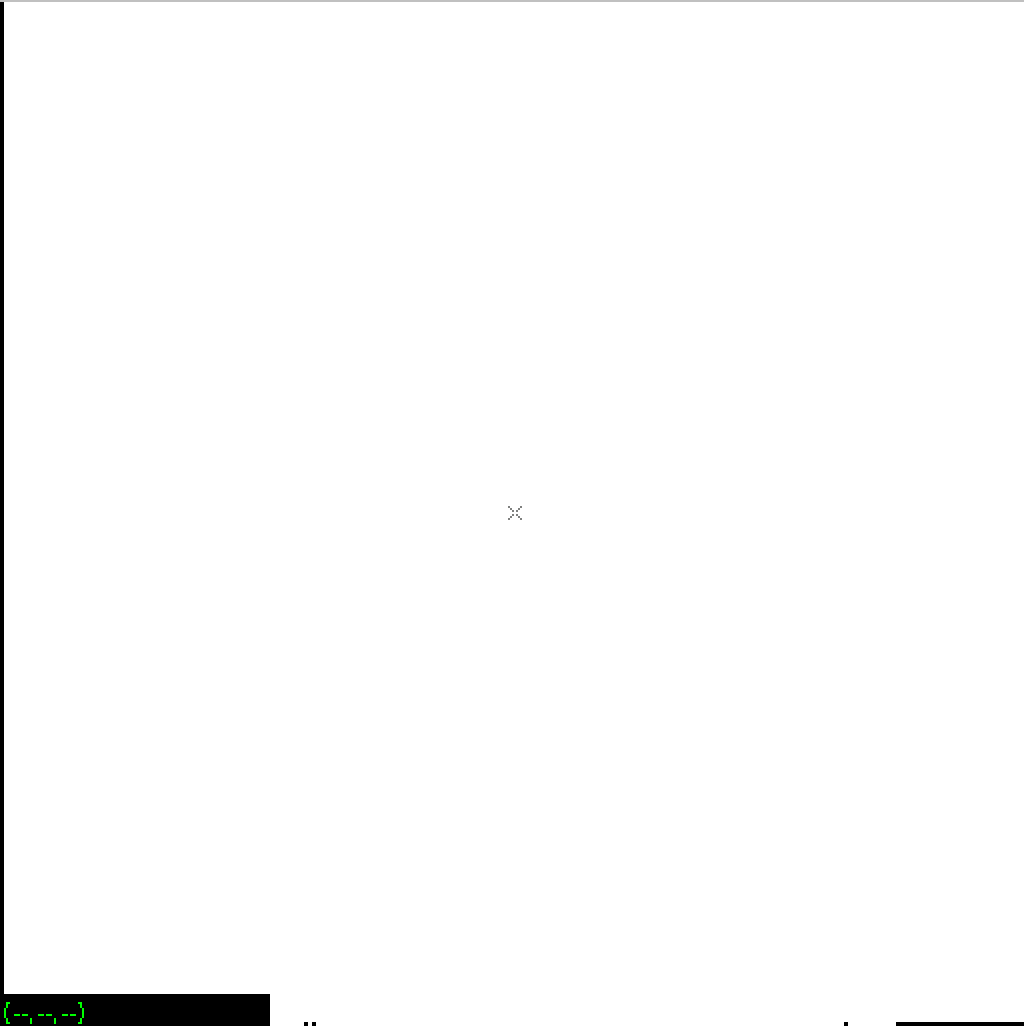


vtpeak d=10 -v facsimile of=vtpeakd10facsimile.vx

maxbin = 240 nxtbin = 0 thresh = 1

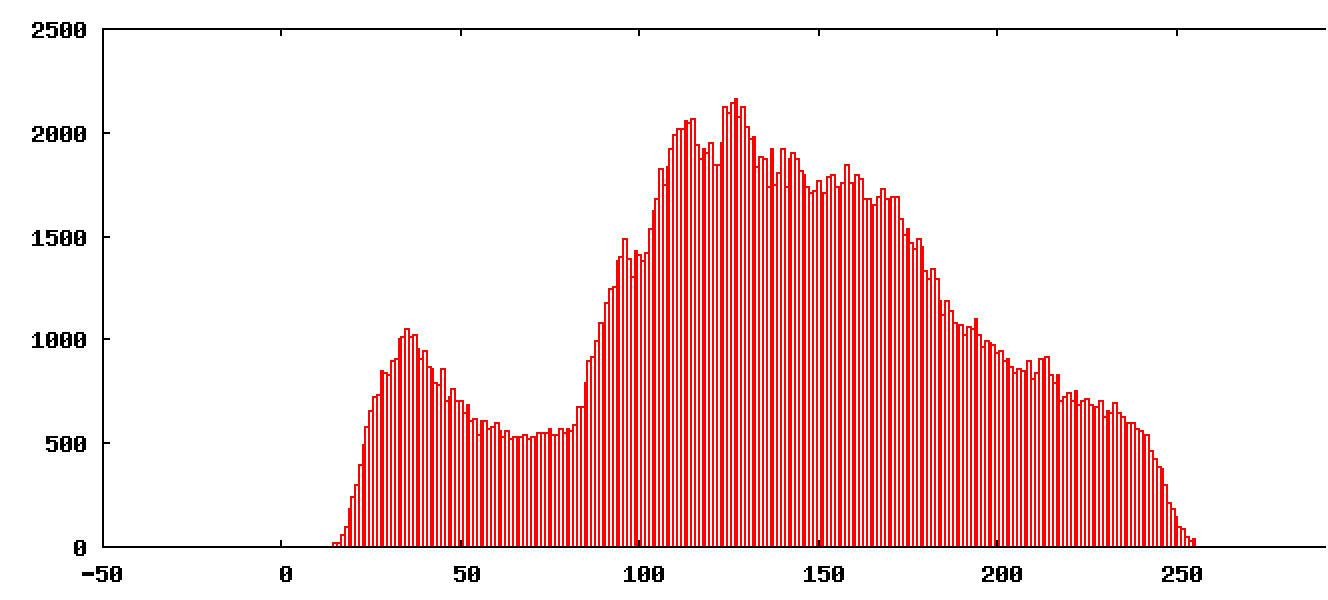


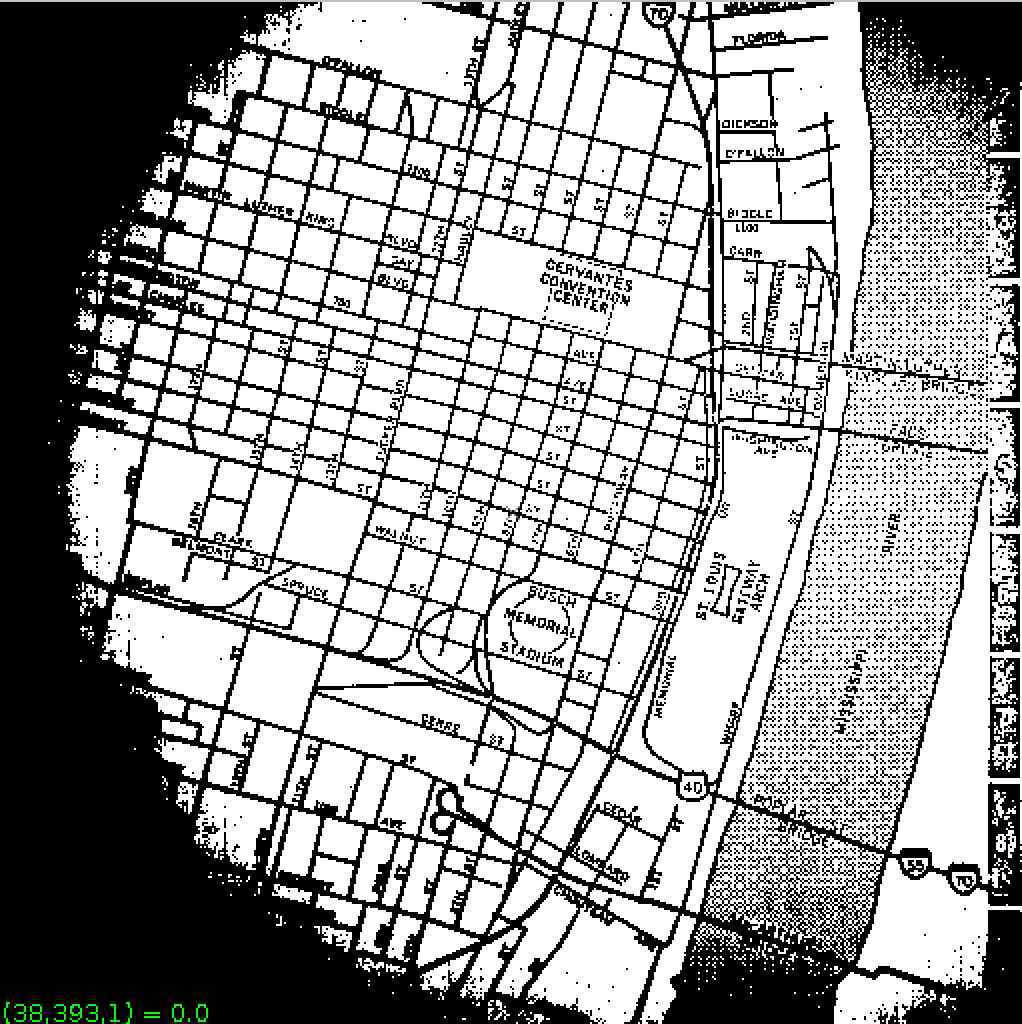




vtpeak d=10 -v map of=vtpeakd10map.vx

maxbin = 127 nxtbin = 115 thresh = 122

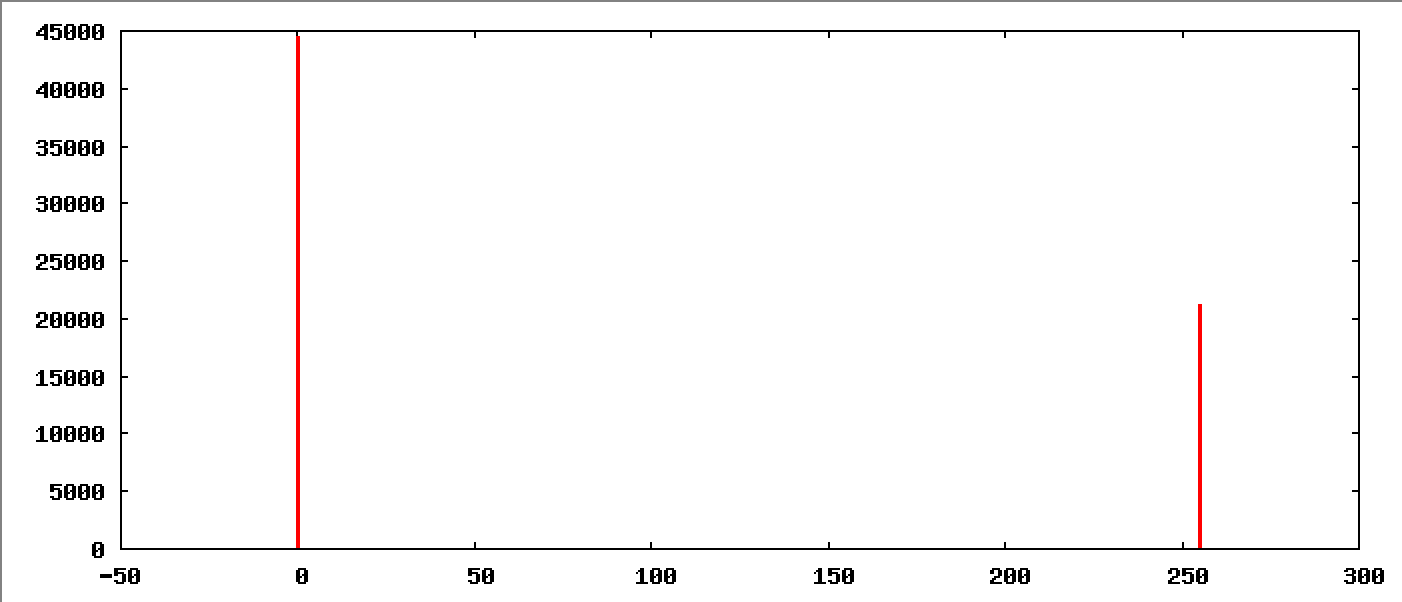
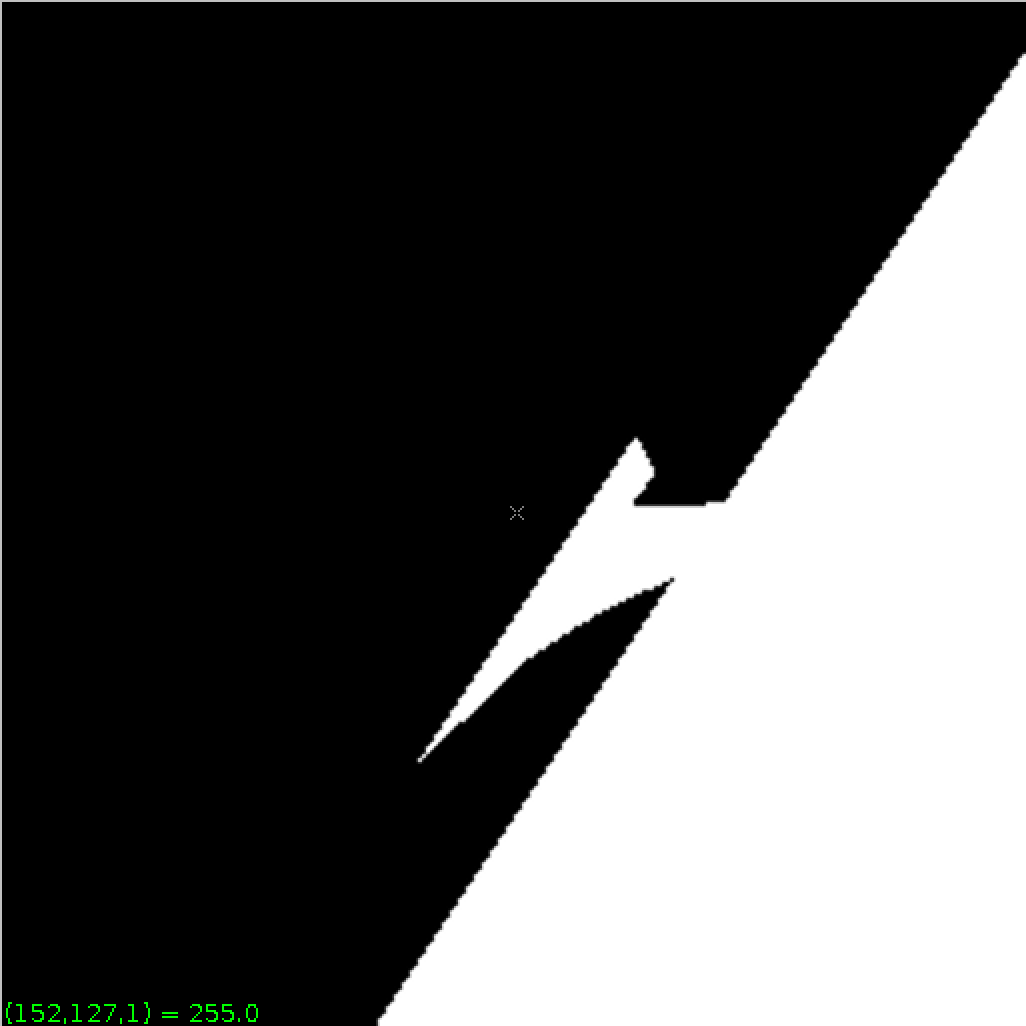
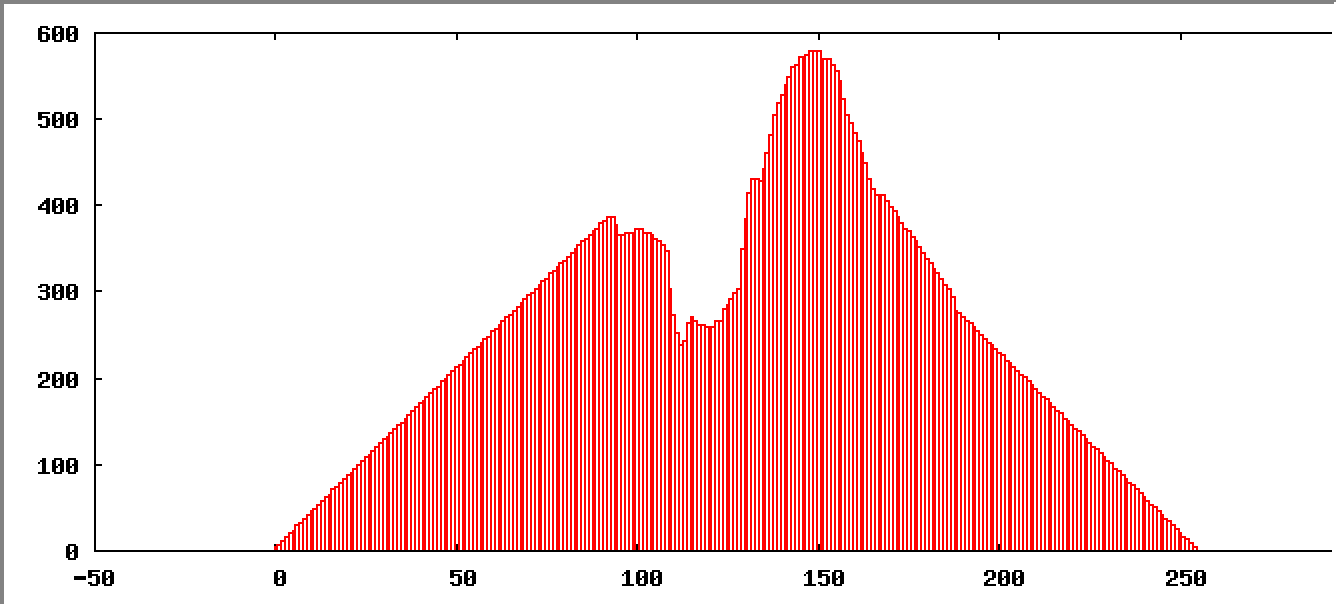
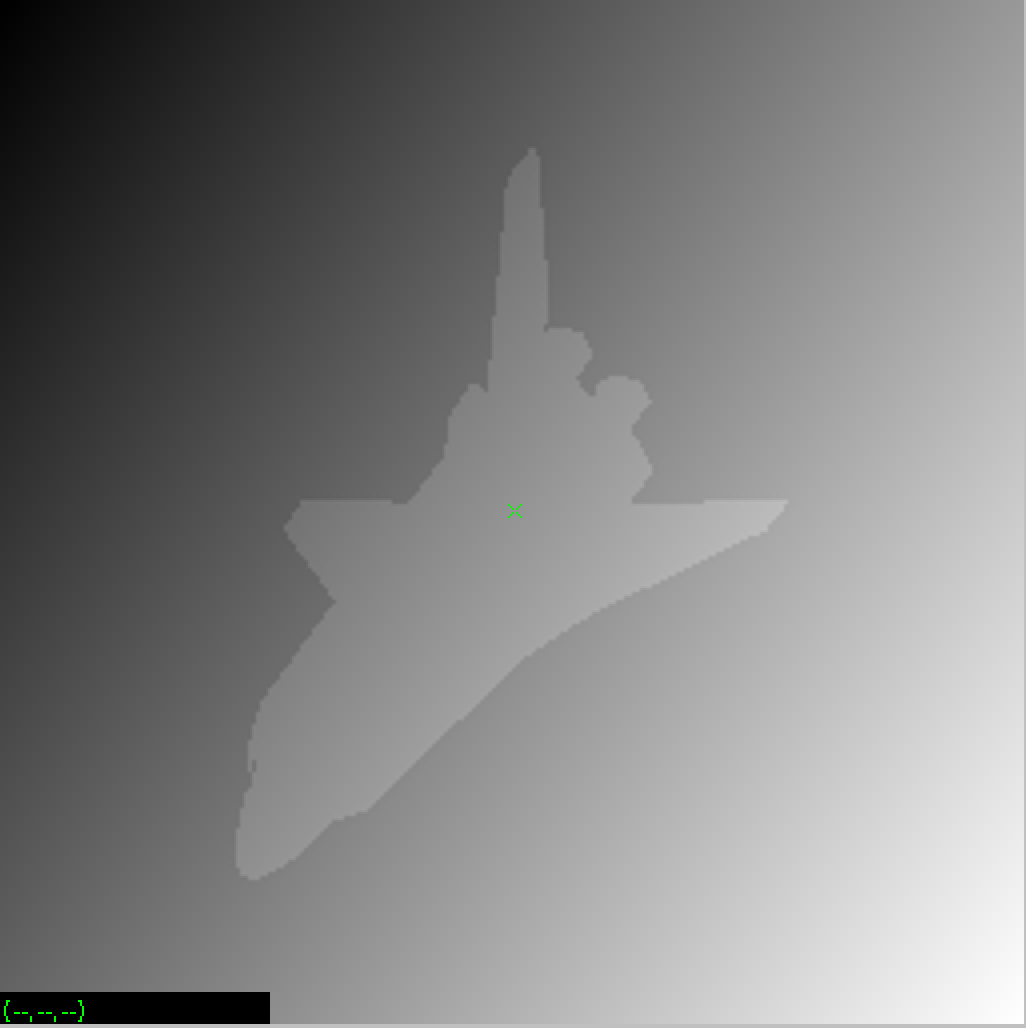






vtpeak d=10 -v shtl.vx of=vtpeakd10shtl.vx

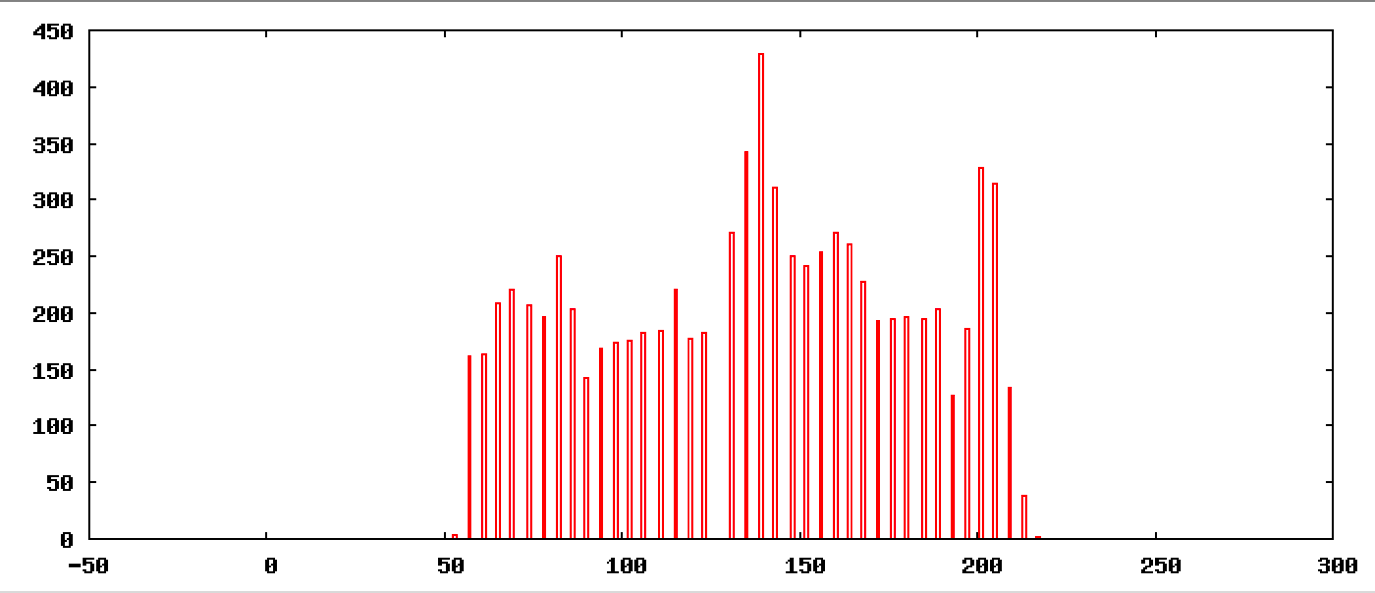
maxbin = 148 nxtbin = 158 thresh = 158

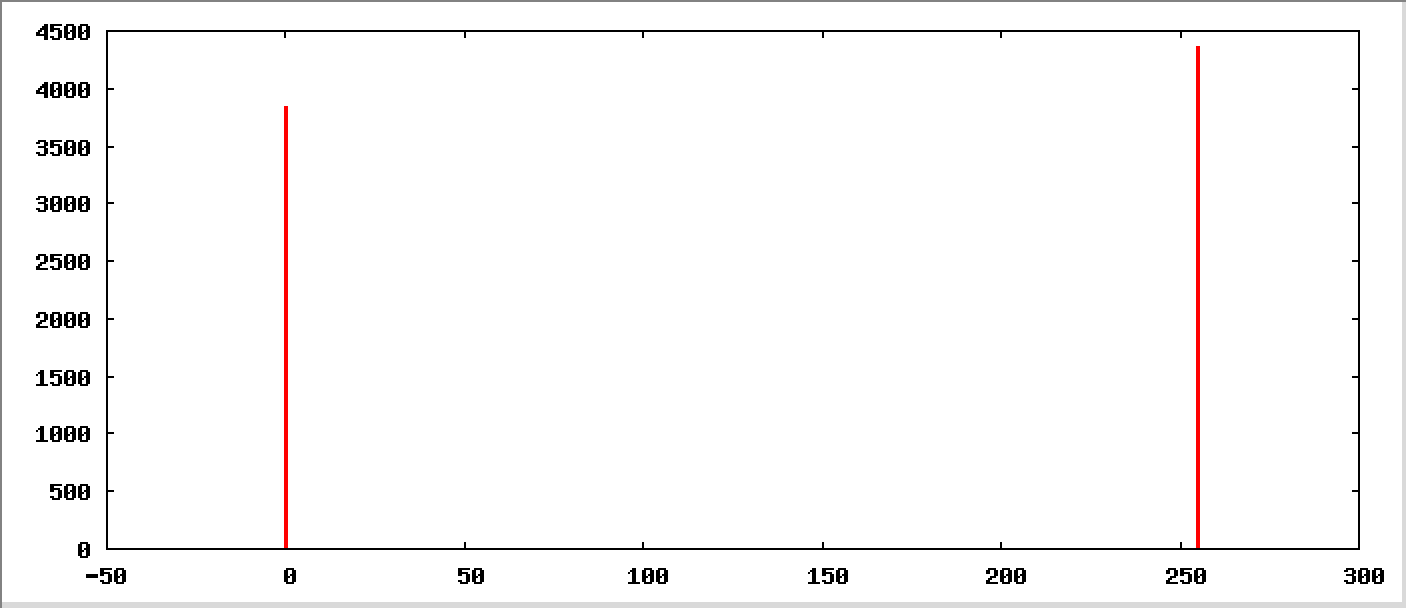


### 3. Iterative Threshold Selection (vits)

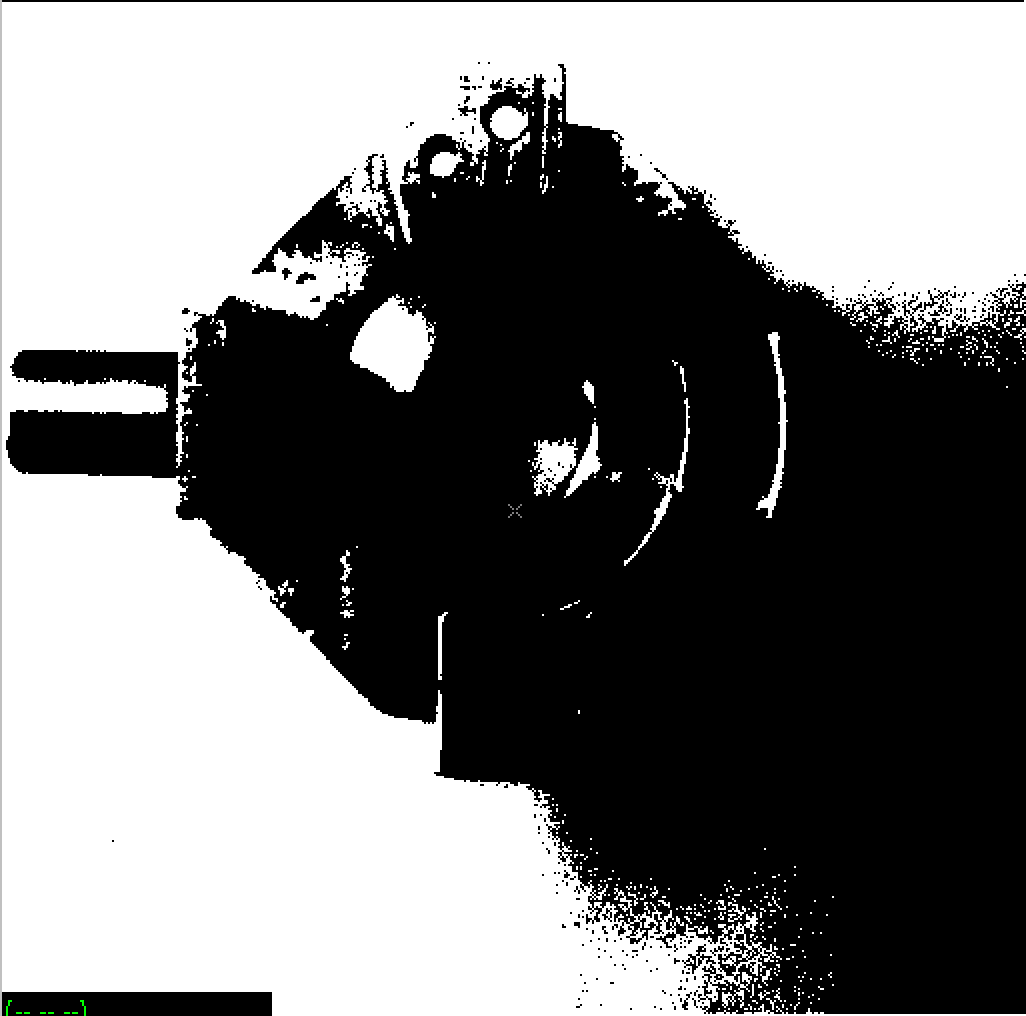
vcc vits.c -o vits

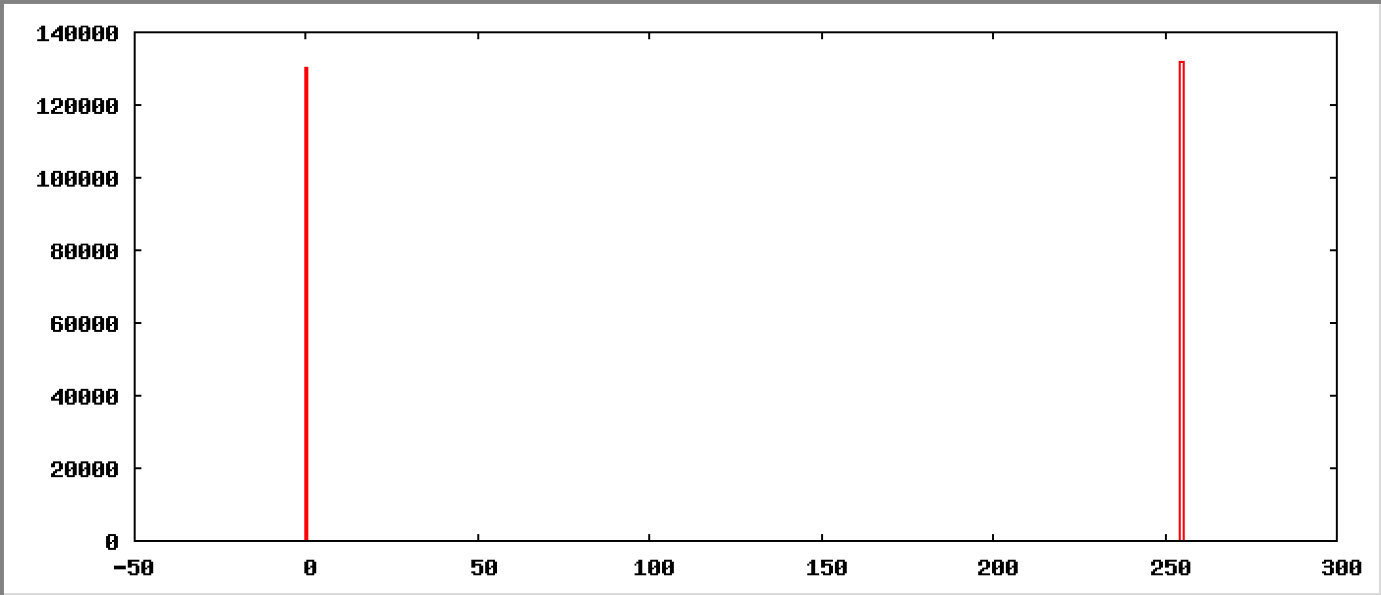
vits nb.vx of=vitsnb.vx





vits mp.vx of=vitsmp.vx





en-ec-ecelinux-10:~/lab3 [12:49am] **169$**vpix -neg mp.vx | vcapt c="vits input mp.vx" of=vitsinputcaptionmp.vx

en-ec-ecelinux-10:~/lab3 [12:50am] **170$**vxport -png vitsinputcaptionmp.vx

vpix -neg vitsmp.vx |vcapt c="vits output mp" of=vitsoutputcaptionmp.vx

en-ec-ecelinux-10:~/lab3 [12:53am] **175$**vxport -png vitsoutputcaptionmp.vx

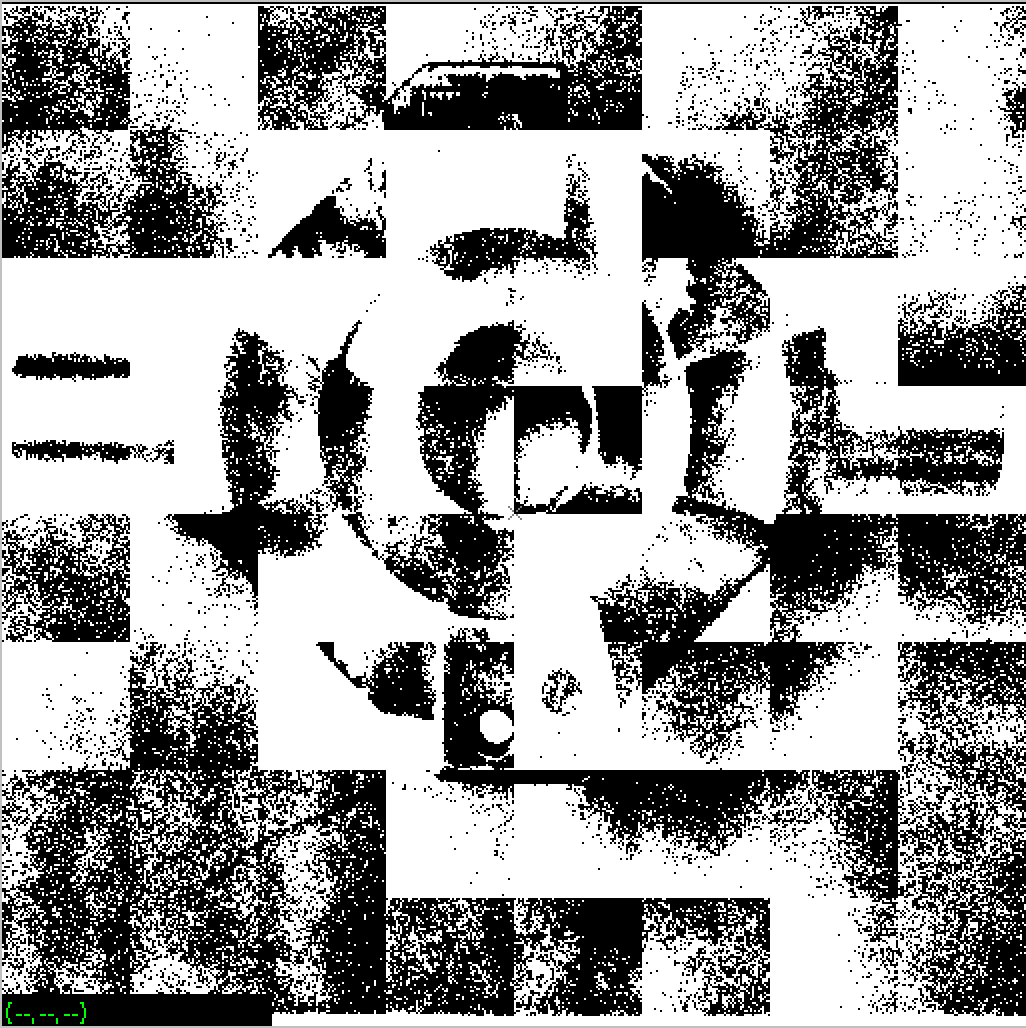
### 4. Adaptive Thresholding

In adaptive thresholding an image may be decomposed into a number of patches for which a separate threshold is computed. The command vpatch decomposes an image into a sequence of overlapping patches. vquiltwith appropriate parameters will reassemble the patches back into an image. For example, consider the command:

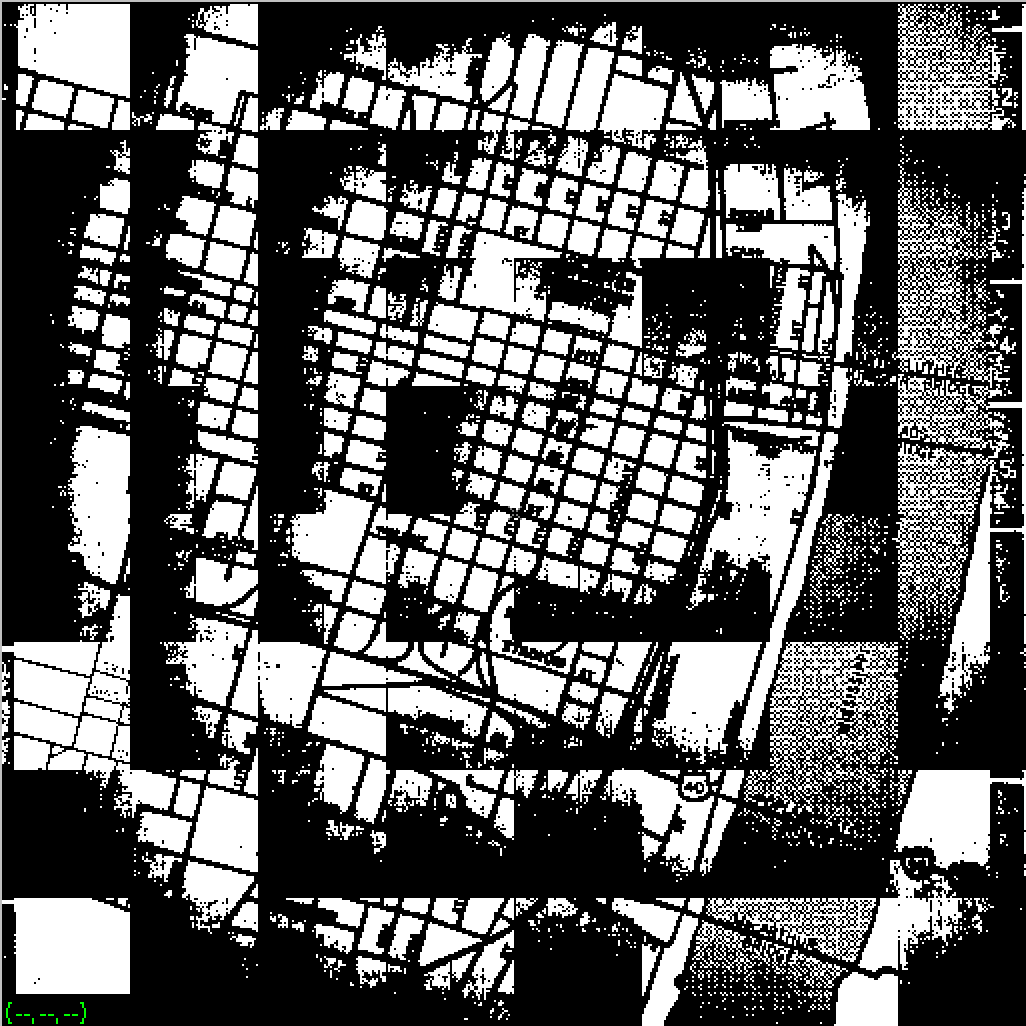
vpatch p=64 l=0 mp.vx | vits | vquilt -p of=vitsp64l0mpout.vx



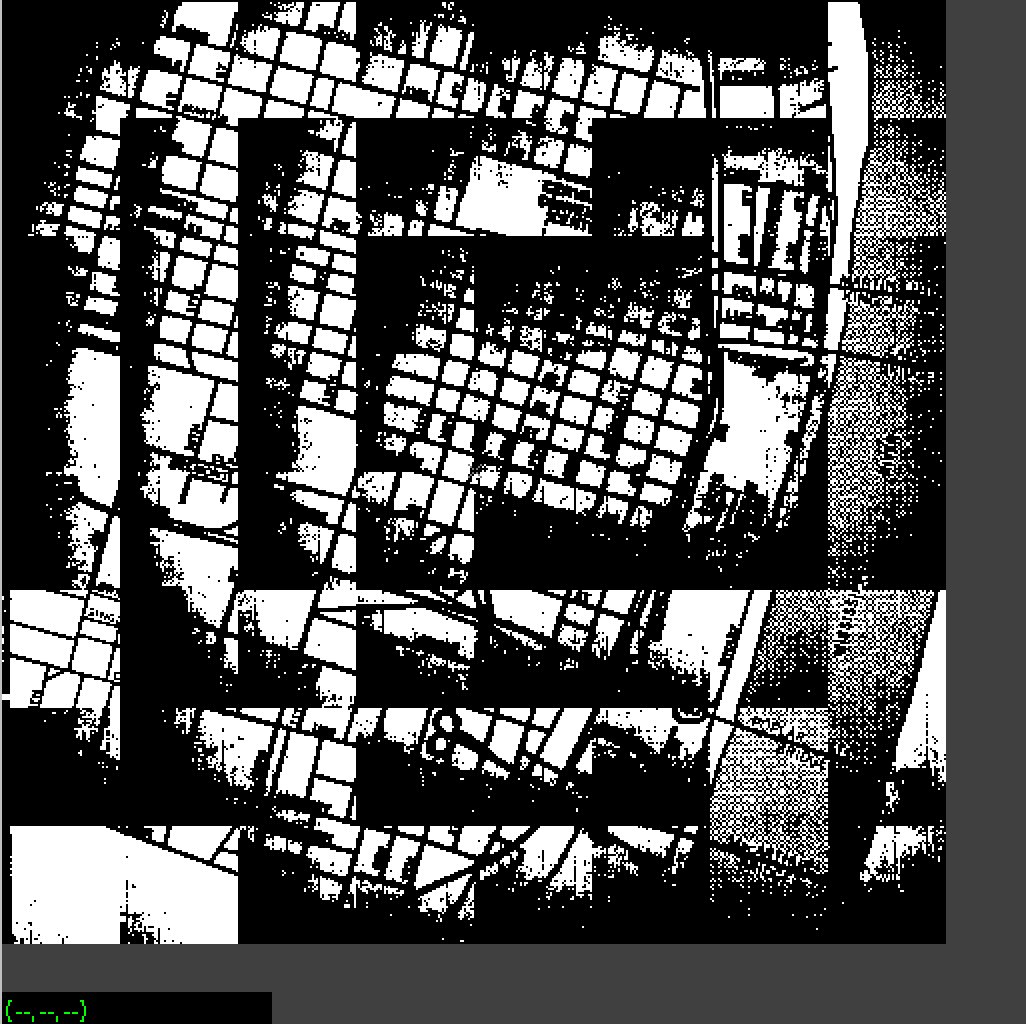
vpatch p=64 l=0 mp.vx | vtpeak | vquilt -p of=vtpeak64l0outmp.vx



vpatch p=64 l=0 map | vtpeak | vquilt -p of=vtpeak64l0outmap.vx



vpatch p=64 l=5 map | vtpeak | vquilt -p of=vtpeak64l5outmap.vx



vpatch p=64 l=10 map | vtpeak | vquilt -p of=vtpeak64l10outmap.vx





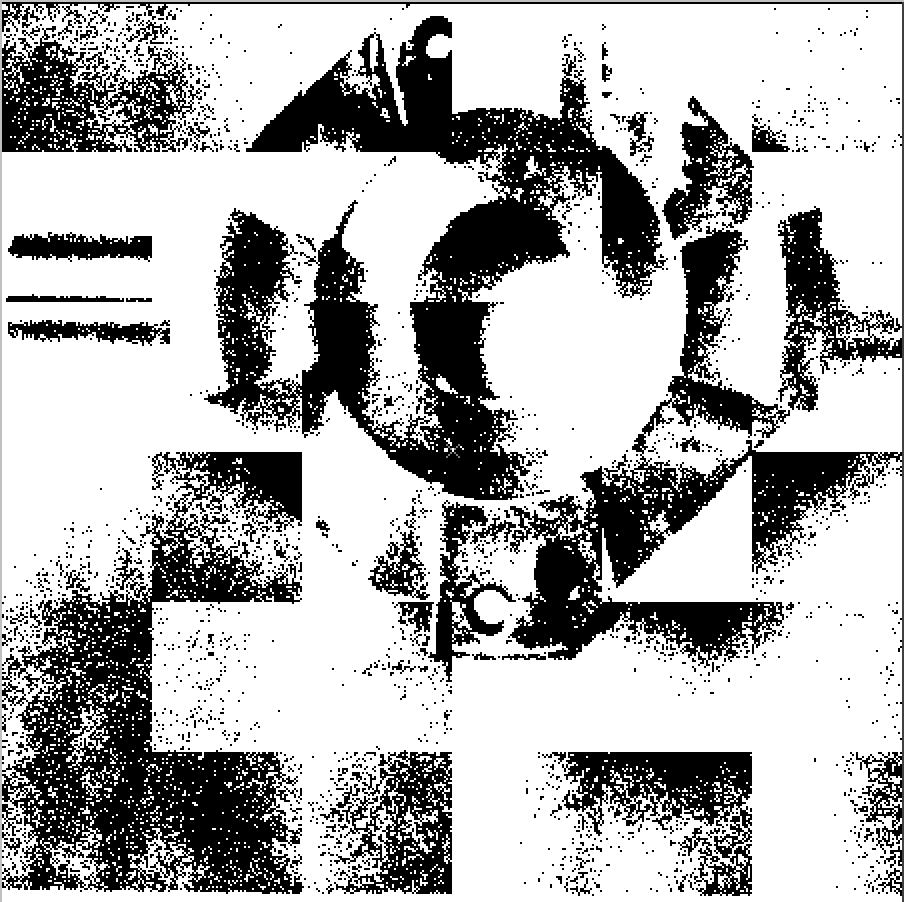
vpatch p=30 l=0 map | vtpeak | vquilt -p of=vtpeak30l0outmap.vx



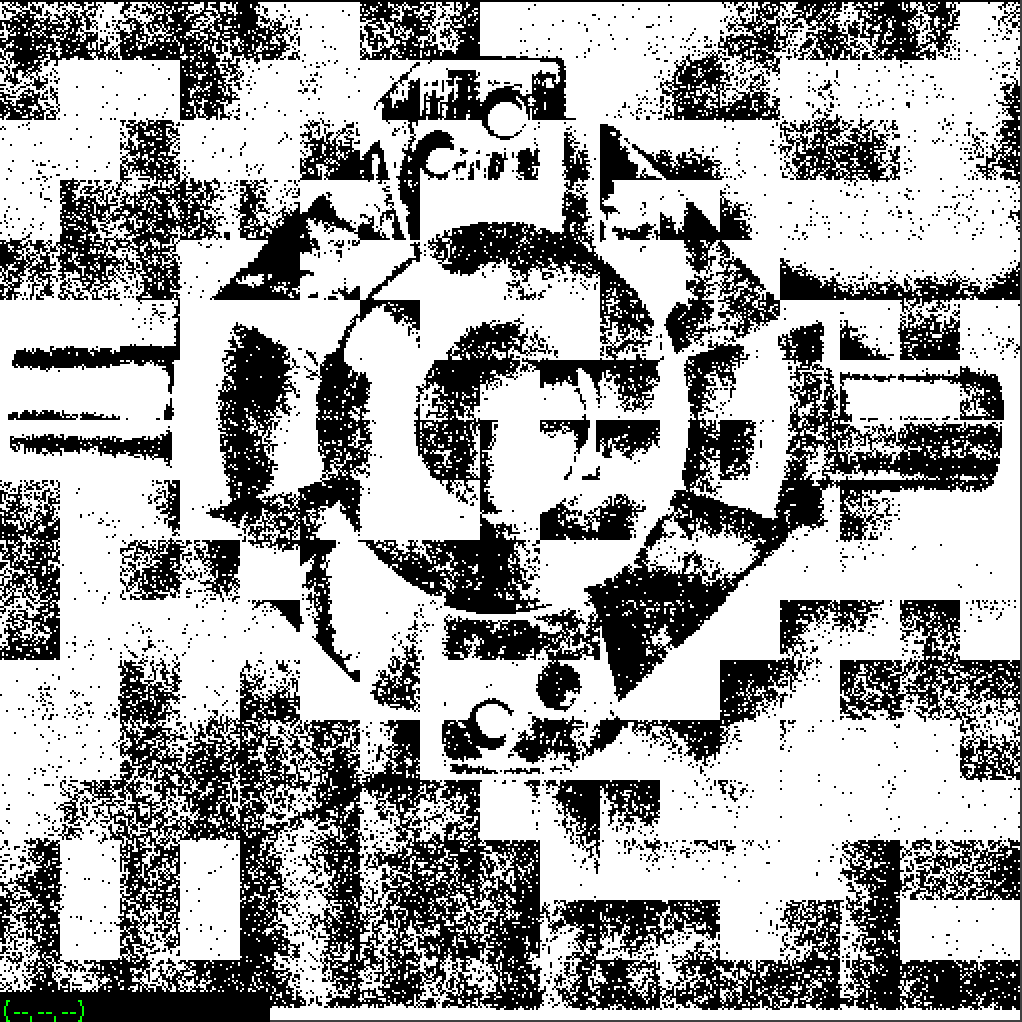
vpatch p=80 l=5 map | vtpeak | vquilt -p of=vtpeak80l5outmap.vx



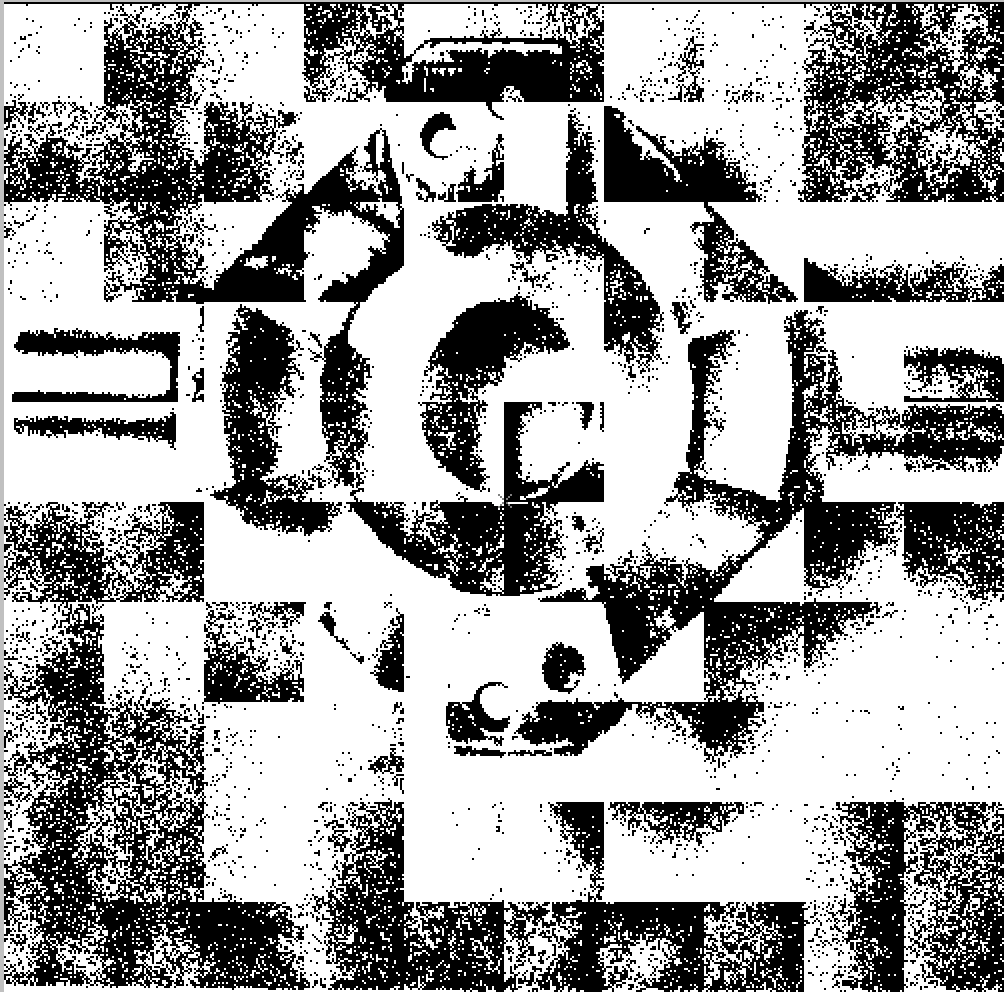
vpatch p=80 l=5 mp.vx | vtpeak | vquilt -p of=vtpeak80l5outmp.vx



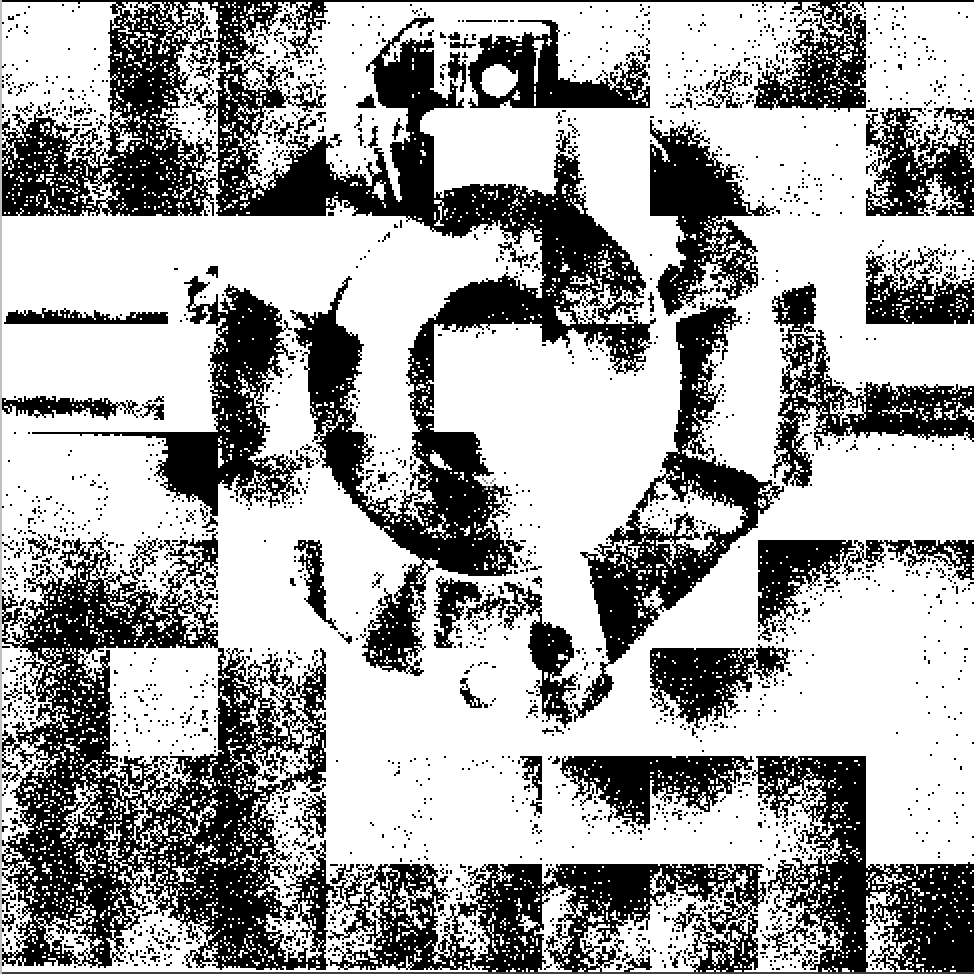
vpatch p=30 l=0 mp.vx | vtpeak | vquilt -p of=vtpeak30l0outmp.vx



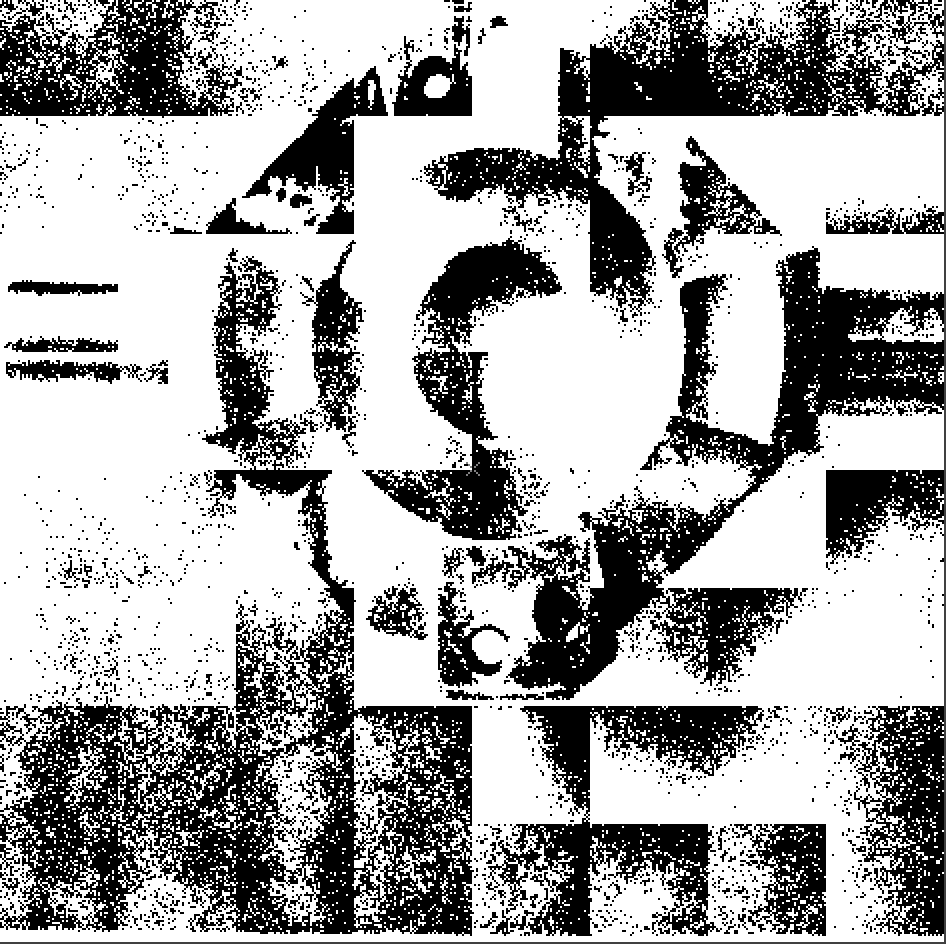
vpatch p=50 l=0 mp.vx | vtpeak | vquilt -p of=vtpeak50l0outmp.vx



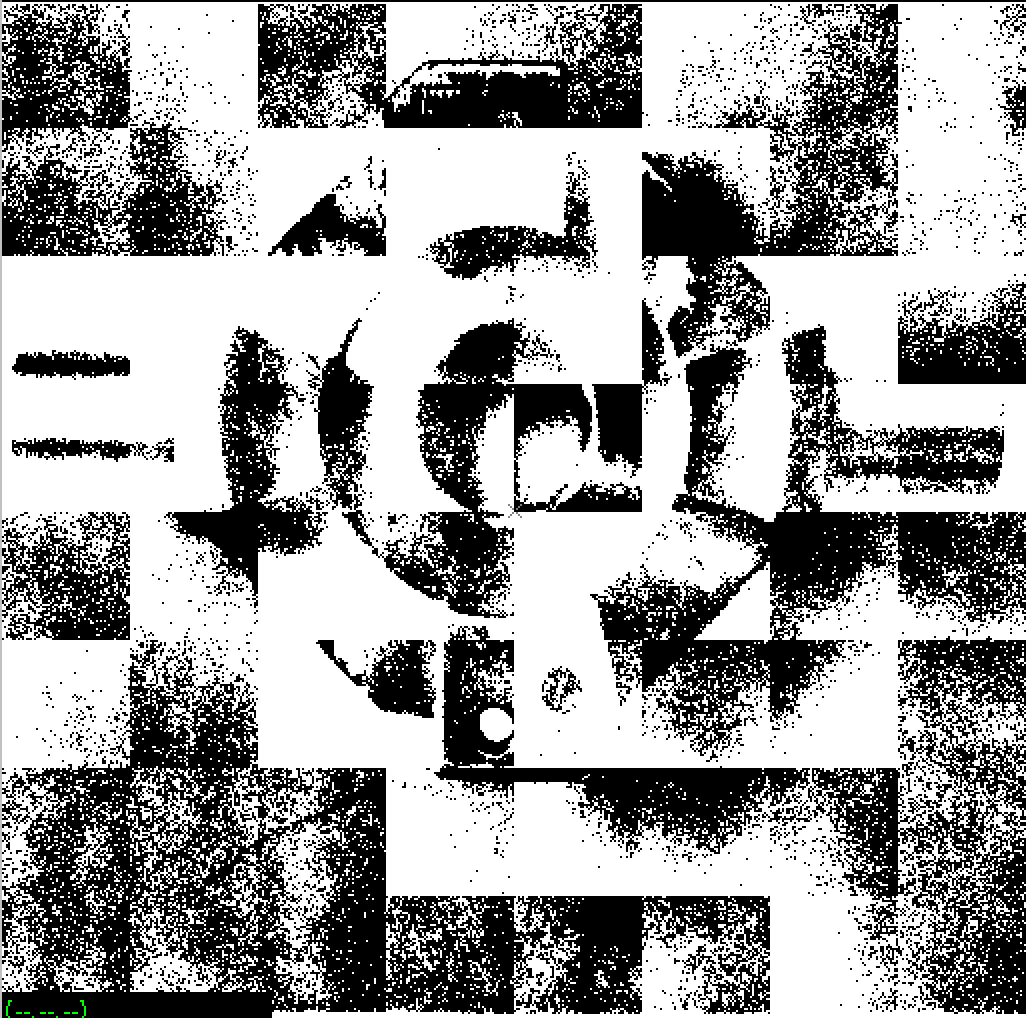
vpatch p=64 l=10 mp.vx | vtpeak | vquilt -p of=vtpeak64l10outmp.vx



vpatch p=64 l=5 mp.vx | vtpeak | vquilt -p of=vtpeak64l5outmp.vx



vpatch p=64 l=0 mp.vx | vtpeak | vquilt -p of=vtpeak64l0outmp.vx



### 5. Region Growing (vgrow)

In cclable.c We had two criteria for calling setlabel on a pixel :

1. the pixel had not yet been labeled
2. the pixel was part of the object (i.e. non-zero)

f we change the second criteria above from simply checking for non-zero pixels to one that checks if the pixel is within a range of values, we can use the algorithm to identify all the connected pixels that are within a given range of values. Optionally, instead of sequentially numbering each region, we can use a pixel value that is representative of the region, (e.g. the value of the first pixel in the region), as the output "label". With the above changes we can effectively "grow regions" of similar pixels. Start by copying cclabel.c to vgrow.c.

The suggested algorithm for the vgrow program is as follows :

1. Add a new input parameter, "r=", to vgrow. This is an integer value that will set the region pixel range. Assign this input value, (or default), to a **global** variable range.
2. Add a new flag, "-p", to vgrow to select the labeling scheme. When this flag is set, the label value will be the value of the first pixel in the region. Otherwise the regions are numbered sequentially.
3. Initialize im and tm as in cclabel. Note that tm and im should be **global** variables.
4. Scan the input image, tm, if the image pixel is not labeled (i.e. the output image, im, is zero for this pixel), then
   1. set the **global** variable first to the pixel value.
   2. if the "-p" flag is set, then call setlabel(j, i, first) else call setlabel with the next sequential label number. Note that you must go back to label number 1 after using label number 255.
5. Repeat step 4 for all unlabeled pixels.

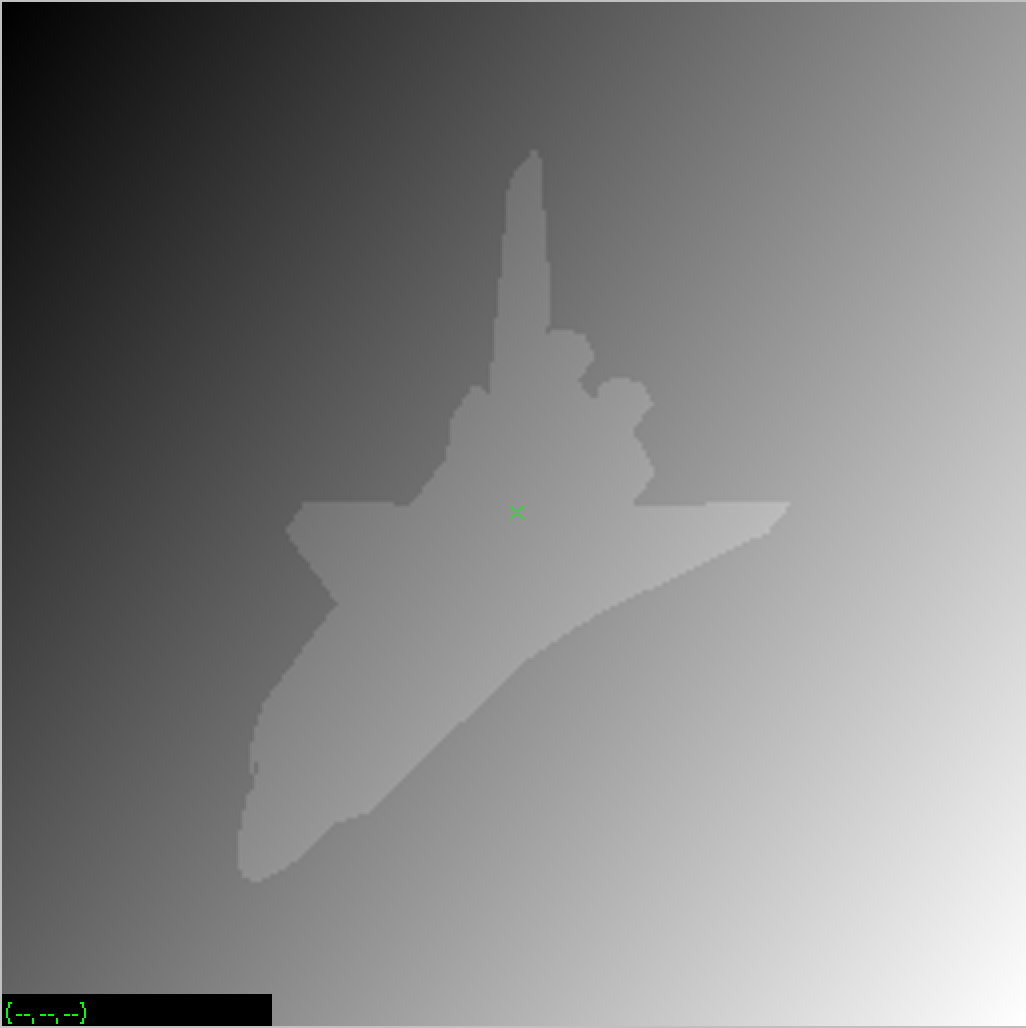
The suggested procedure for recursive function setlabel(x, y, l) is

1. Set the output image pixel at (x, y) to l.
2. For each of the four neighbors of (x, y), check the following three criteria in the order given :
   1. if the image pixel, tm, is non-zero \\ (since we only have zero pixels at the image border, this check prevents us from leaving the bounds of im)
   2. and the pixel is unlabeled
   3. and the pixel is **in range**  
      (**in range** means that the difference between this pixel and the first pixel for this region is less than range.)

then call setlabel for this neighbor.

vcc vgrow.c -o vgrow

vgrow shtl.vx of=vgrowshtl1.vx



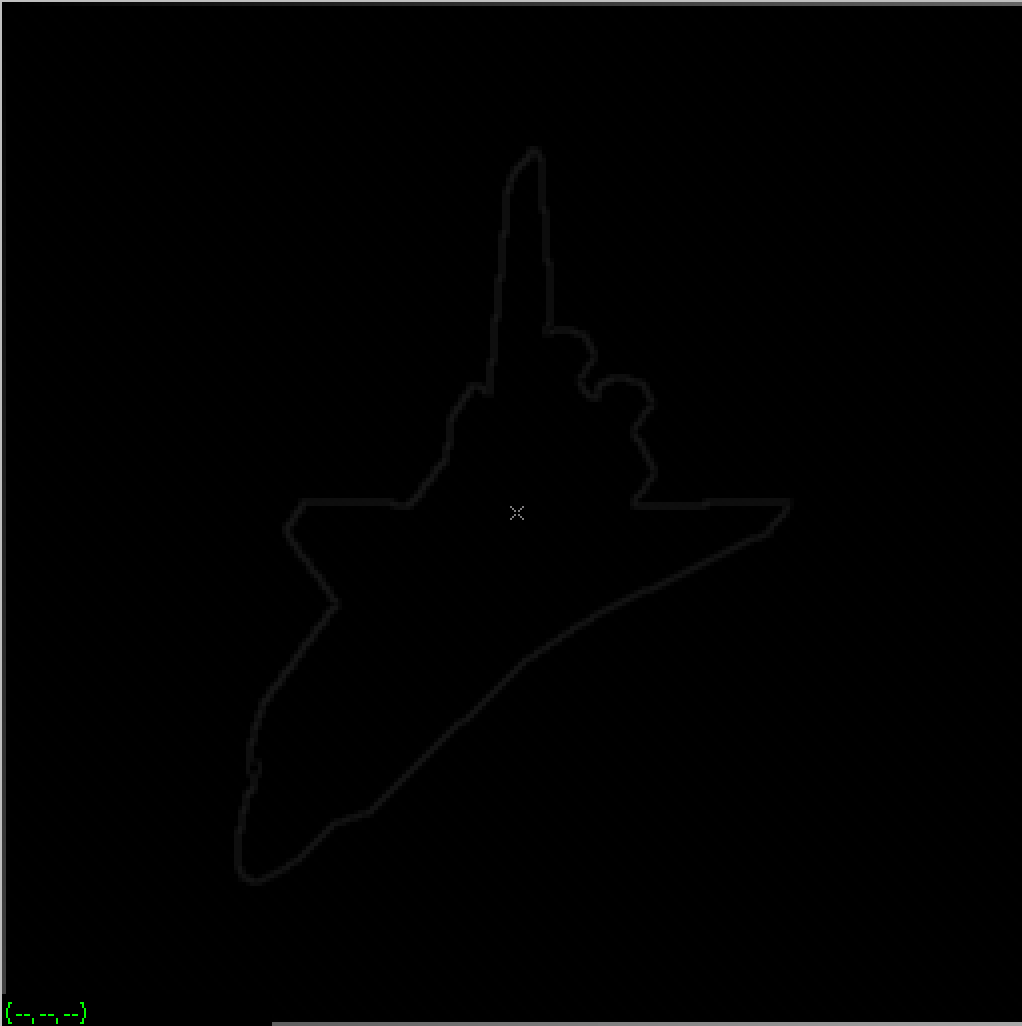
有问题

vsobel shtl.vx of=vsobelshtledge.vx



vgrow vsobelshtledge.vx of=vgrowvsobelshtledge.vx

value = 1；



vgrow nb.vx of=vgrownb1.vx



vgrow map of=vgrowmap1.vx

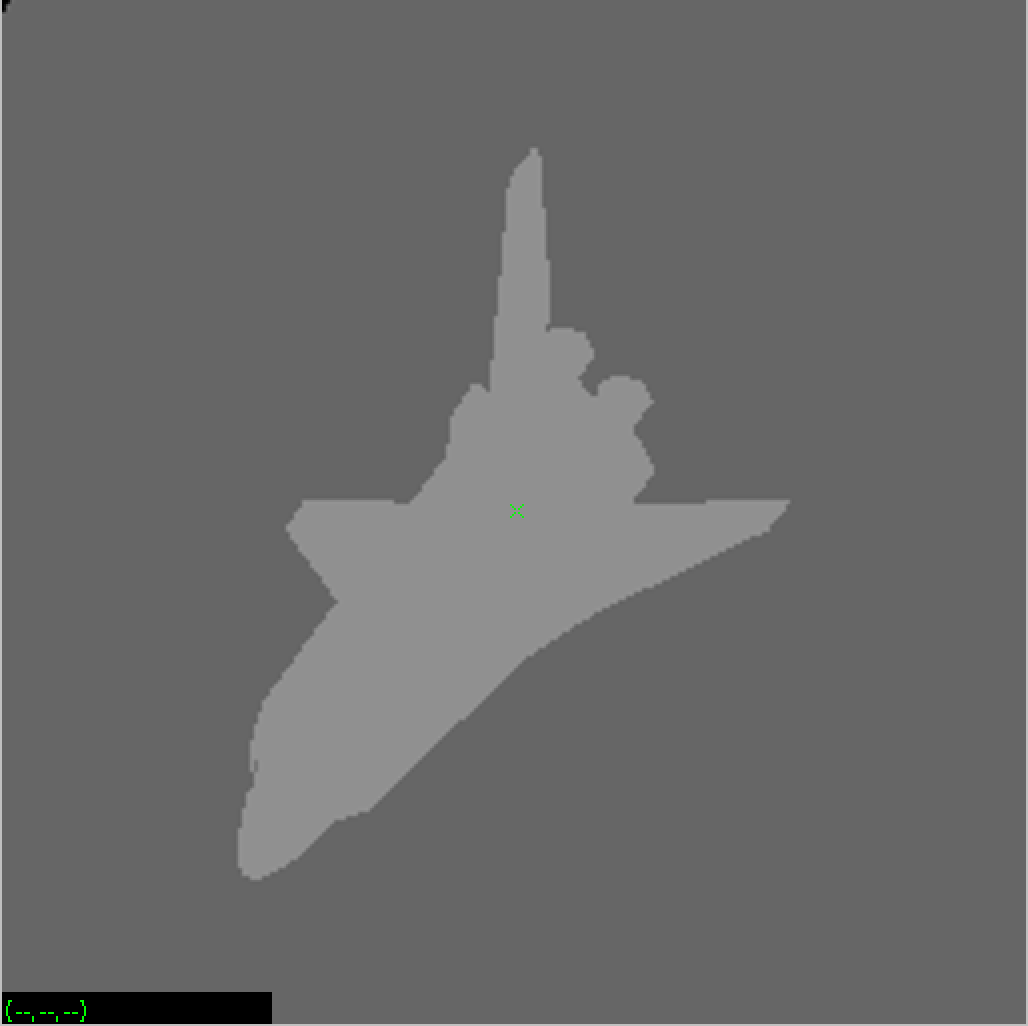


vpatch p=50 l=0 map | vtpeak | vquilt -p of=vtpeak50l0outmap.vx

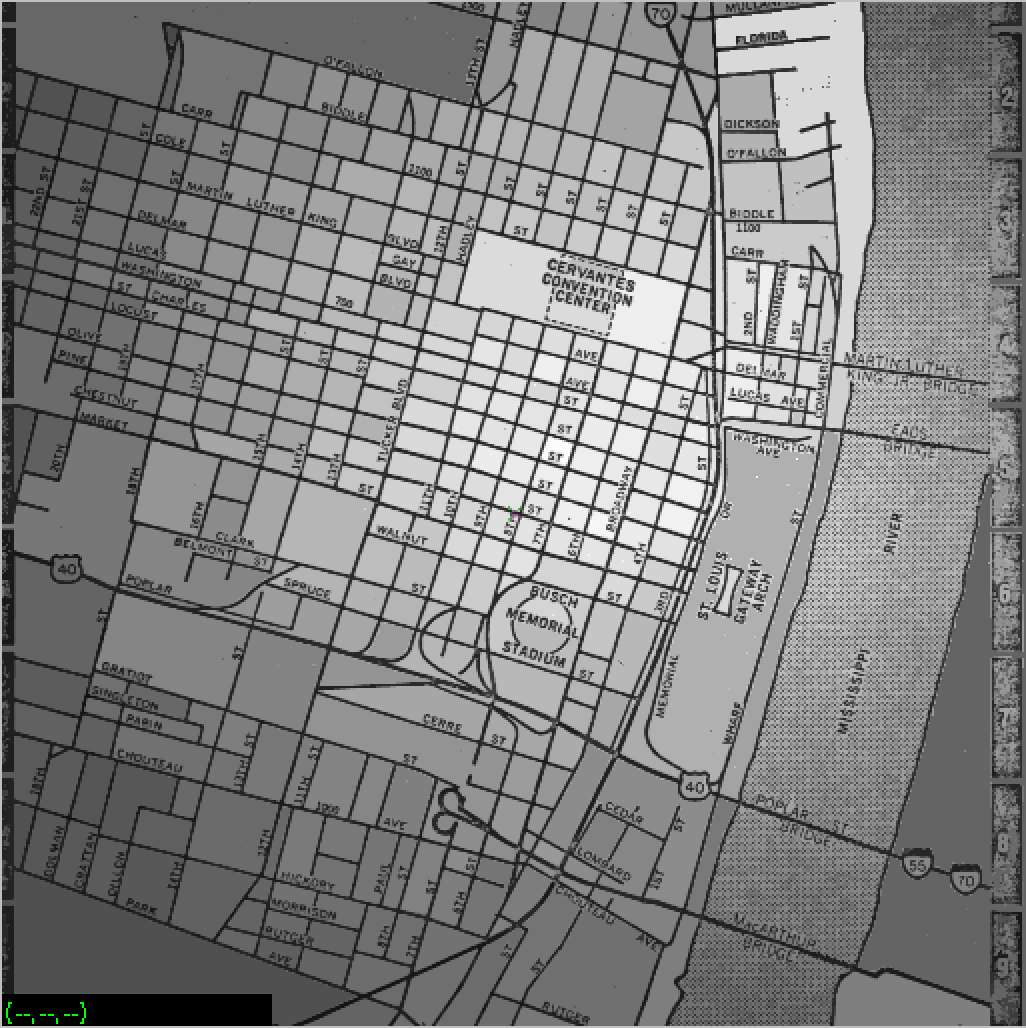
vgrow shtl.vx of=vgrowshtl20.vx



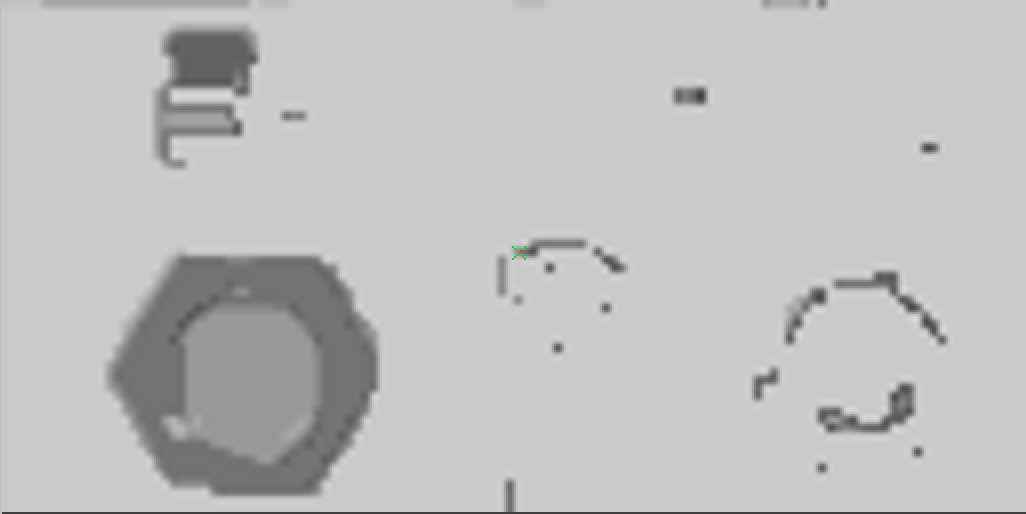
vgrow shtl.vx of=vgrowshtl10.vx



vgrow map of=vgrowmap10.vx



vgrow nb.vx of=vgrownb10.vx



vgrow nb.vx of=vgrownb5.vx



vgrow shtl.vx of=vgrowshtl5.vx

