Appendix A

What follows is an explanation of our *orient* algorithm. We denote the Radon transform of an image, centered at $\left(\frac{R_I}{2}, \frac{C_I}{2}\right)$, at an angle θ, as $\Re_\theta\{I(r,c)\}(\theta,t)$ and the Fourier transform of $R(\theta,t)$ in the t dimension as $\mathbb{F}_t\{R(\theta,t)\}(\theta, f)$. Using the discrete versions of these transforms (Hough and FFT, respectively), our *orient* minimization algorithm calculates:

$$\phi = \left\{\arg\max_\theta \left[\sigma_f(\theta)\right]\right\} = \left\{\arg\min_\theta \left[-\sigma_f(\theta)\right]\right\} \qquad 0 \le \theta \le 180^o \tag{1}$$

Where:

$$\begin{aligned} R(\theta,t) &= \Re_\theta\{I(x,y)\} \approx \mathrm{H}_\theta\{I(r,c)\} \\ \hat{R}(\theta, f) &= \left|\mathbb{F}_t\{R(\theta,t)\}\right| \approx \left|FFT_t\{R(\theta,t)\}\right| \\ \mu_f(\theta) &\approx \frac{1}{N}\sum_{f=0..N} \hat{R}(\theta, f) \\ \sigma_f(\theta) &\approx \sqrt{\frac{1}{N-1}\sum_{f=0..N}\left(\hat{R}(\theta, f) - \mu_f(\theta)\right)^2} \end{aligned} \tag{2}$$

Note that while we do not prove that maximizing $\sigma_f(\theta)$ gives an optimal alignment, it has performed as well as or better than the regionprops Orientation property in our experiments. We describe the logic behind our algorithm below.

Consider fastener 6 from our preliminary data set:

Figure 1 – Fastener 6, before rotation

Below are the two plots $R(\theta_{long}, t)$ and $R(\theta_{short}, t)$. Here θ_{long} (the green line) is the set of projections along the long axis and θ_{short} is 90 degrees perpendicular. Note θ_{long} in this plot is 90 degrees perpendicular to the long axis of the fastener itself, so θ_{short} gives us the optimal orientation.

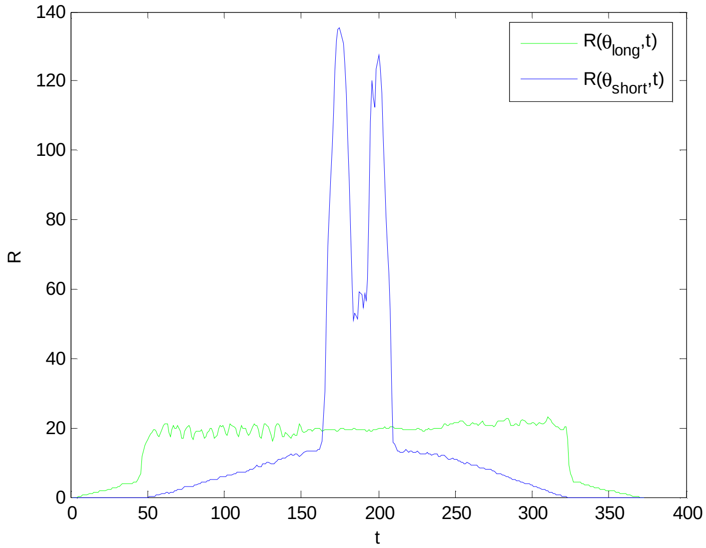


Figure 2 – Two Radon projections of fastener 6

1. The Radon transform of an image has the same support area (and mean value) from all angles. This implies that the DC frequency is the same in the FFT of the Radon transform at any angle.
2. If we apply the Radon transform along the short axis, we expect to see oscillatory behavior with "higher amplitudes" than from any other axis. Note this behavior in the blue line in Figure 2, especially as it reaches a peak. This is also apparent from the longer slopes along the side lobes of the blue line.

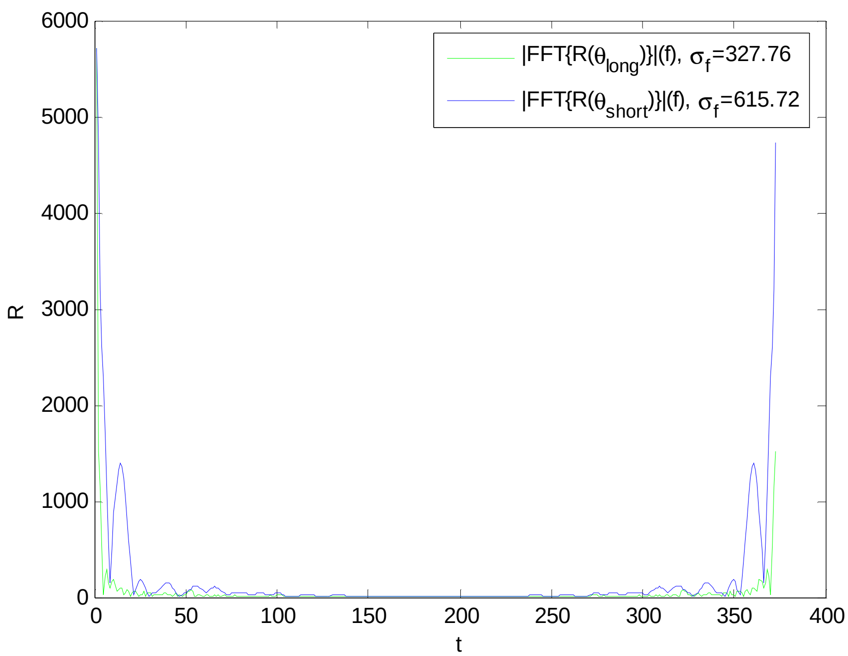


Figure 3 – FFT of Radon Projections of fastener 6

3. These amplitudes of highly oscillatory behavior are apparent in the FFT (Figure 3), as side lobes at high frequencies. One easy way of comparing the frequency side lobes of two functions with the same DC amplitude is by taking the standard deviation of the FFT. Higher standard deviations imply more side lobes with higher amplitudes.

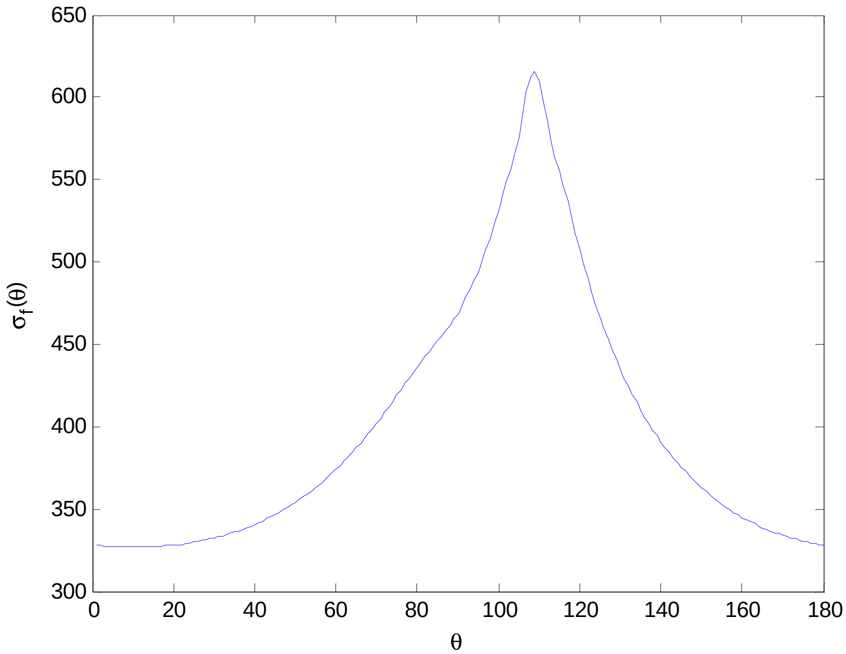


Figure 4 – Function to maximize over theta

Thus we expect ϕ to be the positive angle offset of the fastener *x'* (minor) axis from our *x* axis.

Finally, we calculate the means of $R(\theta_{long}, 0 \le t \le N/2)$ and $R(\theta_{long}, N/2 \le t \le N)$. The term with the higher mean is more likely to be the head, as the head tends to be longer in the perpendicular direction: its projection will have a larger value. If the first term has a higher mean, we flip the image by 180^o.

Appendix B

We have assumed that the contours $x_L(y)$ and $x_R(y)$ are continuous functions. Unfortunately, we only have a sample set of points for these functions. Furthermore, the sampling is *non-uniform*. In other words., we may have $x_L(y)$ for *y=.01,.015,.018,.02,.021,…*. Taking very simple finite differences will not always provide a good approximation of the first and second derivatives. Instead, we derived the finite difference formulas for non-uniformly sampled functions. Specifically, for a function *f(x)* sampled at points $f(x), f(x+h_1), f(x-h_2)$, we can calculate:

$$f'(x) = \frac{f(x+h_1)-f(x-h_2)}{h_1+h_2} - \frac{1}{2!}\left(\frac{h_1^2-h_2^2}{h_1+h_2}\right) f''(x) + \ldots \qquad (1)$$

The error of this approximation can also be calculated and bounded:

$$e(f;h_1,h_2) = O\left(\frac{h_1^2-h_2^2}{h_1+h_2}\right) = O\left(h_1-h_2\right) \qquad (2)$$

Note that this is a first order approximation, and as h_1 approaches h_2, this formula becomes the standard second order central difference method.

A similar derivation for the second derivative given 4 points around *f(x)*, in this order: $f(x-h_{22})$, $f(x-h_2)$, $f(x)$, $f(x+h_1)$, $f(x+h_{11})$, can be calculated as follows:

$$\begin{aligned} \alpha &= 2\left[f(x+h_{11}) + f(x-h_{22}) + \frac{f(x-h_2)-f(x+h_1)}{\left(h_1+h_2\right)}\left(h_{11}-h_{22}\right) - 2f(x) \right] \\ f''(x) &\simeq \frac{\alpha}{h_{11}^2+h_{22}^2} \end{aligned} \qquad (3)$$

This second formula provides the foundations of our *findcurv2* function.

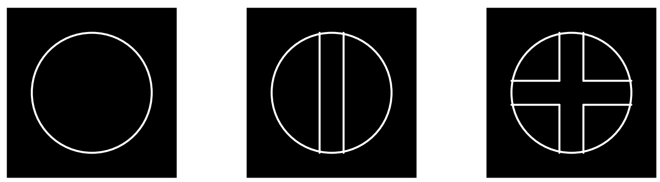
Appendix C

Here we propose an algorithm for differentiating the Fastener Head Type from among multiple possible types using head models and optimizing cross-correlations. This algorithm uses the entire head image, as opposed to just its contour.

Input:

G: Gradient of the head image
r: approximate radius of the head
c: approximate centroid of the head

Algorithm:
Generate 3 black and white images: I_b, I_f, I_p, respectively:



I_b $\qquad$ I_f $\qquad$ I_p

where each of the circles has the radius r. These images are models of the head gradient we might expect to see in ideal conditions.

Denote R_ϕ as the rotation operator on ϕ degrees, $\kappa_{m,n}[I_1, I_2]$ as the cross-correlation operator in 2 dimensions, with $\kappa_{m,n}[I_1, I_2] = \sum_{j=1} \sum_{i=1} I_1(i, j) I_2^*(i+m, j+n)$, $K[I_1, I_2] = \sum_n \sum_m \kappa_{m,n}[I_1, I_2]^2$ as the squared cross correlation norm.

Calculate:

$$K_1 = \max_\phi \left\{ K[\| G \|, R_\phi I_f] \right\}$$

$$K_2 = \max_\phi \left\{ K[\| G \|, R_\phi I_p] \right\}$$

$$K_3 = K[\| G \|, R_\phi I_b]$$

Output:

The head type with the highest cross-correlational norm (e.g., I_b, I_f, or I_p)

Appendix D

The list structure of the FASTENER_DATA object.

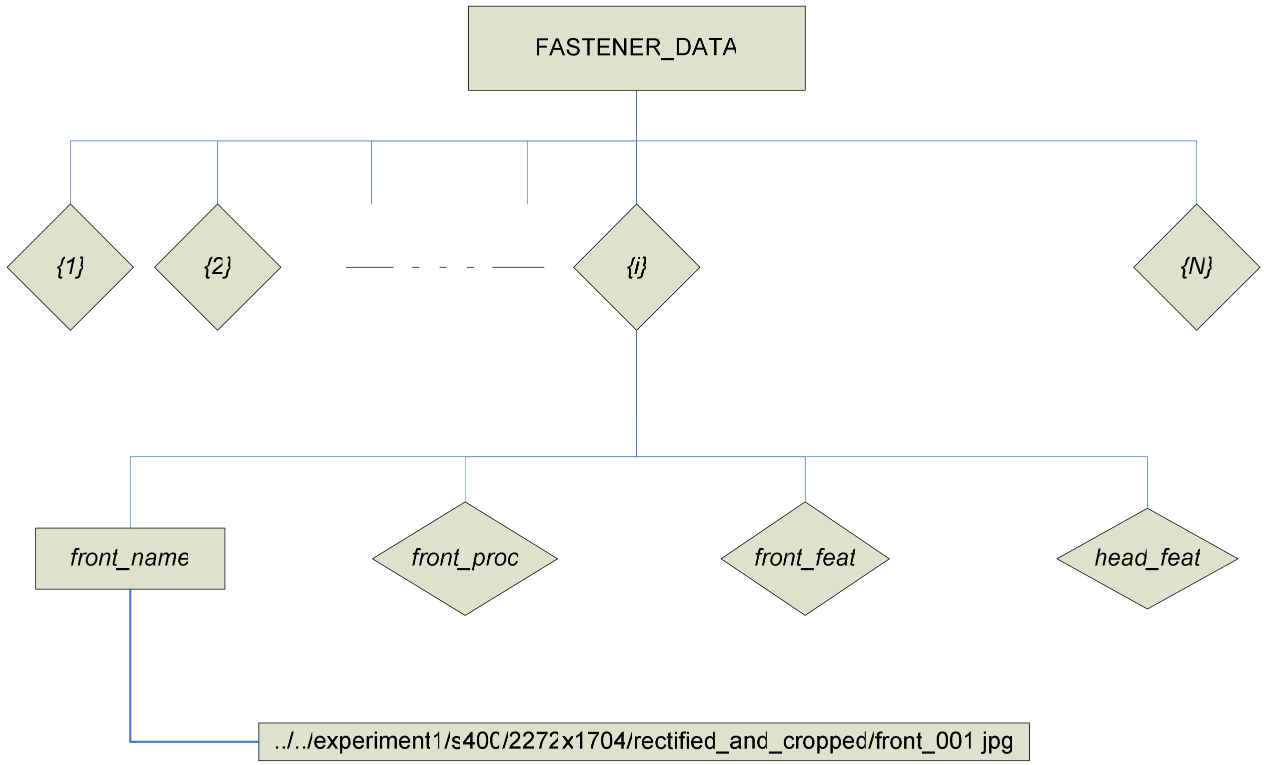


Figure 1 – Top level view of FASTENER_DATA list and underlying structures

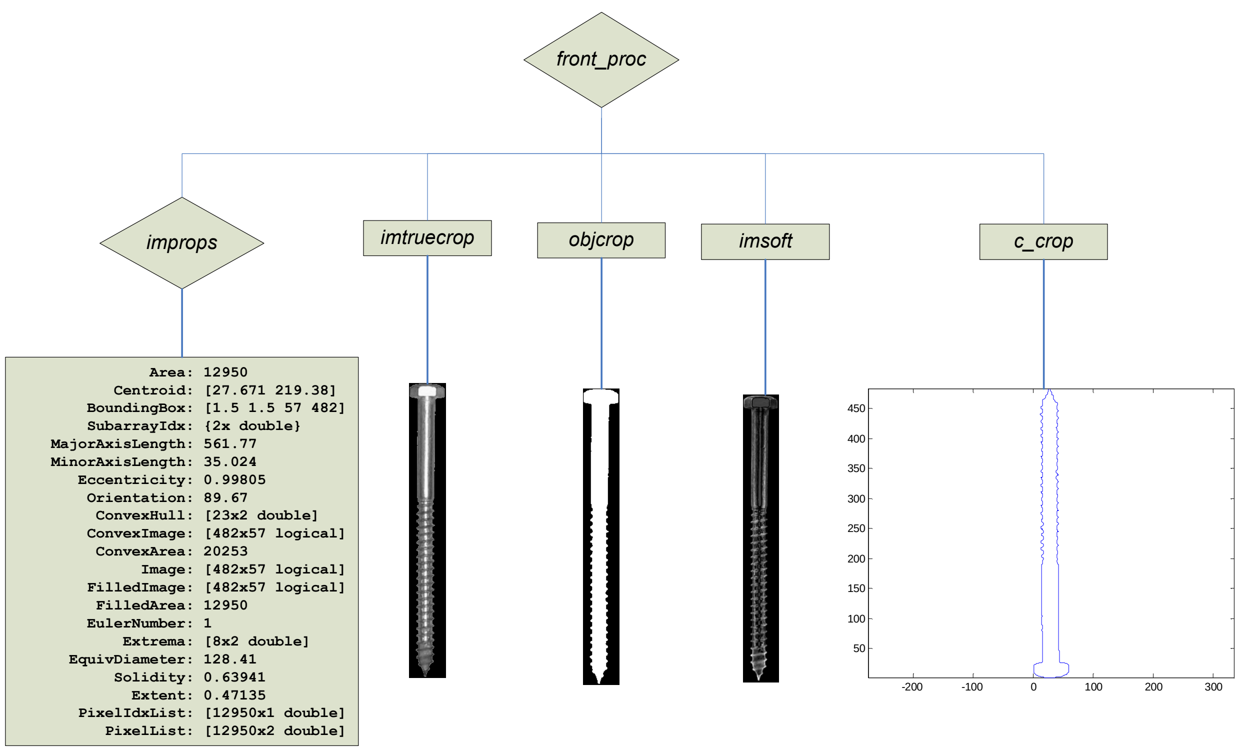


Figure 2 – Sample *fasteners{k}* (frontal post processing) object as found in FASTENER_DATA

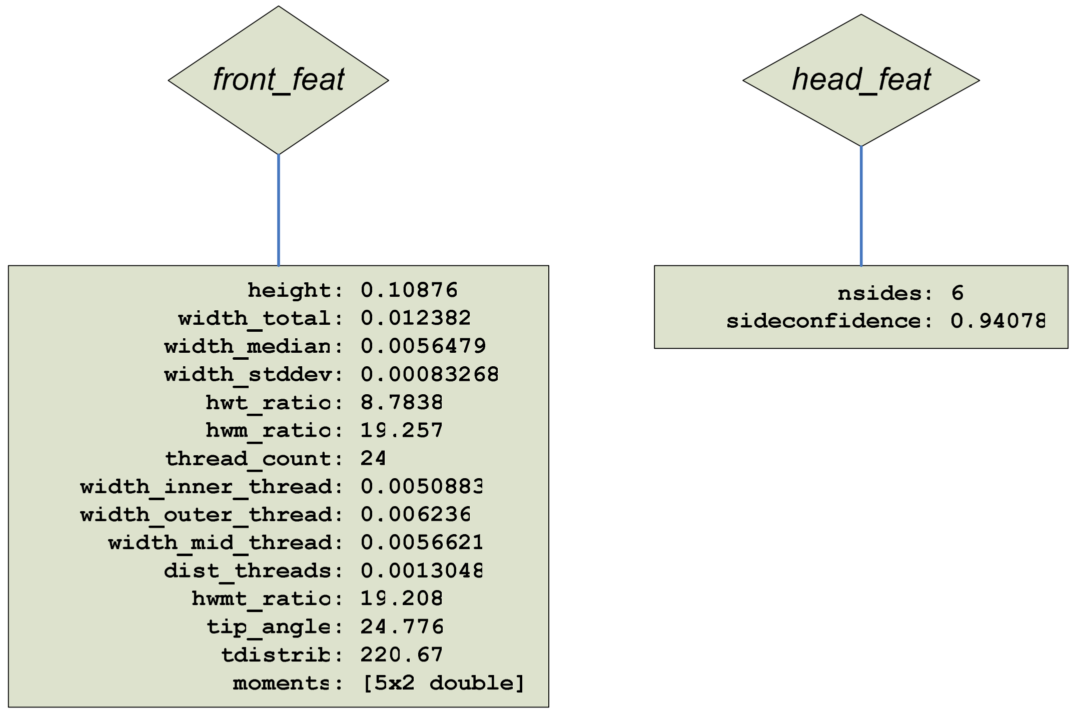
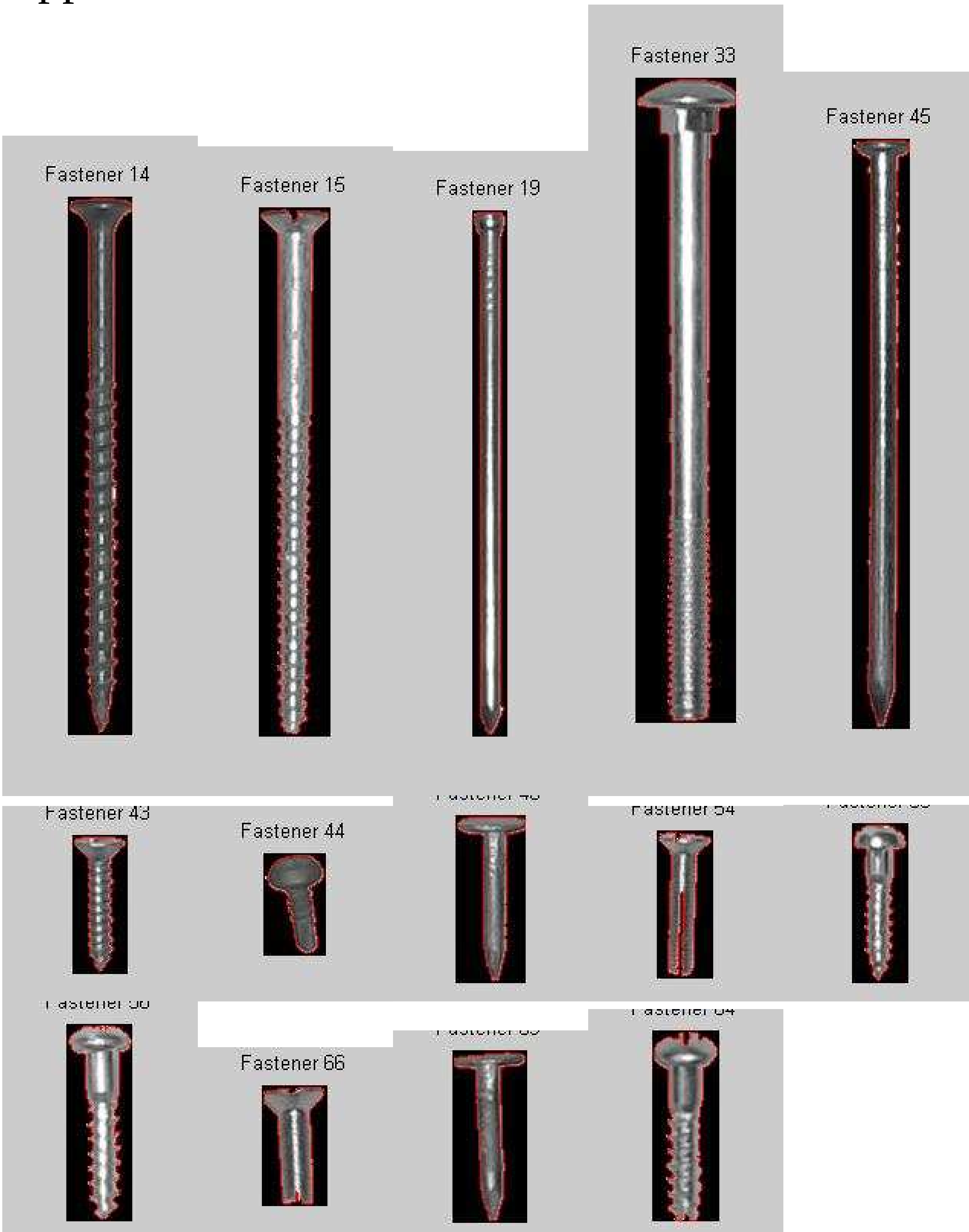
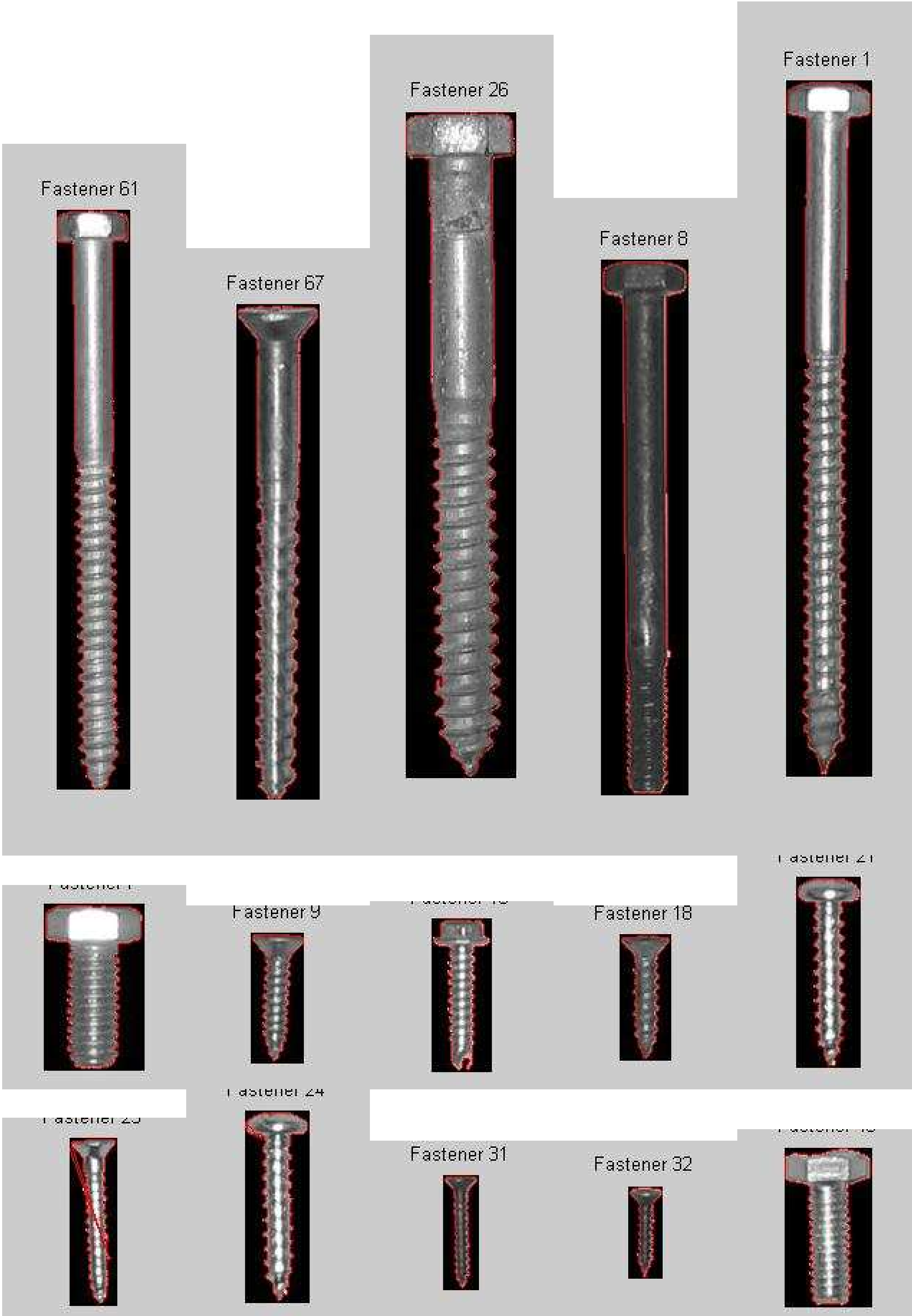


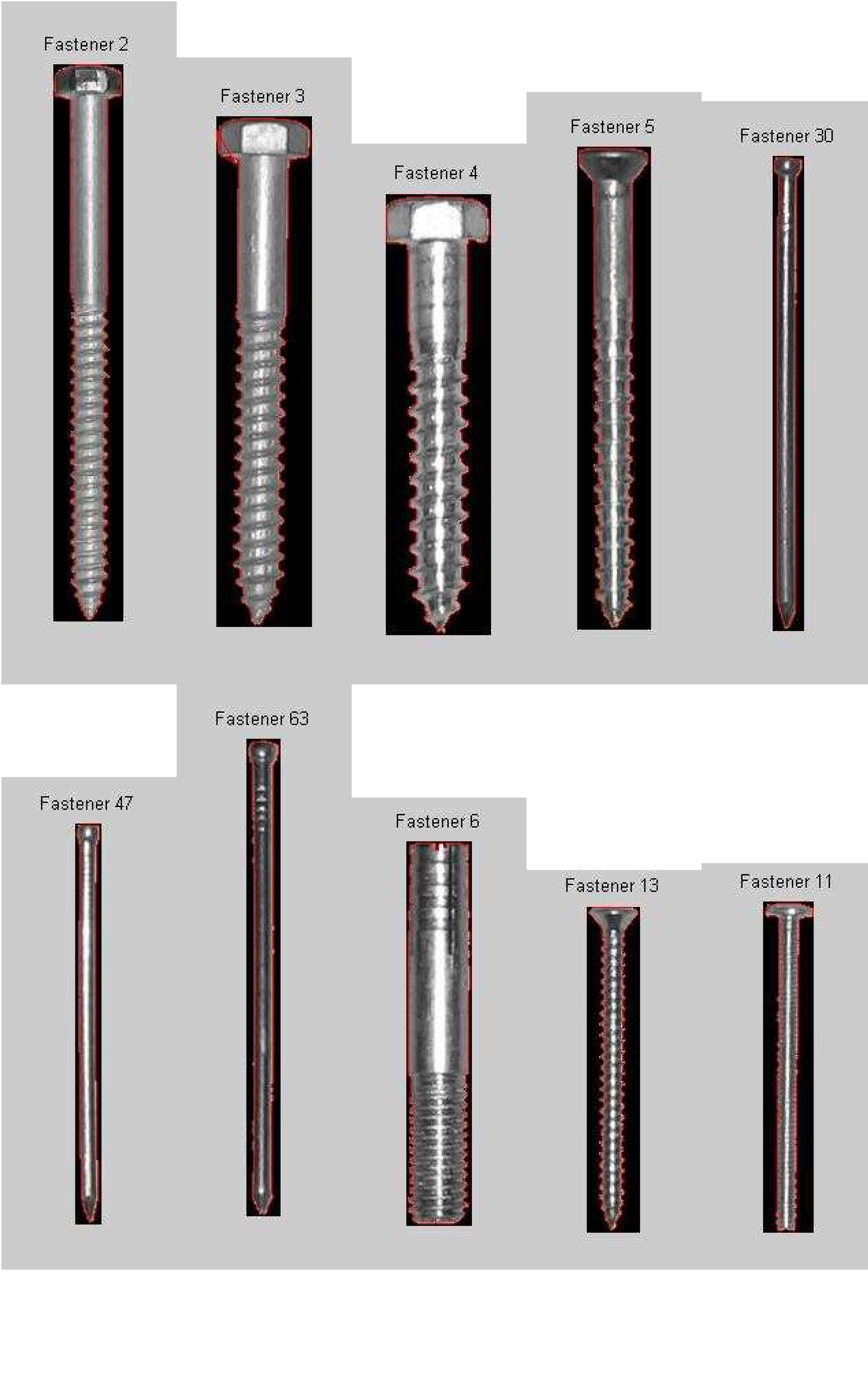
Figure 3 – Sample *front_features* and *head_features* object data as found in FASTENER_DATA

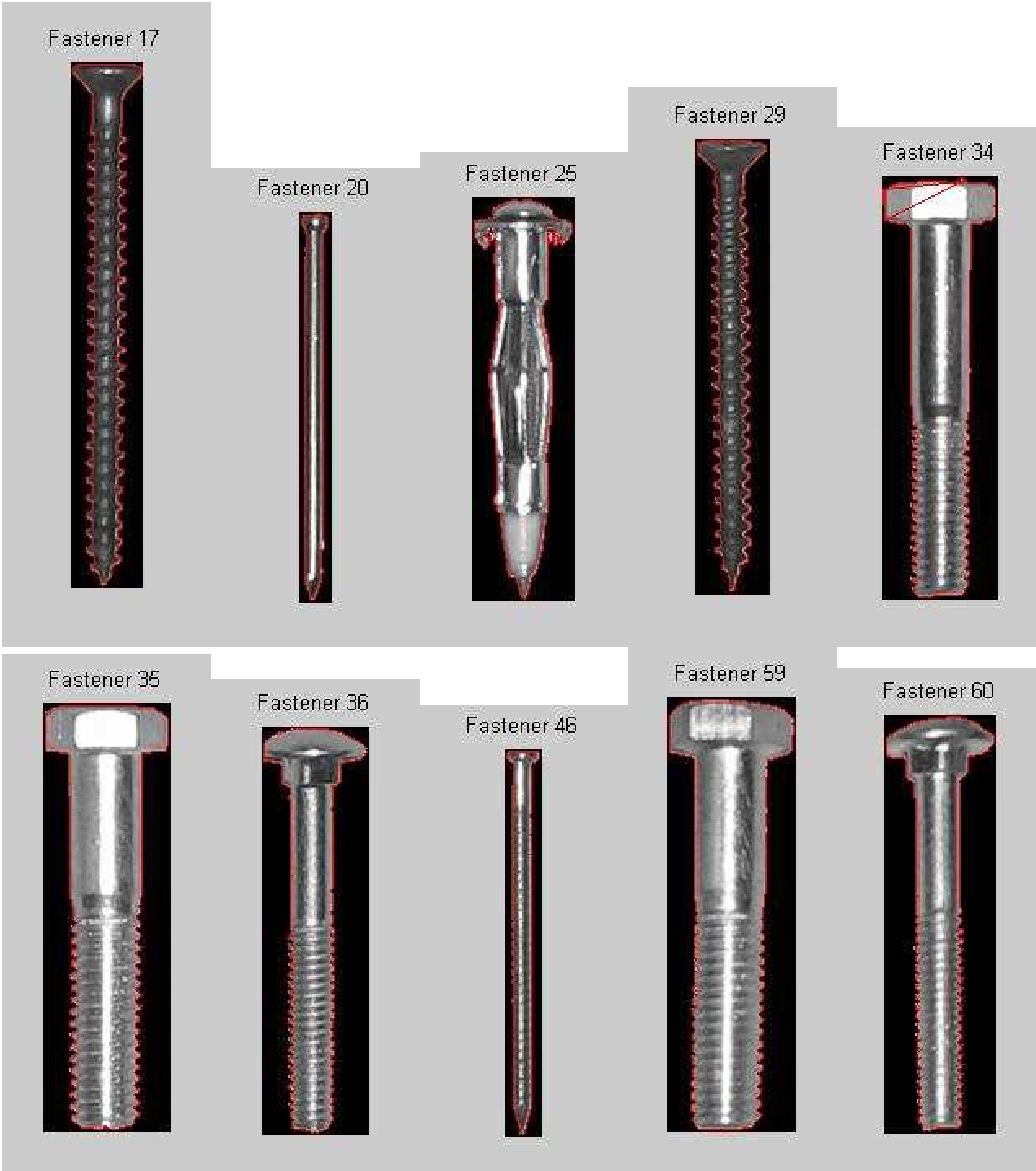
Appendix E

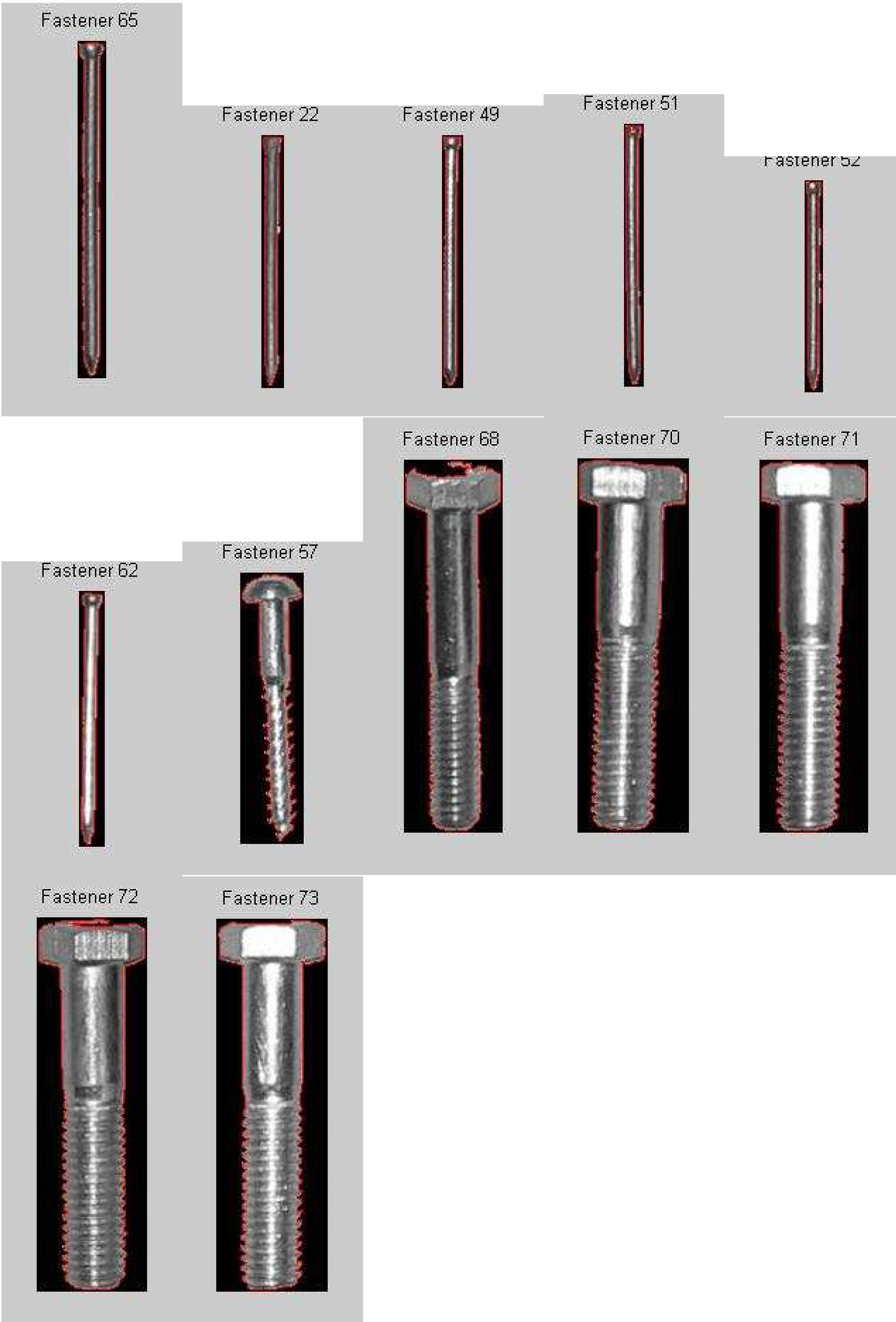


Fastener 33
Fastener 45
Fastener 14
Fastener 15
Fastener 19
Fastener 43
Fastener 44
Fastener 66


Fastener 61
Fastener 67
Fastener 26
Fastener 8
Fastener 1
Fastener 9
Fastener 18
Fastener 31
Fastener 32


Fastener 2
Fastener 3
Fastener 4
Fastener 5
Fastener 30
Fastener 47
Fastener 63
Fastener 6
Fastener 13
Fastener 11


Fastener 17
Fastener 20
Fastener 25
Fastener 29
Fastener 34
Fastener 35
Fastener 36
Fastener 46
Fastener 59
Fastener 60


Fastener 65
Fastener 22
Fastener 49
Fastener 51
Fastener 52
Fastener 68
Fastener 70
Fastener 71
Fastener 62
Fastener 57
Fastener 72
Fastener 73

Appendix F

Thresholding on Well-Defined Image

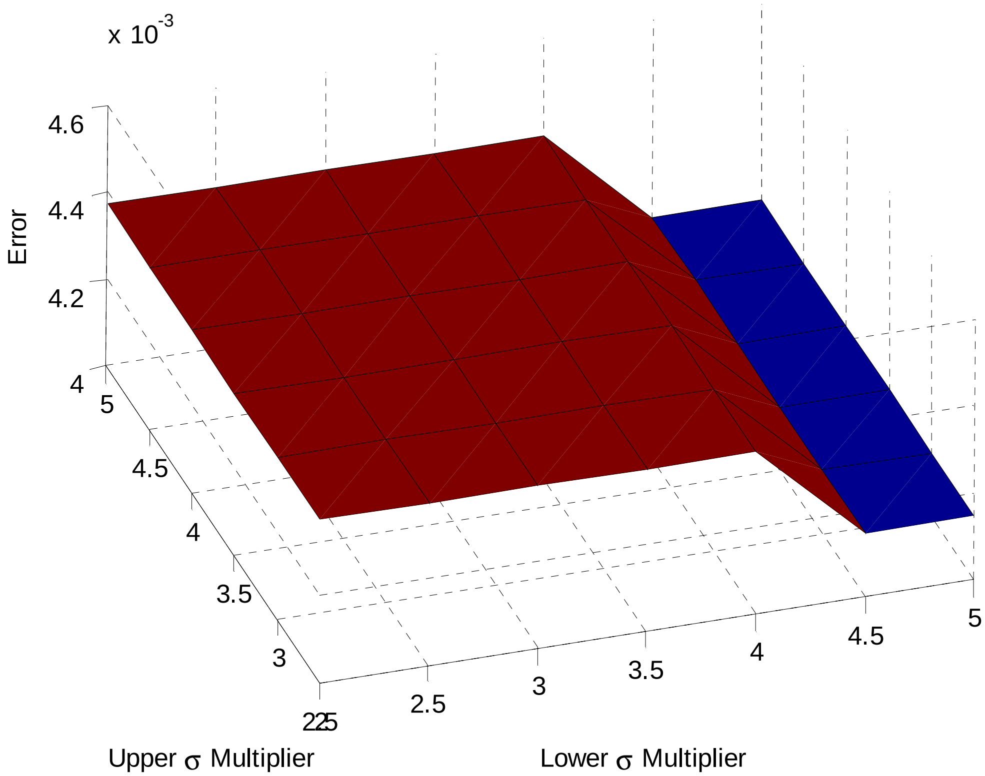
lower	upper	height	width_outer	thread count	height_err	width_err	thread_err
2	2.5	0.121848	0.007938	23	0.004373	5E-07	0.5
2	3	0.121848	0.007939	23	0.004373	1.5E-06	0.5
2	3.5	0.121848	0.007838	21	0.004373	9.95E-05	1.5
2	4	0.121848	0.007838	21	0.004373	9.95E-05	1.5
2	4.5	0.121848	0.007838	21	0.004373	9.95E-05	1.5
2	5	0.121848	0.007838	21	0.004373	9.95E-05	1.5
2.5	2.5	0.121848	0.007933	22	0.004373	4.5E-06	0.5
2.5	3	0.121848	0.007929	22	0.004373	8.5E-06	0.5
2.5	3.5	0.121848	0.007811	22	0.004373	0.000127	0.5
2.5	4	0.121848	0.007811	22	0.004373	0.000127	0.5
2.5	4.5	0.121848	0.007811	22	0.004373	0.000127	0.5
2.5	5	0.121848	0.007811	22	0.004373	0.000127	0.5
3	2.5	0.121848	0.007898	22.5	0.004373	3.95E-05	0
3	3	0.121848	0.007928	23.5	0.004373	9.5E-06	1
3	3.5	0.121848	0.007769	22	0.004373	0.000169	0.5
3	4	0.121848	0.007769	22	0.004373	0.000169	0.5
3	4.5	0.121848	0.007769	22	0.004373	0.000169	0.5
3	5	0.121848	0.007769	22	0.004373	0.000169	0.5
3.5	2.5	0.121848	0.007898	22.5	0.004373	3.95E-05	0
3.5	3	0.121848	0.007928	23.5	0.004373	9.5E-06	1
3.5	3.5	0.121848	0.007769	22	0.004373	0.000169	0.5
3.5	4	0.121848	0.007769	22	0.004373	0.000169	0.5
3.5	4.5	0.121848	0.007769	22	0.004373	0.000169	0.5
3.5	5	0.121848	0.007769	22	0.004373	0.000169	0.5
4	2.5	0.121848	0.007892	22	0.004373	4.55E-05	0.5
4	3	0.121848	0.00776	22	0.004373	0.000178	0.5
4	3.5	0.121848	0.007705	22	0.004373	0.000233	0.5
4	4	0.121848	0.007705	22	0.004373	0.000233	0.5
4	4.5	0.121848	0.007705	22	0.004373	0.000233	0.5
4	5	0.121848	0.007705	22	0.004373	0.000233	0.5
4.5	2.5	0.121622	0.0077798	21	0.004147	0.000158	1.5
4.5	3	0.121622	0.0077236	21	0.004147	0.000214	1.5
4.5	3.5	0.121622	0.0076495	21	0.004147	0.000288	1.5
4.5	4	0.121622	0.0076495	21	0.004147	0.000288	1.5
4.5	4.5	0.121622	0.0076495	21	0.004147	0.000288	1.5
4.5	5	0.121622	0.0076495	21	0.004147	0.000288	1.5
5	2.5	0.121622	0.00764	21	0.004147	0.000298	1.5
5	3	0.121622	0.00764	21	0.004147	0.000298	1.5
5	3.5	0.121622	0.0076157	21	0.004147	0.000322	1.5
5	4	0.121622	0.0076157	21	0.004147	0.000322	1.5
5	4.5	0.121622	0.0076157	21	0.004147	0.000322	1.5
5	5	0.121622	0.0076157	21	0.004147	0.000322	1.5

Thresholding on a Poorly Defined Image

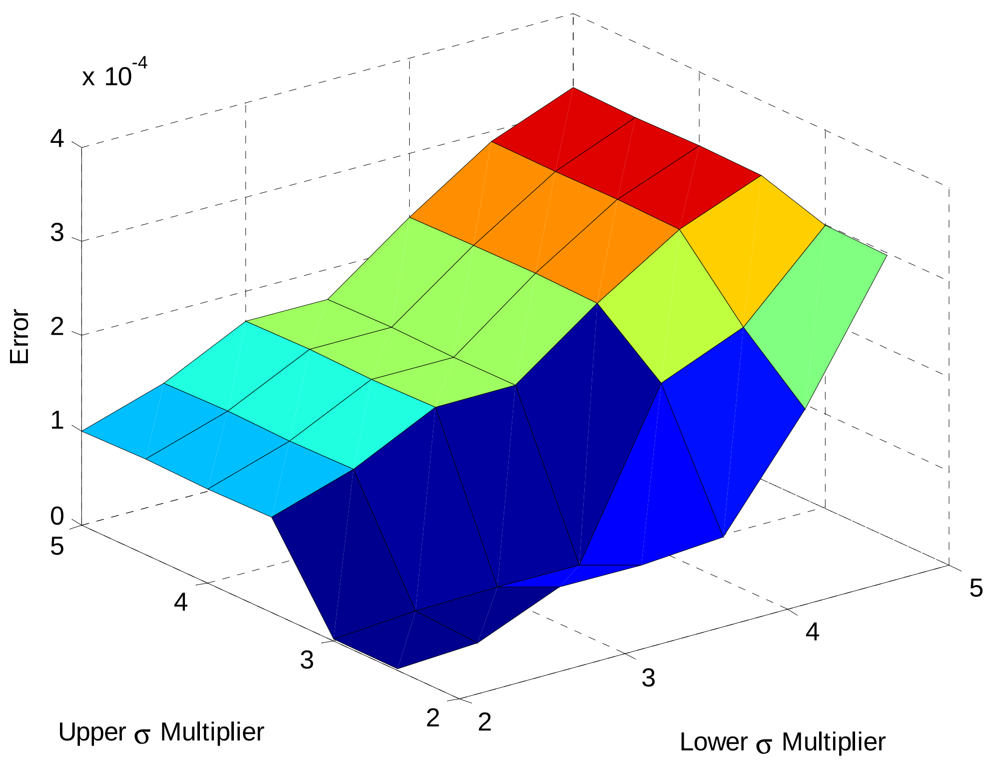
lower	upper	height	width_outer	thread count	height_err	width_err	thread_err
2	2.5	0.022113	0.00422	5.5	0.00226925	0.00025125	6.5
2	3	0.021887	0.004117	4	0.00204325	0.00014825	8
2	3.5	0.021436	0.003929	2.5	0.00159225	3.975E-05	9.5
2	4	0.021436	0.003929	2.5	0.00159225	3.975E-05	9.5
2	4.5	0.021436	0.003929	2.5	0.00159225	3.975E-05	9.5
2	5	0.021436	0.003929	2.5	0.00159225	3.975E-05	9.5
2.5	2.5	0.020985	0.004125	4.5	0.00114125	0.00015625	7.5
2.5	3	0.020985	0.003979	4	0.00114125	1.025E-05	8
2.5	3.5	0.02121	0.003801	2	0.00136625	0.00016775	10
2.5	4	0.02121	0.003801	2	0.00136625	0.00016775	10
2.5	4.5	0.02121	0.003801	2	0.00136625	0.00016775	10
2.5	5	0.02121	0.003844	2	0.00136625	0.00012475	10
3	2.5	0.020985	0.004015	2.5	0.00114125	4.625E-05	9.5
3	3	0.020985	0.00396	4	0.00114125	8.75E-06	8
3	3.5	0.02121	0.003859	4	0.00136625	0.00010975	8
3	4	0.02121	0.003833	4	0.00136625	0.00013575	8
3	4.5	0.02121	0.003833	4	0.00136625	0.00013575	8
3	5	0.02121	0.003797	5	0.00136625	0.00017175	7
3.5	2.5	0.020985	0.004382	6	0.00114125	0.00041325	6
3.5	3	0.020985	0.003836	4.5	0.00114125	0.00013275	7.5
3.5	3.5	0.02121	0.003846	3.5	0.00136625	0.00012275	8.5
3.5	4	0.02121	0.003742	5.5	0.00136625	0.00022675	6.5
3.5	4.5	0.02121	0.003703	7.5	0.00136625	0.00026575	4.5
3.5	5	0.02121	0.003739	7.5	0.00136625	0.00022975	4.5
4	2.5	0.020985	0.004131	3	0.00114125	0.00016225	9
4	3	0.020985	0.003776	2.5	0.00114125	0.00019275	9.5
4	3.5	0.020985	0.003864	3	0.00114125	0.00010475	9
4	4	0.020985	0.003728	5	0.00114125	0.00024075	7
4	4.5	0.020985	0.003716	6	0.00114125	0.00025275	6
4	5	0.020985	0.003768	6	0.00114125	0.00020075	6
4.5	2.5	0.020985	0.00425	2	0.00114125	0.00028125	10
4.5	3	0.020985	0.004056	3	0.00114125	8.725E-05	9
4.5	3.5	0.020759	0.003843	2	0.00091525	0.00012575	10
4.5	4	0.020759	0.003716	4	0.00091525	0.00025275	8
4.5	4.5	0.020759	0.003702	5	0.00091525	0.00026675	7
4.5	5	0.020759	0.003766	5	0.00091525	0.00020275	7
5	2.5	0.020985	0.003928	2.5	0.00114125	0.00004075	9.5
5	3	0.020985	0.00399	3	0.00114125	2.125E-05	9
5	3.5	0.020759	0.003766	2.5	0.00091525	0.00020275	9.5
5	4	0.020759	0.003704	6	0.00091525	0.00026475	6
5	4.5	0.020759	0.0037	7	0.00091525	0.00026875	5
5	5	0.020759	0.003762	7	0.00091525	0.00020675	5

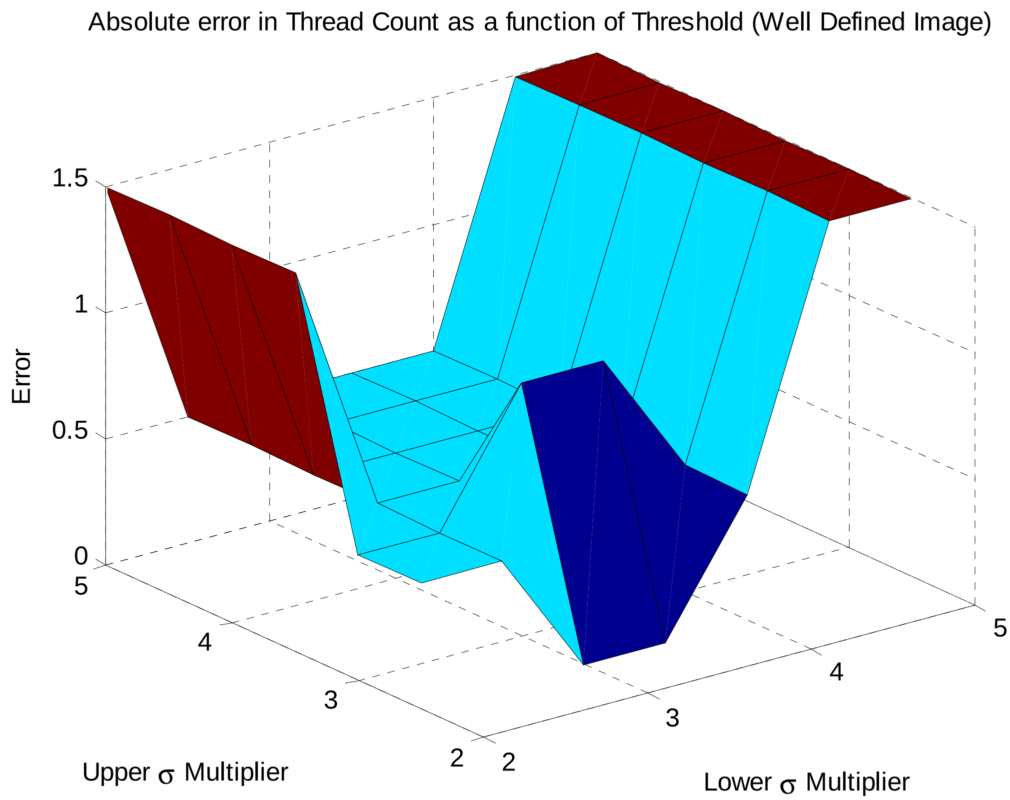
Some charts of the error columns in this data are provided below

Absolute error in Height as a function of Threshold (Well Defined Image)

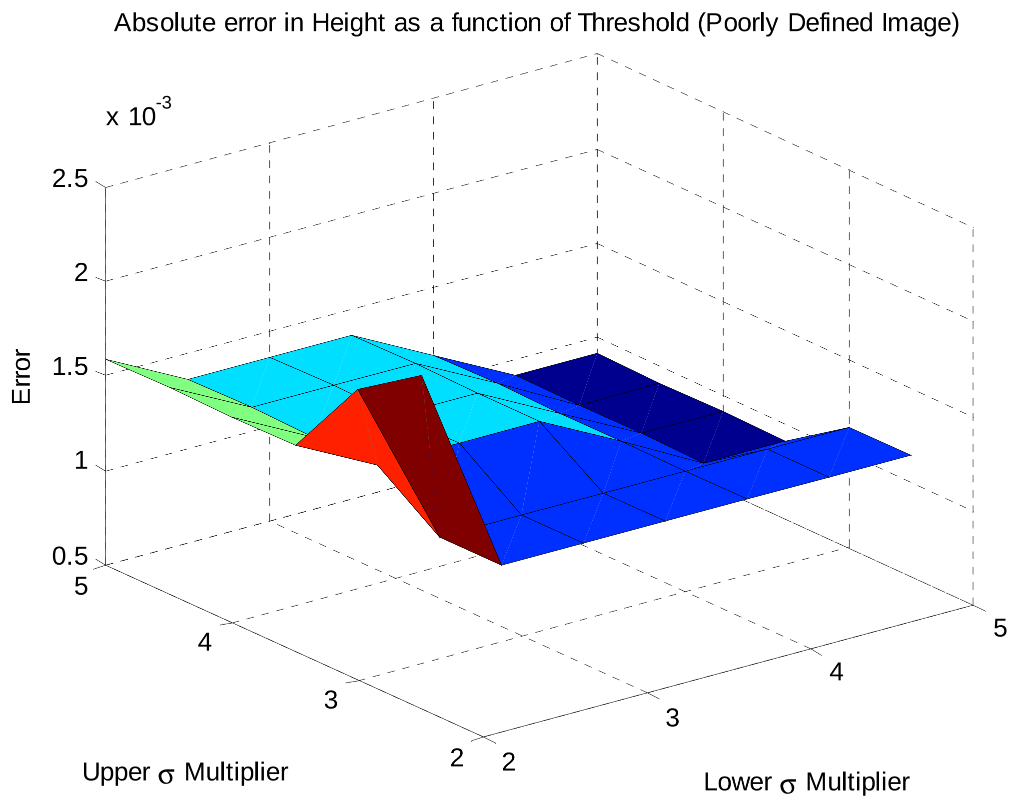


Absolute error in Outer Width as a function of Threshold (Well Defined Image)



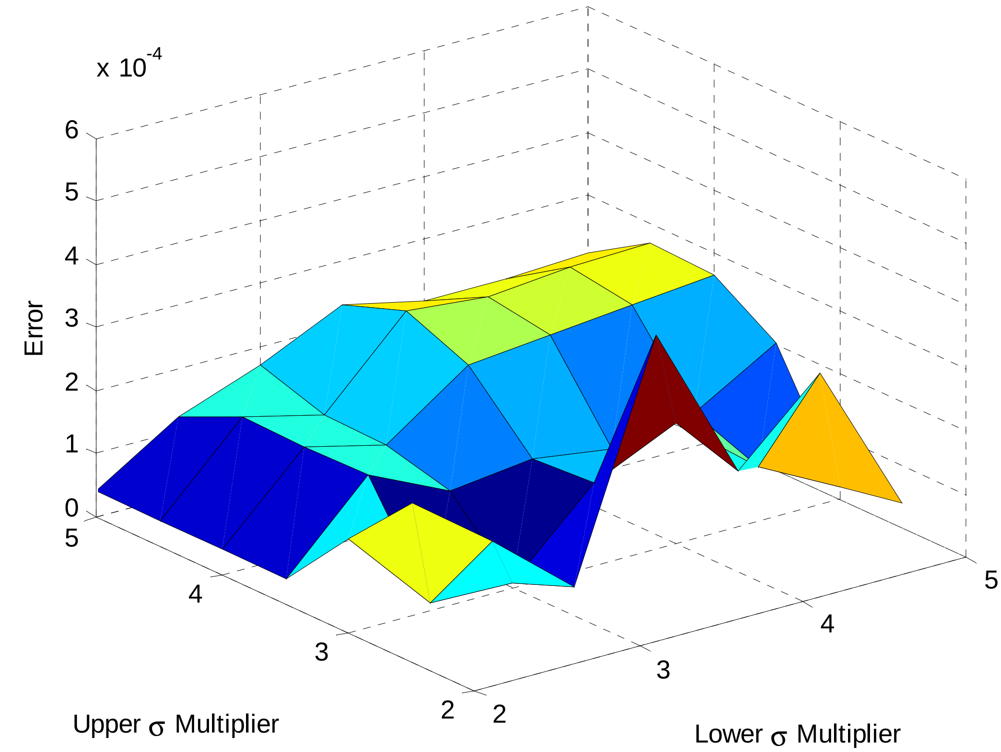


Absolute error in Thread Count as a function of Threshold (Well Defined Image)
Error
1.5
1
0.5
0
5
4
3
2
Upper σ Multiplier
2
3
4
5
Lower σ Multiplier



Absolute error in Height as a function of Threshold (Poorly Defined Image)
x 10-3
Error
2.5
2
1.5
1
0.5
5
4
3
2
Upper σ Multiplier
2
3
4
5
Lower σ Multiplier

Absolute error in Outer Width as a function of Threshold (Poorly Defined Image)



Absolute error in Threshold Count as a function of Threshold (Poorly Defined Image)