In [46]:
```
%matplotlib inline
import cv2
import numpy as np
from matplotlib import pyplot as plt
import math
```

In [47]:
```
#load edge map from sobel opretator
f='../images/sobel_edge_map1.png'
img = cv2.imread(f)
edges_temp=img/255
edges=edges_temp.transpose(2, 0, 1)[0]
edges.shape
```

Out[47]:
```
(496, 1203)
```

In [48]:
```
# function to draw a line into image numpy array.
def draw_detected(img,k,s):
    width=len(img[0])
    cv2.line(img, (0, k), (width, k), (0, 0, 255))
    for i in [1,2]:
        cv2.line(img, ( 0, k+i*s), (width, k+i*s), (0, 0, 255))
        cv2.line(img, (0, k-i*s), (width, k-i*s), (0, 0, 255))
    return img
```

In [49]:
```
# function to draw a line and same to image file.
def draw_lines(k,s,img,f):
    n_img=draw_detected(img,k,s)
    cv2.imwrite(f,n_img)
```

In [50]:
```
#perfrom
vote=np.zeros([len(edges),50])
for y in xrange(len(edges)):
    print y,
    for x in xrange(len(edges[0])):
        if edges[y][x]==1:#if thie pixel is edge point
            try:
                for s in xrange(2,50):
                    k_set=[y-2*s,y-s,y,y+s,y+2*s]
                    for k in k_set:
                        vote[k][s]+=1
            except(IndexError):
                pass

#for development
vote_original=np.copy(vote)
```

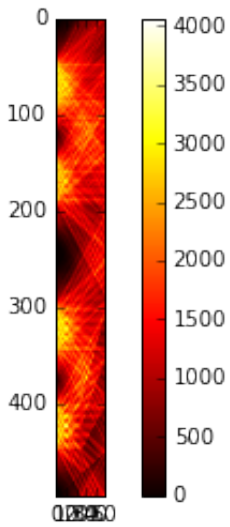
```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94
95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 1
13 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 1
30 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 1
47 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 1
64 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 1
81 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 1
98 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 2
15 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 2
32 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 2
49 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 2
66 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 2
83 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 3
00 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 3
17 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 3
34 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 3
51 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 3
68 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 3
85 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 4
02 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 4
19 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 4
36 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 4
53 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 4
70 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 4
87 488 489 490 491 492 493 494 495
```

In [51]:
```
#show top ten points
top_points=vote.argsort(axis=None)[::-1][0:10]
for i in top_points:
    index=np.unravel_index(i, vote.shape)
    print np.unravel_index(i, vote.shape),vote[index[0]][index[1]]
```

```
(423, 10) 4068.0
(422, 10) 3865.0
(322, 10) 3822.0
(323, 11) 3644.0
(413, 10) 3507.0
(422, 11) 3495.0
(423, 11) 3489.0
(169, 10) 3477.0
(433, 10) 3456.0
(170, 10) 3437.0
```

In [52]:
```
#plot the vote in parameter space.
plt.imshow(vote, cmap=plt.cm.hot)
plt.colorbar()
#plt.savefig("graph.pdf")
```

Out[52]: <matplotlib.colorbar.Colorbar at 0x10aab73d0>



In [53]:
```
#draw top ten lines into image files.
top_points=vote.argsort(axis=None)[::-1][0:10]
for (rank,point) in enumerate(top_points):
    index=np.unravel_index(point, vote.shape)
    value=vote[index[0]][index[1]]
    k=index[0]
    s=index[1]
    img = cv2.imread(f)
    n_img=draw_detected(img,k,s)
    cv2.imwrite(f+"_lines%d_%d_%d_%d.png"%(rank,k,s,value),n_img)
```

In [54]:
```
#try descritization with 5*5 bin size and show top ten points
k_div=5
s_div=5
vote_descrete=np.zeros([int(len(edges)/k_div),int(50/s_div)])
for l in xrange(len(vote_descrete)):
    for m in xrange(len(vote_descrete[0])):
        for plus_k in xrange(k_div):
            for plus_s in xrange(s_div):
                desc_k=k_div*l+plus_k
                desc_s=s_div*m+plus_s
                try:
                    if(vote[desc_k][desc_s]):
                        vote_descrete[l][m]+=vote[desc_k][desc_s]
                except(IndexError):
                    pass;


top_points=vote_descrete.argsort(axis=None)[::-1][0:10]
for i in top_points:
    index=np.unravel_index(i, vote_descrete.shape)
    k=np.unravel_index(i, vote_descrete.shape)[0]*k_div+k_div/2
    s=np.unravel_index(i, vote_descrete.shape)[1]*s_div+s_div/2
    print (k,s),vote_descrete[index[0]][index[1]]
```

```
(322, 7) 71856.0
(67, 7) 70303.0
(72, 7) 70081.0
(332, 7) 69503.0
(327, 7) 69192.0
(62, 7) 68821.0
(317, 7) 68725.0
(322, 12) 68328.0
(77, 7) 68291.0
(422, 7) 68121.0
```

In [55]:
```
#plot the descritized voting space
plt.imshow(vote_descrete, cmap=plt.cm.hot)
plt.colorbar()
#plt.savefig("graph.pdf")
```

Out[55]: <matplotlib.colorbar.Colorbar at 0x10a4c2650>