角度识别设计

目录

[1 初次设计草图 4](#_Toc326588373)

[2. 匹配方法一 5](#_Toc326588374)

[3. 匹配方法二 5](#_Toc326588375)

[4. 匹配前处理 6](#_Toc326588376)

[5. 总体框架 6](#_Toc326588377)

[附录1： 7](#_Toc326588378)

1 初次设计草图

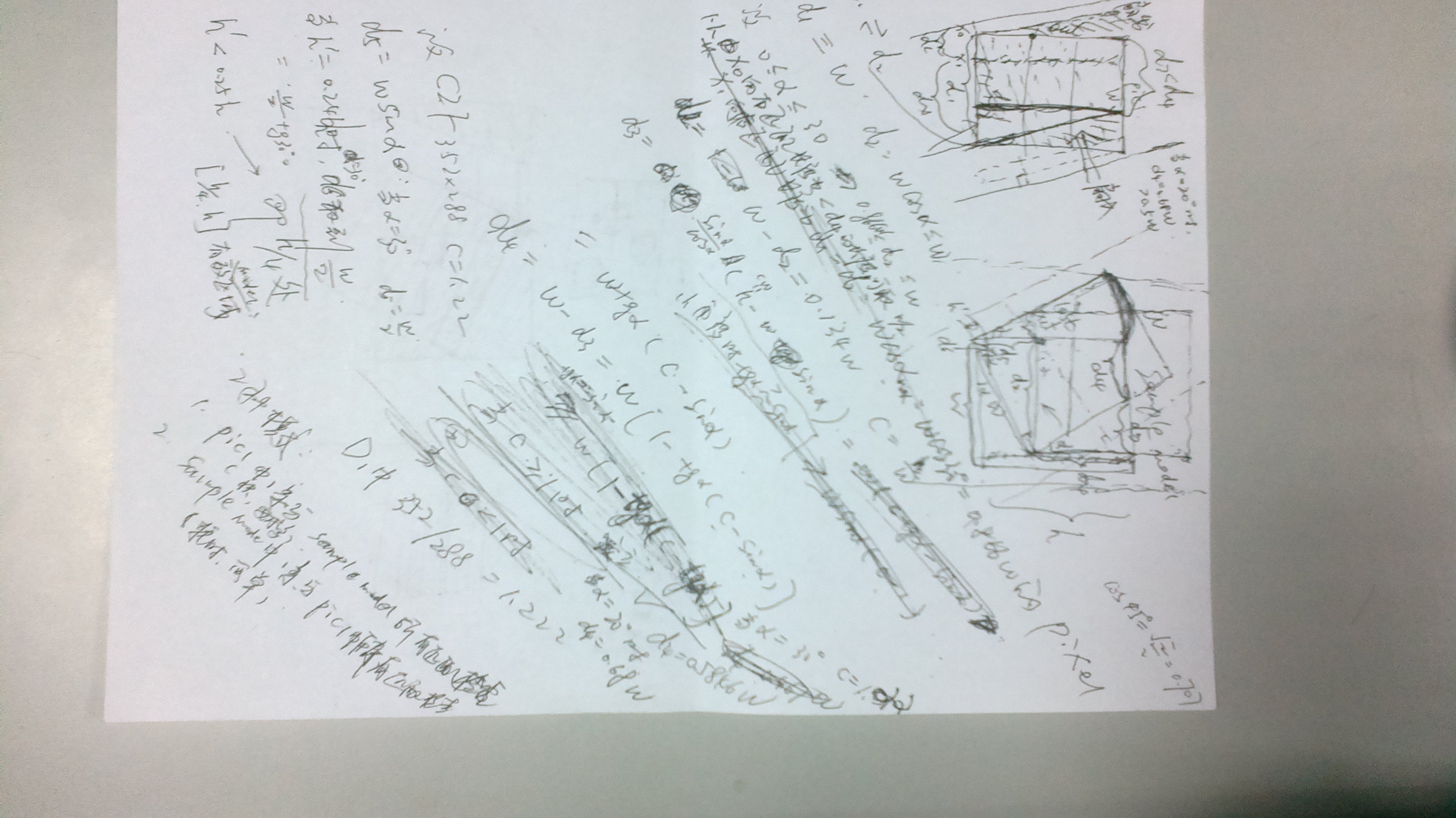


图1 匹配方法草图

图1中，左边为实时采样图。Sample. 右边为模版图 Model

模板图为旋转了一定角度的图片。采样图为机器实际运行时的实际偏移图。理想情况下两副图中的一定区域内的元素应该要完全匹配（标准差为零）。实际中以标准差([cvAvgSdv](#cvAvgSdv))最小为达到匹配的标准。。

## 2.匹配方法一

从可匹配区域的每一行像素进行匹配。精确。计算量大。不知实时是否能来得及。

2．1基本原理：

设a[x]为某一行像素，仅仅当成一组统计数据而已。虽然不是对一个随机变量的采样。 但仍可以用统计方法对其求均值([cvAvg](#cvAvg))，方差([cvAvgSdv](#cvAvgSdv))。每行看成一个向量,向量的维数为数组长度。

设向量α代表模板图中某一行的像素。

β为采样图中某一行元素，为期待与α匹配的一个向量(理想情况下向量相等即匹配)。

γ=β-α的mean=0 ，σ2=0时，β与α完美匹配。

**即对两行元素的差值进行统计分析。E(γ)==0. σ越小表示两行元素越接近。**

γ为一随机变量，γi为其各值.

例如：β=[1.0,2.0,3.0,4.0,5.0], α=[1.1,2.3,3.0,3.9,4.7]

γ=β-α=[-0.1,-0.3,0,0.1,0.3]

mean(γ)= (1/5)Σγi=0;

σ2= (1/5) Σ(γi-γm)2 = 0.04

有以下两种方法。

2.2 S->M

S->M（sample to model）用”采样图中的每一行”去匹配”模板图中固定的一行” 。

这种方法，采样图每一行都要计算均值和方差。摄像机每秒可输入多张采样图，这样的话，计算量就变得很大，对实时性的影响会比较大。

暂且不用这种方法。

2.3 M->S

M->S(model to sample)用”模板图中每一行”匹配”采样图中固定的一行”。

采样图中只取一行（选择哪一行，后面再讨论），进行二值化和统计计算。

而模板图中每一行都要进行二值化和统计预计算。（模板图的计算可以开机前，或刚开开进行了一次纠正之后（此时发生偏移并需要纠正的概率比较小，可以暂时不进行纠正处理））。因而这样就节省了在测试运行中的匹配计算时间。

匹配时，最坏情况下需要匹配的次数为T=,其中h为图片高度,cif 时,h==352.时T=15840，显然这样的计算量仍然太大。

可能还要考虑利用滑动数组对数组进行较正。这样计算量更大。

## 3. 匹配方法二

统计方法。可能不精确。计算量小。

前提：不同角度的区域🡺均值(cvAvg)以及方差不同。

从一个循环单元总体考虑。（可考虑是否对最小循环单元进行排序）

取sample中一块长方形区域RecSample。左下角点X。开始，宽度为d7,高度为3h/4

取model中同样大小的区域RecModel. 起点左上角。

求出每个Rec的均值和方差。与model进行匹配。

可看出，区域越小，速度越快。所以应该使图片中最小循环单元的个数尽量多。

**所以综合可先用方法二匹配，再用方法一进行确认,在指定区域内再匹配一次。如不能发现匹配，可丢弃此次（包括方法二所做的结果）。**

## 4.匹配前处理

为了使图片能够较趋近于理想化，需对图片进行预处理。如二值化和仿射变换。

4.1二值化：sample和model都要进行二值化处理。二值化可使数据变得规范化，要易处理些。二值化方法可采用自适应阀值法（对图片的爆光不良有更好的适应性）。

4.2仿射变换：对于model需进行仿射变换，以得到各种不同旋转角度的model.无需缩放，只要旋转即可。

## 5.二分法减少匹配次数

为减少匹配次数，参考使用二分法可将匹配次数在最坏情况下降低到一半。

比如，当度数总共为32=2^5,最大情况下需要6次即可匹配完毕。

1st判断：是否在(0,32)true内或(-32,0)false内.

2nd判断：if(1st==true)是否在(0,16)true内或(16,32)false.

3rd, if(2nd==true)(0,8)true,or(8,16)false

4……(0,4)

5…………(0,2)

6……….(1)

6. 将图像变为二值图像方法

a.先将图像用cvloadImage(filename.jpg, CV\_LOAD\_IMAGE\_GRAYSCALE)装载成为单通道灰度图像.

b. 或者1.cvloadImage(file.jpg, CV\_LOAD\_IMAGE\_COLOR)

2. cvCvtColor(img,gray1channel,CV\_RGB2GRAY);

## n. 主要函数

void **cvSetImageROI**([IplImage](http://opencv.willowgarage.com/documentation/c/core_basic_structures.html#IplImage)\*image, [CvRect](http://opencv.willowgarage.com/documentation/c/core_basic_structures.html#CvRect)rect)---turn on ROI

[CvRect](http://opencv.willowgarage.com/documentation/c/core_basic_structures.html#CvRect)  **cvGetImageROI**(const [IplImage](http://opencv.willowgarage.com/documentation/c/core_basic_structures.html#IplImage)\*image)

## void cvResetImageROI([IplImage](http://opencv.willowgarage.com/documentation/c/core_basic_structures.html#IplImage)\*image)—turn off ROI

## void cvSetZero([CvArr](http://opencv.willowgarage.com/documentation/c/core_basic_structures.html#CvArr)\*arr)

与ROI平行的另一种方法，新建非在位子图像，可参考英文版 learning opencv p46,

## 附录1：

图像处理和图像识别中常用的OpenCV函数

1、cvLoadImage：将图像文件加载至内存；

2、cvNamedWindow：在屏幕上创建一个窗口；

3、cvShowImage：在一个已创建好的窗口中显示图像；

4、cvWaitKey：使程序暂停，等待用户触发一个按键操作；

5、cvReleaseImage：释放图像文件所分配的内存；

6、cvDestroyWindow：销毁显示图像文件的窗口；

7、cvCreateFileCapture：通过参数设置确定要读入的AVI文件；

8、cvQueryFrame：用来将下一帧视频文件载入内存；

9、cvReleaseCapture：释放CvCapture结构开辟的内存空间；

10、cvCreateTrackbar：创建一个滚动条；

11、cvSetCaptureProperty：设置CvCapture对象的各种属性；

12、cvGetCaptureProperty：查询CvCapture对象的各种属性；

13、cvGetSize：当前图像结构的大小；

14、cvSmooth：对图像进行平滑处理；

15、cvPyrDown：图像金字塔，降采样，图像缩小为原来四分之一；

16、cvCanny：Canny边缘检测；

17、cvCreateCameraCapture：从摄像设备中读入数据；

18、cvCreateVideoWriter：创建一个写入设备以便逐帧将视频流写入视频文件；

19、cvWriteFrame：逐帧将视频流写入文件；

20、cvReleaseVideoWriter：释放CvVideoWriter结构开辟的内存空间；

21、CV\_MAT\_ELEM：从矩阵中得到一个元素；

22、cvAbs：计算数组中所有元素的绝对值；

23、cvAbsDiff：计算两个数组差值的绝对值；

24、cvAbsDiffS：计算数组和标量差值的绝对值；

25、cvAdd：两个数组的元素级的加运算；

26、cvAddS：一个数组和一个标量的元素级的相加运算；

27、cvAddWeighted：两个数组的元素级的加权相加运算(alpha运算)；

28、cvAvg：计算数组中所有元素的平均值；

29、cvAvgSdv：计算数组中所有元素的绝对值和标准差；

30、cvCalcCovarMatrix：计算一组n维空间向量的协方差；

31、cvCmp：对两个数组中的所有元素运用设置的比较操作；

32、cvCmpS：对数组和标量运用设置的比较操作；

33、cvConvertScale：用可选的缩放值转换数组元素类型；

34、cvCopy：把数组中的值复制到另一个数组中；

35、cvCountNonZero：计算数组中非0值的个数；

36、cvCrossProduct：计算两个三维向量的向量积(叉积)；

37、cvCvtColor：将数组的通道从一个颜色空间转换另外一个颜色空间；

38、cvDet：计算方阵的行列式；

39、cvDiv：用另外一个数组对一个数组进行元素级的除法运算；

40、cvDotProduct：计算两个向量的点积；

41、cvEigenVV：计算方阵的特征值和特征向量；

42、cvFlip：围绕选定轴翻转；

43、cvGEMM：矩阵乘法；

44、cvGetCol：从一个数组的列中复制元素；

45、cvGetCols：从数据的相邻的多列中复制元素；

46、cvGetDiag：复制数组中对角线上的所有元素；

47、cvGetDims：返回数组的维数；

48、cvGetDimSize：返回一个数组的所有维的大小；

49、cvGetRow：从一个数组的行中复制元素值；

50、cvGetRows：从一个数组的多个相邻的行中复制元素值；

51、cvGetSize：得到二维的数组的尺寸，以CvSize返回；

52、cvGetSubRect：从一个数组的子区域复制元素值；

53、cvInRange：检查一个数组的元素是否在另外两个数组中的值的范围内；

54、cvInRangeS：检查一个数组的元素的值是否在另外两个标量的范围内；

55、cvInvert：求矩阵的逆；

56、cvMahalonobis：计算两个向量间的马氏距离；

57、cvMax：在两个数组中进行元素级的取最大值操作；

58、cvMaxS：在一个数组和一个标量中进行元素级的取最大值操作；

59、cvMerge：把几个单通道图像合并为一个多通道图像；

60、cvMin：在两个数组中进行元素级的取最小值操作；

61、cvMinS：在一个数组和一个标量中进行元素级的取最小值操作；

62、cvMinMaxLoc：寻找数组中的最大最小值；

63、cvMul：计算两个数组的元素级的乘积(点乘)；

64、cvNot：按位对数组中的每一个元素求反；

65、cvNormalize：将数组中元素进行归一化；

66、cvOr：对两个数组进行按位或操作；

67、cvOrs：在数组与标量之间进行按位或操作；

68、cvReduce：通过给定的操作符将二维数组简为向量；

69、cvRepeat：以平铺的方式进行数组复制；

70、cvSet：用给定值初始化数组；

71、cvSetZero：将数组中所有元素初始化为0；

72、cvSetIdentity：将数组中对角线上的元素设为1，其他置0；

73、cvSolve：求出线性方程组的解；

74、cvSplit：将多通道数组分割成多个单通道数组；

75、cvSub：两个数组元素级的相减；

76、cvSubS：元素级的从数组中减去标量；

77、cvSubRS：元素级的从标量中减去数组；

78、cvSum：对数组中的所有元素求和；

79、cvSVD：二维矩阵的奇异值分解；

80、cvSVBkSb：奇异值回代计算；

81、cvTrace：计算矩阵迹；

82、cvTranspose：矩阵的转置运算；

83、cvXor：对两个数组进行按位异或操作；

84、cvXorS：在数组和标量之间进行按位异或操作；

85、cvZero：将所有数组中的元素置为0；

86、cvConvertScaleAbs：计算可选的缩放值的绝对值之后再转换数组元素的类型；

87、cvNorm：计算数组的绝对范数， 绝对差分范数或者相对差分范数；

88、cvAnd：对两个数组进行按位与操作；

89、cvAndS：在数组和标量之间进行按位与操作；

90、cvScale：是cvConvertScale的一个宏，可以用来重新调整数组的内容，并且可以将参数从一种数

                  据类型转换为另一种；

91、cvT：是函数cvTranspose的缩写；

92、cvLine：画直线；

93、cvRectangle：画矩形；

94、cvCircle：画圆；

95、cvEllipse：画椭圆；

96、cvEllipseBox：使用外接矩形描述椭圆；

97、cvFillPoly、cvFillConvexPoly、cvPolyLine：画多边形；

98、cvPutText：在图像上输出一些文本；

99、cvInitFont：采用一组参数配置一些用于屏幕输出的基本个特定字体；

100、cvSave：矩阵保存；

101、cvLoad：矩阵读取；

102、cvOpenFileStorage：为读/写打开存储文件；

103、cvReleaseFileStorage：释放存储的数据；

104、cvStartWriteStruct：开始写入新的数据结构；

105、cvEndWriteStruct：结束写入数据结构；

106、cvWriteInt：写入整数型；

107、cvWriteReal：写入浮点型；

108、cvWriteString：写入字符型；

109、cvWriteComment：写一个XML或YAML的注释字串；

110、cvWrite：写一个对象；

111、cvWriteRawData：写入多个数值；

112、cvWriteFileNode：将文件节点写入另一个文件存储器；

113、cvGetRootFileNode：获取存储器最顶层的节点；

114、cvGetFileNodeByName：在映图或存储器中找到相应节点；

115、cvGetHashedKey：为名称返回一个惟一的指针；

116、cvGetFileNode：在映图或文件存储器中找到节点；

117、cvGetFileNodeName：返回文件的节点名；

118、cvReadInt：读取一个无名称的整数型；

119、cvReadIntByName：读取一个有名称的整数型；

120、cvReadReal：读取一个无名称的浮点型；

121、cvReadRealByName：读取一个有名称的浮点型；

122、cvReadString：从文件节点中寻找字符串；

123、cvReadStringByName：找到一个有名称的文件节点并返回它；

124、cvRead：将对象解码并返回它的指针；

125、cvReadByName：找到对象并解码；

126、cvReadRawData：读取多个数值；

127、cvStartReadRawData：初始化文件节点序列的读取；

128、cvReadRawDataSlice：读取文件节点的内容；

129、cvGetModuleInfo：检查IPP库是否已经正常安装并且检验运行是否正常；

130、cvResizeWindow：用来调整窗口的大小；

131、cvSaveImage：保存图像；

132、cvMoveWindow：将窗口移动到其左上角为x,y的位置；

133、cvDestroyAllWindow：用来关闭所有窗口并释放窗口相关的内存空间；

134、cvGetTrackbarPos：读取滑动条的值；

135、cvSetTrackbarPos：设置滑动条的值；

136、cvGrabFrame：用于快速将视频帧读入内存；

137、cvRetrieveFrame：对读入帧做所有必须的处理；

138、cvConvertImage：用于在常用的不同图像格式之间转换；

139、cvErode：形态腐蚀；

140、cvDilate：形态学膨胀；

141、cvMorphologyEx：更通用的形态学函数；

142、cvFloodFill：漫水填充算法，用来进一步控制哪些区域将被填充颜色；

143、cvResize：放大或缩小图像；

144、cvPyrUp：图像金字塔，将现有的图像在每个维度上都放大两倍；

145、cvPyrSegmentation：利用金字塔实现图像分割；

146、cvThreshold：图像阈值化；

147、cvAcc：可以将8位整数类型图像累加为浮点图像；

148、cvAdaptiveThreshold：图像自适应阈值；

149、cvFilter2D：图像卷积；

150、cvCopyMakeBorder：将特定的图像轻微变大，然后以各种方式自动填充图像边界；

151、cvSobel：图像边缘检测，Sobel算子；

152、cvLaplace：拉普拉斯变换、图像边缘检测；

153、cvHoughLines2：霍夫直线变换；

154、cvHoughCircles：霍夫圆变换；

155、cvRemap：图像重映射，校正标定图像，图像插值；

156、cvWarpAffine：稠密仿射变换；

157、cvGetQuadrangleSubPix：仿射变换；

158、cvGetAffineTransform：仿射映射矩阵的计算；

159、cvCloneImage：将整个IplImage结构复制到新的IplImage中；

160、cv2DRotationMatrix：仿射映射矩阵的计算；

161、cvTransform：稀疏仿射变换；

162、cvWarpPerspective：密集透视变换(单应性)；

163、cvGetPerspectiveTransform：计算透视映射矩阵；

164、cvPerspectiveTransform：稀疏透视变换；

165、cvCartToPolar：将数值从笛卡尔空间到极坐标(极性空间)进行映射；

166、cvPolarToCart：将数值从极性空间到笛卡尔空间进行映射；

167、cvLogPolar：对数极坐标变换；

168、cvDFT：离散傅里叶变换；

169、cvMulSpectrums：频谱乘法；

170、cvDCT：离散余弦变换；

171、cvIntegral：计算积分图像；

172、cvDistTransform：图像的距离变换；

173、cvEqualizeHist：直方图均衡化；

174、cvCreateHist：创建一新直方图；

175、cvMakeHistHeaderForArray：根据已给出的数据创建直方图；

176、cvNormalizeHist：归一化直方图；

177、cvThreshHist：直方图阈值函数；

178、cvCalcHist：从图像中自动计算直方图；

179、cvCompareHist：用于对比两个直方图的相似度；

180、cvCalcEMD2：陆地移动距离(EMD)算法；

181、cvCalcBackProject：反向投影；

182、cvCalcBackProjectPatch：图块的方向投影；

183、cvMatchTemplate：模板匹配；

184、cvCreateMemStorage：用于创建一个内存存储器；

185、cvCreateSeq：创建序列；

186、cvSeqInvert：将序列进行逆序操作；

187、cvCvtSeqToArray：复制序列的全部或部分到一个连续内存数组中；

188、cvFindContours：从二值图像中寻找轮廓；

189、cvDrawContours：绘制轮廓；

190、cvApproxPoly：使用多边形逼近一个轮廓；

191、cvContourPerimeter：轮廓长度；

192、cvContoursMoments：计算轮廓矩；

193、cvMoments：计算Hu不变矩；

194、cvMatchShapes：使用矩进行匹配；

195、cvInitLineIterator：对任意直线上的像素进行采样；

196、cvSampleLine：对直线采样；

197、cvAbsDiff：帧差；

198、cvWatershed：分水岭算法；

199、cvInpaint：修补图像；

200、cvGoodFeaturesToTrack：寻找角点；

201、cvFindCornerSubPix：用于发现亚像素精度的角点位置；

202、cvCalcOpticalFlowLK：实现非金字塔的Lucas-Kanade稠密光流算法；

203、cvMeanShift：mean-shift跟踪算法；

204、cvCamShift：camshift跟踪算法；

205、cvCreateKalman：创建Kalman滤波器；

206、cvCreateConDensation：创建condensation滤波器；

207、cvConvertPointsHomogenious：对齐次坐标进行转换；

208、cvFindChessboardCorners：定位棋盘角点；

209、cvFindHomography：计算单应性矩阵；

210、cvRodrigues2：罗德里格斯变换；

211、cvFitLine：直线拟合算法；

212、cvCalcCovarMatrix：计算协方差矩阵；

213、cvInvert：计算协方差矩阵的逆矩阵；

214、cvMahalanobis：计算Mahalanobis距离；

215、cvKMeans2：K均值；

216、cvCloneMat：根据一个已有的矩阵创建一个新矩阵；

217、cvPreCornerDetect：计算用于角点检测的特征图；

218、cvGetImage：CvMat图像数据格式转换成IplImage图像数据格式；

219、cvMatMul：两矩阵相乘；