

# 2015-06-17-084340-opencv

William Stein

6/17/2015

## Contents

```
import cv
```

```
help(cv)
```

Help on module cv:

NAME

cv

FILE

/usr/local/lib/python2.7/dist-packages/cv.py

FUNCTIONS

16SC(...)

CV\_16SC(n) -> int

16UC(...)

CV\_16UC(n) -> int

32FC(...)

CV\_32FC(n) -> int

32SC(...)

CV\_32SC(n) -> int

64FC(...)

CV\_64FC(n) -> int

8SC(...)

CV\_8SC(n) -> int

8UC(...)

CV\_8UC(n) -> int

Abs(...)

Abs(src, dst) -> None

AbsDiff(...)

```

    AbsDiff(src1, src2, dst) -> None

AbsDiffS(...)
    AbsDiffS(src, dst, value) -> None

Acc(...)
    Acc(image, sum [, mask]) -> None

AdaptiveThreshold(...)
    AdaptiveThreshold(src, dst, maxValue [, adaptive_method [, thresholdType [,
blockSize [, param1]]]]) -> None

Add(...)
    Add(src1, src2, dst [, mask]) -> None

AddS(...)
    AddS(src, value, dst [, mask]) -> None

AddWeighted(...)
    AddWeighted(src1, alpha, src2, beta, gamma, dst) -> None

And(...)
    And(src1, src2, dst [, mask]) -> None

AndS(...)
    AndS(src, value, dst [, mask]) -> None

ApproxChains(...)
    ApproxChains(src_seq, storage [, method [, parameter [, minimal_perimeter [,
recursive]]]]) -> CvSeq*

ApproxPoly(...)
    ApproxPoly(src_seq, storage, method [, parameter [, parameter2]]) -> None

ArcLength(...)
    ArcLength(curve [, slice [, isClosed]]) -> double

Avg(...)
    Avg(arr [, mask]) -> CvScalar

AvgSdv(...)
    AvgSdv(arr [, mask]) -> mean,stdDev

BackProjectPCA(...)
    BackProjectPCA(proj, avg, eigenvects, result) -> None

BoundingRect(...)
    BoundingRect(points [, update]) -> CvRect

BoxPoints(...)
    BoxPoints(box) -> points

```

```
CMP(...)
    CV_CMP(a, b) -> int

CV_16SC(...)
    CV_16SC(n) -> int

CV_16UC(...)
    CV_16UC(n) -> int

CV_32FC(...)
    CV_32FC(n) -> int

CV_32SC(...)
    CV_32SC(n) -> int

CV_64FC(...)
    CV_64FC(n) -> int

CV_8SC(...)
    CV_8SC(n) -> int

CV_8UC(...)
    CV_8UC(n) -> int

CV_CMP(...)
    CV_CMP(a, b) -> int

CV_FOURCC(...)
    CV_FOURCC(c1, c2, c3, c4) -> int

CV_IABS(...)
    CV_IABS(a) -> int

CV_IS_SEQ_CLOSED(...)
    CV_IS_SEQ_CLOSED(s) -> int

CV_IS_SEQ_CONVEX(...)
    CV_IS_SEQ_CONVEX(s) -> int

CV_IS_SEQ_CURVE(...)
    CV_IS_SEQ_CURVE(s) -> int

CV_IS_SEQ_HOLE(...)
    CV_IS_SEQ_HOLE(s) -> int

CV_IS_SEQ_INDEX(...)
    CV_IS_SEQ_INDEX(s) -> int

CV_IS_SEQ_SIMPLE(...)
    CV_IS_SEQ_SIMPLE(s) -> int
```

```

CV_MAKETYPE(...)
    CV_MAKETYPE(depth, cn) -> int

CV_MAT_CN(...)
    CV_MAT_CN(i) -> int

CV_MAT_DEPTH(...)
    CV_MAT_DEPTH(i) -> int

CV_RGB(...)
    CV_RGB(red, grn, blu) -> CvScalar

CV_SIGN(...)
    CV_SIGN(a) -> int

CalcArrBackProject(...)
    CalcArrBackProject(image, back_project, hist) -> None

CalcArrHist(...)
    CalcArrHist(image, hist [, accumulate [, mask]]) -> None

CalcBackProject(...)
    CalcBackProject(image, back_project, hist) -> None

CalcBackProjectPatch(...)
    CalcBackProjectPatch(images, dst, patch_size, hist, method, factor) -> None

CalcCovarMatrix(...)
    CalcCovarMatrix(vects, covMat, avg, flags) -> None

CalcEMD2(...)
    CalcEMD2(signature1, signature2, distance_type [, distance_func [, cost_matrix [,
flow [, lower_bound [, userdata]]]]]) -> float

CalcGlobalOrientation(...)
    CalcGlobalOrientation(orientation, mask, mhi, timestamp, duration) -> double

CalcHist(...)
    CalcHist(image, hist [, accumulate [, mask]]) -> None

CalcMotionGradient(...)
    CalcMotionGradient(mhi, mask, orientation, delta1, delta2 [, apertureSize]) ->
None

CalcOpticalFlowBM(...)
    CalcOpticalFlowBM(prev, curr, blockSize, shiftSize, max_range, usePrevious, velx,
vely) -> None

CalcOpticalFlowFarneback(...)
    CalcOpticalFlowFarneback(prev, curr, flow [, pyr_scale [, levels [, winsize [,

```

```

iterations [, poly_n [, poly_sigma [, flags]]]]]])) -> None

CalcOpticalFlowHS(...)
  CalcOpticalFlowHS(prev, curr, usePrevious, velx, vely, lambda, criteria) -> None

CalcOpticalFlowLK(...)
  CalcOpticalFlowLK(prev, curr, winSize, velx, vely) -> None

CalcOpticalFlowPyrLK(...)
  CalcOpticalFlowPyrLK(prev, curr, prevPyr, currPyr, prevFeatures, winSize, level,
criteria, flags [, guesses]) -> currFeatures,status,track_error

CalcPCA(...)
  CalcPCA(data, avg, eigenvalues, eigenvectors, flags) -> None

CalcProbDensity(...)
  CalcProbDensity(hist1, hist2, dst_hist [, scale]) -> None

CalcSubdivVoronoi2D(...)
  CalcSubdivVoronoi2D(subdiv) -> None

CalibrateCamera2(...)
  CalibrateCamera2(objectPoints, imagePoints, pointCounts, imageSize, cameraMatrix,
distCoeffs, rvecs, tvecs [, flags]) -> None

CalibrationMatrixValues(...)
  CalibrationMatrixValues(calibMatr, image_size [, apertureWidth [,
apertureHeight]]) -> fovx,fovy,focalLength,principalPoint,pixelAspectRatio

CamShift(...)
  CamShift(prob_image, window, criteria) -> int,comp,box

Canny(...)
  Canny(image, edges, threshold1, threshold2 [, aperture_size]) -> None

CaptureFromCAM(...)
  CaptureFromCAM(index) -> CvCapture*

CaptureFromFile(...)
  CaptureFromFile(filename) -> CvCapture*

CartToPolar(...)
  CartToPolar(x, y, magnitude [, angle [, angleInDegrees]]) -> None

Cbrt(...)
  Cbrt(value) -> float

Ceil(...)
  Ceil(value) -> int

CheckArr(...)

```

```

    CheckArr(arr [, flags [, min_val [, max_val]]]) -> int

CheckContourConvexity(...)
    CheckContourConvexity(contour) -> int

Circle(...)
    Circle(img, center, radius, color [, thickness [, lineType [, shift]]]) -> None

ClearHist(...)
    ClearHist(hist) -> None

ClearND(...)
    ClearND(arr, idx) -> None

ClearSeq(...)
    ClearSeq(seq) -> None

ClearSubdivVoronoi2D(...)
    ClearSubdivVoronoi2D(subdiv) -> None

ClipLine(...)
    ClipLine(imgSize, pt1, pt2) -> point1,point2

CloneImage(...)
    CloneImage(image) -> IplImage*

CloneMat(...)
    CloneMat(mat) -> CvMat*

CloneMatND(...)
    CloneMatND(mat) -> CvMatND*

CloneSeq(...)
    CloneSeq(seq, storage) -> None

Cmp(...)
    Cmp(src1, src2, dst, cmpOp) -> None

CmpS(...)
    CmpS(src, value, dst, cmpOp) -> None

CompareHist(...)
    CompareHist(hist1, hist2, method) -> double

ComputeCorrespondEpilines(...)
    ComputeCorrespondEpilines(points, whichImage, F, lines) -> None

ContourArea(...)
    ContourArea(contour [, slice]) -> double

Convert(...)

```

```

    Convert(src, dst) -> None

ConvertImage(...)
    ConvertImage(src, dst [, flags]) -> None

ConvertMaps(...)
    ConvertMaps(mapx, mapy, mapxy, mapalpha) -> None

ConvertPointsHomogeneous(...)
    ConvertPointsHomogeneous(src, dst) -> None

ConvertScale(...)
    ConvertScale(src, dst [, scale [, shift]]) -> None

ConvertScaleAbs(...)
    ConvertScaleAbs(src, dst [, scale [, shift]]) -> None

ConvexHull2(...)
    ConvexHull2(points, storage [, orientation [, return_points]]) -> CvSeq*

ConvexityDefects(...)
    ConvexityDefects(contour, convexhull, storage) -> CvSeqOfCvConvexityDefect*

Copy(...)
    Copy(src, dst [, mask]) -> None

CopyMakeBorder(...)
    CopyMakeBorder(src, dst, offset, bordertype [, value]) -> None

CornerEigenValsAndVecs(...)
    CornerEigenValsAndVecs(image, eigenvv, blockSize [, aperture_size]) -> None

CornerHarris(...)
    CornerHarris(image, harris_dst, blockSize [, aperture_size [, k]]) -> None

CornerMinEigenVal(...)
    CornerMinEigenVal(image, eigenval, blockSize [, aperture_size]) -> None

CountNonZero(...)
    CountNonZero(arr) -> int

CreateCameraCapture(...)
    CreateCameraCapture(index) -> CvCapture*

CreateData(...)
    CreateData(arr) -> None

CreateFileCapture(...)
    CreateFileCapture(filename) -> CvCapture*

CreateHist(...)

```

```

    CreateHist(dims, type [, ranges [, uniform]]) -> CvHistogram

CreateImage(...)
    CreateImage(size, depth, channels) -> IplImage*

CreateImageHeader(...)
    CreateImageHeader(size, depth, channels) -> IplImage*

CreateKalman(...)
    CreateKalman(dynam_params, measure_params [, control_params]) -> CvKalman*

CreateMat(...)
    CreateMat(rows, cols, type) -> CvMat

CreateMatHeader(...)
    CreateMatHeader(rows, cols, type) -> CvMat

CreateMatND(...)
    CreateMatND(dims, type) -> CvMatND

CreateMatNDHeader(...)
    CreateMatNDHeader(dims, type) -> CvMatND

CreateMemStorage(...)
    CreateMemStorage( [, blockSize]) -> CvMemStorage

CreatePOSITObject(...)
    CreatePOSITObject(points) -> CvPOSITObject*

CreateStereoBMState(...)
    CreateStereoBMState( [, preset [, numberOfDisparities]]) -> CvStereoBMState*

CreateStereoGCState(...)
    CreateStereoGCState(numberOfDisparities, maxIters) -> CvStereoGCState*

CreateStructuringElementEx(...)
    CreateStructuringElementEx(cols, rows, anchorX, anchorY, shape [, values]) ->
IplConvKernel*

CreateSubdivDelaunay2D(...)
    CreateSubdivDelaunay2D(rect, storage) -> CvSubdiv2D*

CreateTrackbar(...)
    CreateTrackbar(trackbarName, windowName, value, count, onChange) -> None

CreateVideoWriter(...)
    CreateVideoWriter(filename, fourcc, fps, frame_size [, is_color]) ->
CvVideoWriter*

CrossProduct(...)
    CrossProduct(src1, src2, dst) -> None

```



```

CvtColor(...)
    CvtColor(src, dst, code) -> None

CvtPixToPlane(...)
    CvtPixToPlane(src, dst0, dst1, dst2, dst3) -> None

CvtScale(...)
    CvtScale(src, dst [, scale [, shift]]) -> None

DCT(...)
    DCT(src, dst, flags) -> None

DFT(...)
    DFT(src, dst, flags [, nonzeroRows]) -> None

DecodeImage(...)
    DecodeImage(buf [, iscolor]) -> IplImage*

DecodeImageM(...)
    DecodeImageM(buf [, iscolor]) -> CvMat*

DecomposeProjectionMatrix(...)
    DecomposeProjectionMatrix(projMatrix, cameraMatrix, rotMatrix, transVect [,
rotMatrX [, rotMatrY [, rotMatrZ]]) -> eulerAngles

DestroyAllWindows(...)
    DestroyAllWindows() -> None

DestroyWindow(...)
    DestroyWindow(name) -> None

Det(...)
    Det(mat) -> double

Dilate(...)
    Dilate(src, dst [, element [, iterations]]) -> None

DistTransform(...)
    DistTransform(src, dst [, distance_type [, mask_size [, mask [, labels]]]]) ->
None

Div(...)
    Div(src1, src2, dst [, scale]) -> None

DotProduct(...)
    DotProduct(src1, src2) -> double

DrawChessboardCorners(...)
    DrawChessboardCorners(image, patternSize, corners, patternWasFound) -> None

```

```

DrawContours(...)
    DrawContours(img, contour, external_color, hole_color, max_level [, thickness [,
lineType [, offset]]) -> None

EigenVV(...)
    EigenVV(mat, evecs, evals, eps [, lowindex [, highindex]]) -> None

Ellipse(...)
    Ellipse(img, center, axes, angle, start_angle, end_angle, color [, thickness [,
lineType [, shift]]) -> None

EllipseBox(...)
    EllipseBox(img, box, color [, thickness [, lineType [, shift]]) -> None

EncodeImage(...)
    EncodeImage(ext, image [, params]) -> CvMat*

EqualizeHist(...)
    EqualizeHist(src, dst) -> None

Erode(...)
    Erode(src, dst [, element [, iterations]]) -> None

EstimateRigidTransform(...)
    EstimateRigidTransform(A, B, M, full_affine) -> None

Exp(...)
    Exp(src, dst) -> None

ExtractSURF(...)
    ExtractSURF(image, mask, storage, params) -> keypoints,descriptors

FOURCC(...)
    CV_FOURCC(c1, c2, c3, c4) -> int

FastArctan(...)
    FastArctan(y, x) -> float

FillConvexPoly(...)
    FillConvexPoly(img, pn, color [, lineType [, shift]]) -> None

FillPoly(...)
    FillPoly(img, polys, color [, lineType [, shift]]) -> None

Filter2D(...)
    Filter2D(src, dst, kernel [, anchor]) -> None

FindChessboardCorners(...)
    FindChessboardCorners(image, patternSize [, flags]) -> corners

FindContours(...)

```

```

    FindContours(image, storage [, mode [, method [, offset]]]) -> CvSeq

FindCornerSubPix(...)
    FindCornerSubPix(image, corners, win, zero_zone, criteria) -> corners

FindDataMatrix(...)

FindExtrinsicCameraParams2(...)
    FindExtrinsicCameraParams2(objectPoints, imagePoints, cameraMatrix, distCoeffs,
rvec, tvec [, useExtrinsicGuess]) -> None

FindFundamentalMat(...)
    FindFundamentalMat(points1, points2, fundamentalMatrix [, method [, param1 [,
param2 [, status]]]]) -> int

FindHomography(...)
    FindHomography(srcPoints, dstPoints, H [, method [, ransacReprojThreshold [,
status]]]) -> None

FindNearestPoint2D(...)
    FindNearestPoint2D(subdiv, pt) -> CvSubdiv2DPoint*

FindStereoCorrespondenceBM(...)
    FindStereoCorrespondenceBM(left, right, disparity, state) -> None

FindStereoCorrespondenceGC(...)
    FindStereoCorrespondenceGC(left, right, dispLeft, dispRight, state [,
useDisparityGuess]) -> None

FitEllipse2(...)
    FitEllipse2(points) -> CvBox2D

FitLine(...)
    FitLine(points, dist_type, param, reps, aeps) -> line

Flip(...)
    Flip(src [, dst [, flipMode]]) -> None

FloodFill(...)
    FloodFill(image, seed_point, new_val [, lo_diff [, up_diff [, flags [, mask]]]])
-> comp

Floor(...)
    Floor(value) -> int

GEMM(...)
    GEMM(src1, src2, alpha, src3, beta, dst [, tABC]) -> None

Get1D(...)
    Get1D(arr, idx) -> CvScalar

```

```

Get2D(...)
    Get2D(arr, idx0, idx1) -> CvScalar

Get3D(...)
    Get3D(arr, idx0, idx1, idx2) -> CvScalar

GetAffineTransform(...)
    GetAffineTransform(src, dst, mapMatrix) -> None

GetCaptureProperty(...)
    GetCaptureProperty(capture, property_id) -> double

GetCentralMoment(...)
    GetCentralMoment(moments, x_order, y_order) -> double

GetCol(...)
    GetCol(arr, col) -> submat

GetCols(...)
    GetCols(arr, startCol, endCol) -> submat

GetDiag(...)
    GetDiag(arr [, diag]) -> submat

GetDims(...)
    GetDims(arr) -> dim1,dim2,...

GetElemType(...)
    GetElemType(arr) -> int

GetHuMoments(...)
    GetHuMoments(moments) -> hu

GetImage(...)
    GetImage(arr) -> IplImage

GetImageCOI(...)
    GetImageCOI(image) -> int

GetImageROI(...)
    GetImageROI(image) -> CvRect

GetMat(...)
    GetMat(arr [, allowND]) -> CvMat

GetMinMaxHistValue(...)
    GetMinMaxHistValue(hist) -> min_value,max_value,min_idx,max_idx

GetND(...)
    GetND(arr, indices) -> CvScalar

```

```

GetNormalizedCentralMoment(...)
    GetNormalizedCentralMoment(moments, x_order, y_order) -> double

GetOptimalDFTSize(...)
    GetOptimalDFTSize(size0) -> int

GetOptimalNewCameraMatrix(...)
    GetOptimalNewCameraMatrix(cameraMatrix, distCoeffs, imageSize, alpha,
newCameraMatrix [, newImageSize [, validPixROI [, centerPrincipalPoint]]]) -> None

GetPerspectiveTransform(...)
    GetPerspectiveTransform(src, dst, mapMatrix) -> None

GetQuadrangleSubPix(...)
    GetQuadrangleSubPix(src, dst, mapMatrix) -> None

GetReal1D(...)
    GetReal1D(arr, idx0) -> double

GetReal2D(...)
    GetReal2D(arr, idx0, idx1) -> double

GetReal3D(...)
    GetReal3D(arr, idx0, idx1, idx2) -> double

GetRealND(...)
    GetRealND(arr, idx) -> double

GetRectSubPix(...)
    GetRectSubPix(src, dst, center) -> None

GetRotationMatrix2D(...)
    GetRotationMatrix2D(center, angle, scale, mapMatrix) -> None

GetRow(...)
    GetRow(arr, row) -> submat

GetRows(...)
    GetRows(arr, startRow, endRow [, deltaRow]) -> submat

GetSize(...)
    GetSize(arr) -> CvSize

GetSpatialMoment(...)
    GetSpatialMoment(moments, x_order, y_order) -> double

GetStarKeypoints(...)
    GetStarKeypoints(image, storage [, params]) -> CvSeqOfCvStarKeypoint*

GetSubRect(...)
    GetSubRect(arr, rect) -> submat

```

```

GetTextSize(...)
    GetTextSize(textString, font) -> textSize,baseline

GetTickCount(...)
    GetTickCount() -> int64

GetTickFrequency(...)
    GetTickFrequency() -> int64

GetTrackbarPos(...)
    GetTrackbarPos(trackbarName, windowName) -> int

GetWindowProperty(...)
    GetWindowProperty(name, prop_id) -> double

GoodFeaturesToTrack(...)
    GoodFeaturesToTrack(image, eigImage, tempImage, cornerCount, qualityLevel,
minDistance [, mask [, blockSize [, useHarris [, k]]]]) -> cornerCount

GrabCut(...)
    GrabCut(image, mask, rect, bgdModel, fgdModel, iterCount, mode) -> None

GrabFrame(...)
    GrabFrame(capture) -> int

HOGDetectMultiScale(...)
    HOGDetectMultiScale(image, storage [, svm_classifier [, win_stride [,
hit_threshold [, scale [, group_threshold [, padding [, win_size [, block_size [,
block_stride [, cell_size [, nbins [, gammaCorrection]]]]]]]]]]]])) -> CvSeq*

HaarDetectObjects(...)
    HaarDetectObjects(image, cascade, storage [, scale_factor [, min_neighbors [,
flags [, min_size]]]]) -> CvSeqOfCvAvgComp*

HoughCircles(...)
    HoughCircles(image, circle_storage, method, dp, min_dist [, param1 [, param2 [,
min_radius [, max_radius]]]]) -> None

HoughLines2(...)
    HoughLines2(image, storage, method, rho, theta, threshold [, param1 [, param2]])
-> CvSeq*

IABS(...)
    CV_IABS(a) -> int

IS_SEQ_CLOSED(...)
    CV_IS_SEQ_CLOSED(s) -> int

IS_SEQ_CONVEX(...)
    CV_IS_SEQ_CONVEX(s) -> int

```

```

IS_SEQ_CURVE(...)
    CV_IS_SEQ_CURVE(s) -> int

IS_SEQ_HOLE(...)
    CV_IS_SEQ_HOLE(s) -> int

IS_SEQ_INDEX(...)
    CV_IS_SEQ_INDEX(s) -> int

IS_SEQ_SIMPLE(...)
    CV_IS_SEQ_SIMPLE(s) -> int

InRange(...)
    InRange(src, lower, upper, dst) -> None

InRangeS(...)
    InRangeS(src, lower, upper, dst) -> None

InitFont(...)
    InitFont(fontFace, hscale, vscale [, shear [, thickness [, lineHeight]]]) -> font

InitIntrinsicParams2D(...)
    InitIntrinsicParams2D(objectPoints, imagePoints, npoints, imageSize, cameraMatrix
[, aspectRatio]) -> None

InitLineIterator(...)
    InitLineIterator(image, pt1, pt2 [, connectivity [, left_to_right]]) ->
line_iterator

InitUndistortMap(...)
    InitUndistortMap(cameraMatrix, distCoeffs, map1, map2) -> None

InitUndistortRectifyMap(...)
    InitUndistortRectifyMap(cameraMatrix, distCoeffs, R, newCameraMatrix, map1, map2)
-> None

Inpaint(...)
    Inpaint(src, mask, dst, inpaintRadius, flags) -> None

Integral(...)
    Integral(image, sum [, sqsum [, tiltedSum]]) -> None

InvSqrt(...)
    InvSqrt(value) -> float

Invert(...)
    Invert(src, dst [, method]) -> double

IsInf(...)
    IsInf(value) -> int

```

```

IsNaN(...)
    IsNaN(value) -> int

KMeans2(...)
    KMeans2(samples, nclusters, labels, termcrit [, attempts [, flags [, centers]]])
-> double

KalmanCorrect(...)
    KalmanCorrect(kalman, measurement) -> ROCRMat*

KalmanPredict(...)
    KalmanPredict(kalman [, control]) -> ROCRMat*

LUT(...)
    LUT(src, dst, lut) -> None

Laplace(...)
    Laplace(src, dst [, apertureSize]) -> None

Line(...)
    Line(img, pt1, pt2, color [, thickness [, lineType [, shift]]]) -> None

Load(...)
    Load(filename [, storage [, name]]) -> generic

LoadImage(...)
    LoadImage(filename [, iscolor]) -> None

LoadImageM(...)
    LoadImageM(filename [, iscolor]) -> None

Log(...)
    Log(src, dst) -> None

LogPolar(...)
    LogPolar(src, dst, center, M [, flags]) -> None

MAKETYPE(...)
    CV_MAKETYPE(depth, cn) -> int

MAT_CN(...)
    CV_MAT_CN(i) -> int

MAT_DEPTH(...)
    CV_MAT_DEPTH(i) -> int

Mahalonobis(...)
    Mahalonobis(vec1, vec2, mat) -> None

MatMul(...)

```



```

    MatMul(src1, src2, dst) -> None

MatMulAdd(...)
    MatMulAdd(src1, src2, src3, dst) -> None

MatchShapes(...)
    MatchShapes(object1, object2, method [, parameter]) -> double

MatchTemplate(...)
    MatchTemplate(image, templ, result, method) -> None

Max(...)
    Max(src1, src2, dst) -> None

MaxRect(...)
    MaxRect(rect1, rect2) -> CvRect

MaxS(...)
    MaxS(src, value, dst) -> None

MeanShift(...)
    MeanShift(prob_image, window, criteria) -> comp

Merge(...)
    Merge(src0, src1, src2, src3, dst) -> None

Min(...)
    Min(src1, src2, dst) -> None

MinAreaRect2(...)
    MinAreaRect2(points [, storage]) -> CvBox2D

MinEnclosingCircle(...)
    MinEnclosingCircle(points) -> int,center,radius

MinMaxLoc(...)
    MinMaxLoc(arr [, mask]) -> minVal,maxVal,minLoc,maxLoc

MinS(...)
    MinS(src, value, dst) -> None

MixChannels(...)
    MixChannels(src, dst, fromTo) -> None

Moments(...)
    Moments(arr [, binary]) -> moments

MorphologyEx(...)
    MorphologyEx(src, dst, temp, element, operation [, iterations]) -> None

MoveWindow(...)

```

```

    MoveWindow(name, x, y) -> None

Mul(...)
    Mul(src1, src2, dst [, scale]) -> None

MulSpectrums(...)
    MulSpectrums(src1, src2, dst, flags) -> None

MulTransposed(...)
    MulTransposed(src, dst, order [, delta [, scale]]) -> None

MultiplyAcc(...)
    MultiplyAcc(image1, image2, acc [, mask]) -> None

NamedWindow(...)
    NamedWindow(name [, flags]) -> None

Norm(...)
    Norm(arr1, arr2 [, normType [, mask]]) -> double

Normalize(...)
    Normalize(src, dst [, a [, b [, norm_type [, mask]]]]) -> None

NormalizeHist(...)
    NormalizeHist(hist, factor) -> None

Not(...)
    Not(src, dst) -> None

Or(...)
    Or(src1, src2, dst [, mask]) -> None

OrS(...)
    OrS(src, value, dst [, mask]) -> None

POSIT(...)
    POSIT(posit_object, imagePoints, focal_length, criteria) ->
rotationMatrix,translation_vector

PerspectiveTransform(...)
    PerspectiveTransform(src, dst, mat) -> None

PointPolygonTest(...)
    PointPolygonTest(contour, pt, measure_dist) -> double

PolarToCart(...)
    PolarToCart(magnitude, angle, x, y [, angleInDegrees]) -> None

PolyLine(...)
    PolyLine(img, polys, is_closed, color [, thickness [, lineType [, shift]]]) ->
None

```

```

Pow(...)
    Pow(src, dst, power) -> None

PreCornerDetect(...)
    PreCornerDetect(image, corners [, apertureSize]) -> None

ProjectPCA(...)
    ProjectPCA(data, avg, eigenvectors, result) -> None

ProjectPoints2(...)
    ProjectPoints2(objectPoints, rvec, tvec, cameraMatrix, distCoeffs, imagePoints [,
dpdrot [, dpdt [, dpdf [, dpdc [, dpddist]]]]) -> None

PutText(...)
    PutText(img, text, org, font, color) -> None

PyrDown(...)
    PyrDown(src, dst [, filter]) -> None

PyrMeanShiftFiltering(...)
    PyrMeanShiftFiltering(src, dst, sp, sr [, max_level [, termcrit]]) -> None

PyrSegmentation(...)
    PyrSegmentation(src, dst, storage, level, threshold1, threshold2) -> comp

PyrUp(...)
    PyrUp(src, dst [, filter]) -> None

QueryFrame(...)
    QueryFrame(capture) -> ROIplImage*

QueryHistValue_1D(...)
    QueryHistValue_1D(hist, idx0) -> double

QueryHistValue_2D(...)
    QueryHistValue_2D(hist, idx0, idx1) -> double

QueryHistValue_3D(...)
    QueryHistValue_3D(hist, idx0, idx1, idx2) -> double

QueryHistValue_nD(...)
    QueryHistValue_nD(hist, idx) -> double

RGB(...)
    CV_RGB(red, grn, blu) -> CvScalar

RNG(...)
    RNG( [, seed]) -> CvRNG

RQDecomp3x3(...)

```

```

    RQDecomp3x3(M, R, Q [, Qx [, Qy [, Qz]]) -> eulerAngles

RandArr(...)
    RandArr(rng, arr, distType, param1, param2) -> None

RandInt(...)
    RandInt(rng) -> unsigned

RandReal(...)
    RandReal(rng) -> double

RandShuffle(...)
    RandShuffle(mat, rng [, iter_factor]) -> None

Range(...)
    Range(mat, start, end) -> None

RealScalar(...)
    RealScalar(val0) -> CvScalar

Rectangle(...)
    Rectangle(img, pt1, pt2, color [, thickness [, lineType [, shift]]) -> None

Reduce(...)
    Reduce(src, dst [, dim [, op]]) -> None

Remap(...)
    Remap(src, dst, mapx, mapy [, flags [, fillval]]) -> None

Repeat(...)
    Repeat(src, dst) -> None

ReprojectImageTo3D(...)
    ReprojectImageTo3D(disparity, _3dImage, Q [, handleMissingValues]) -> None

ResetImageROI(...)
    ResetImageROI(image) -> None

Reshape(...)
    Reshape(arr, newCn [, newRows]) -> CvMat

ReshapeMatND(...)
    ReshapeMatND(arr, newCn, newDims) -> CvMat

Resize(...)
    Resize(src, dst [, interpolation]) -> None

ResizeWindow(...)
    ResizeWindow(name, width, height) -> None

RetrieveFrame(...)

```

```

RetrieveFrame(capture [, index]) -> ROIplImage*

Rodrigues2(...)
  Rodrigues2(src, dst [, jacobian]) -> None

Round(...)
  Round(value) -> int

RunningAvg(...)
  RunningAvg(image, acc, alpha [, mask]) -> None

SIGN(...)
  CV_SIGN(a) -> int

SVBkSb(...)
  SVBkSb(W, U, V, B, X, flags) -> None

SVD(...)
  SVD(A, W [, U [, V [, flags]]]) -> None

Save(...)
  Save(filename, structPtr [, name [, comment]]) -> None

SaveImage(...)
  SaveImage(filename, image) -> None

Scalar(...)
  Scalar(val0 [, val1 [, val2 [, val3]]) -> CvScalar

ScalarAll(...)
  ScalarAll(val0123) -> CvScalar

Scale(...)
  Scale(src, dst [, scale [, shift]]) -> None

ScaleAdd(...)
  ScaleAdd(src1, scale, src2, dst) -> None

SegmentMotion(...)
  SegmentMotion(mhi, seg_mask, storage, timestamp, seg_thresh) -> CvSeq*

SeqInvert(...)
  SeqInvert(seq) -> None

SeqRemove(...)
  SeqRemove(seq, index) -> None

SeqRemoveSlice(...)
  SeqRemoveSlice(seq, slice) -> None

Set(...)

```

```

    Set(arr, value [, mask]) -> None

Set1D(...)
    Set1D(arr, idx, value) -> None

Set2D(...)
    Set2D(arr, idx0, idx1, value) -> None

Set3D(...)
    Set3D(arr, idx0, idx1, idx2, value) -> None

SetCaptureProperty(...)
    SetCaptureProperty(capture, property_id, value) -> int

SetData(...)
    SetData(arr, data, step) -> None

SetIdentity(...)
    SetIdentity(mat [, value]) -> None

SetImageCOI(...)
    SetImageCOI(image, coi) -> None

SetImageROI(...)
    SetImageROI(image, rect) -> None

SetMouseCallback(...)
    SetMouseCallback(windowName, onMouse [, param]) -> None

SetND(...)
    SetND(arr, indices, value) -> None

SetReal1D(...)
    SetReal1D(arr, idx, value) -> None

SetReal2D(...)
    SetReal2D(arr, idx0, idx1, value) -> None

SetReal3D(...)
    SetReal3D(arr, idx0, idx1, idx2, value) -> None

SetRealND(...)
    SetRealND(arr, indices, value) -> None

SetTrackbarPos(...)
    SetTrackbarPos(trackbarName, windowName, pos) -> None

SetWindowProperty(...)
    SetWindowProperty(name, prop_id, prop_value) -> None

SetZero(...)

```

```

SetZero(arr) -> None

ShowImage(...)
    ShowImage(name, image) -> None

Smooth(...)
    Smooth(src, dst [, smoothtype [, param1 [, param2 [, param3 [, param4]]]]) ->
None

SnakeImage(...)
    SnakeImage(image, points, alpha, beta, gamma, win, criteria [, calc_gradient]) ->
points

Sobel(...)
    Sobel(src, dst, xorder, yorder [, apertureSize]) -> None

Solve(...)
    Solve(A, B, X [, method]) -> None

SolveCubic(...)
    SolveCubic(coeffs, roots) -> None

SolvePoly(...)
    SolvePoly(coeffs, roots [, maxiter [, fig]]) -> None

Sort(...)
    Sort(src, dst, idxmat [, flags]) -> None

Split(...)
    Split(src, dst0, dst1, dst2, dst3) -> None

Sqrt(...)
    Sqrt(value) -> float

SquareAcc(...)
    SquareAcc(image, sqsum [, mask]) -> None

StartWindowThread(...)
    StartWindowThread() -> None

StereoCalibrate(...)
    StereoCalibrate(objectPoints, imagePoints1, imagePoints2, pointCounts,
cameraMatrix1, distCoeffs1, cameraMatrix2, distCoeffs2, imageSize, R, T [, E [, F [,
term_crit [, flags]]]]) -> None

StereoRectify(...)
    StereoRectify(cameraMatrix1, cameraMatrix2, distCoeffs1, distCoeffs2, imageSize,
R, T, R1, R2, P1, P2 [, Q [, flags [, alpha [, newImageSize]]]]) -> roi1,roi2

StereoRectifyUncalibrated(...)
    StereoRectifyUncalibrated(points1, points2, F, imageSize, H1, H2 [, threshold]) ->

```

None

```
Sub(...)
    Sub(src1, src2, dst [, mask]) -> None

SubRS(...)
    SubRS(src, value, dst [, mask]) -> None

SubS(...)
    SubS(src, value, dst [, mask]) -> None

Subdiv2DEdgeDst(...)
    Subdiv2DEdgeDst(edge) -> CvSubdiv2DPoint*

Subdiv2DEdgeOrg(...)
    Subdiv2DEdgeOrg(edge) -> CvSubdiv2DPoint*

Subdiv2DGetEdge(...)
    Subdiv2DGetEdge(edge, type) -> CvSubdiv2DEdge

Subdiv2DLocate(...)
    Subdiv2DLocate(subdiv, pt) -> loc,where

Subdiv2DNextEdge(...)
    Subdiv2DNextEdge(edge) -> CvSubdiv2DEdge

Subdiv2DRotateEdge(...)
    Subdiv2DRotateEdge(edge, rotate) -> CvSubdiv2DEdge

SubdivDelaunay2DInsert(...)
    SubdivDelaunay2DInsert(subdiv, pt) -> CvSubdiv2DPoint*

Sum(...)
    Sum(arr) -> CvScalar

ThreshHist(...)
    ThreshHist(hist, threshold) -> None

Threshold(...)
    Threshold(src, dst, threshold, maxValue, thresholdType) -> None

Trace(...)
    Trace(mat) -> CvScalar

Transform(...)
    Transform(src, dst, transmat [, shiftvec]) -> None

Transpose(...)
    Transpose(src, dst) -> None

Undistort2(...)
```



```

    Undistort2(src, dst, cameraMatrix, distCoeffs) -> None

UndistortPoints(...)
    UndistortPoints(src, dst, cameraMatrix, distCoeffs [, R [, P]]) -> None

UpdateMotionHistory(...)
    UpdateMotionHistory(silhouette, mhi, timestamp, duration) -> None

WaitKey(...)
    WaitKey( [, delay]) -> int

WarpAffine(...)
    WarpAffine(src, dst, mapMatrix [, flags [, fillval]]) -> None

WarpPerspective(...)
    WarpPerspective(src, dst, mapMatrix [, flags [, fillval]]) -> None

Watershed(...)
    Watershed(image, markers) -> None

WriteFrame(...)
    WriteFrame(writer, image) -> int

Xor(...)
    Xor(src1, src2, dst [, mask]) -> None

XorS(...)
    XorS(src, value, dst [, mask]) -> None

Zero(...)
    Zero(arr) -> None

fromarray(...)
    fromarray(array [, allowND]) -> CvMat

mGet(...)
    mGet(mat, row, col) -> double

mSet(...)
    mSet(mat, row, col, value) -> None

temp_test(...)

```

#### DATA

```

CV_16S = 3
CV_16SC1 = 3
CV_16SC2 = 11
CV_16SC3 = 19
CV_16SC4 = 27
CV_16U = 2
CV_16UC1 = 2

```

CV\_16UC2 = 10  
CV\_16UC3 = 18  
CV\_16UC4 = 26  
CV\_32F = 5  
CV\_32FC1 = 5  
CV\_32FC2 = 13  
CV\_32FC3 = 21  
CV\_32FC4 = 29  
CV\_32S = 4  
CV\_32SC1 = 4  
CV\_32SC2 = 12  
CV\_32SC3 = 20  
CV\_32SC4 = 28  
CV\_64F = 6  
CV\_64FC1 = 6  
CV\_64FC2 = 14  
CV\_64FC3 = 22  
CV\_64FC4 = 30  
CV\_8S = 1  
CV\_8SC1 = 1  
CV\_8SC2 = 9  
CV\_8SC3 = 17  
CV\_8SC4 = 25  
CV\_8U = 0  
CV\_8UC1 = 0  
CV\_8UC2 = 8  
CV\_8UC3 = 16  
CV\_8UC4 = 24  
CV\_AA = 16  
CV\_ADAPTIVE\_THRESH\_GAUSSIAN\_C = 1  
CV\_ADAPTIVE\_THRESH\_MEAN\_C = 0  
CV\_AUTOSTEP = 2147483647  
CV\_BACK = 0  
CV\_BGR2BGR555 = 22  
CV\_BGR2BGR565 = 12  
CV\_BGR2BGRA = 0  
CV\_BGR2GRAY = 6  
CV\_BGR2HLS = 52  
CV\_BGR2HSV = 40  
CV\_BGR2Lab = 44  
CV\_BGR2Luv = 50  
CV\_BGR2RGB = 4  
CV\_BGR2RGBA = 2  
CV\_BGR2XYZ = 32  
CV\_BGR2YCrCb = 36  
CV\_BGR5552BGR = 24  
CV\_BGR5552BGRA = 28  
CV\_BGR5552GRAY = 31  
CV\_BGR5552RGB = 25  
CV\_BGR5552RGBA = 29  
CV\_BGR5652BGR = 14

CV\_BGR5652BGRA = 18  
CV\_BGR5652GRAY = 21  
CV\_BGR5652RGB = 15  
CV\_BGR5652RGBA = 19  
CV\_BGRA2BGR = 1  
CV\_BGRA2BGR555 = 26  
CV\_BGRA2BGR565 = 16  
CV\_BGRA2GRAY = 10  
CV\_BGRA2RGB = 3  
CV\_BGRA2RGBA = 5  
CV\_BILATERAL = 4  
CV\_BLUR = 1  
CV\_BLUR\_NO\_SCALE = 0  
CV\_BayerBG2BGR = 46  
CV\_BayerBG2BGR\_VNG = 62  
CV\_BayerBG2RGB = 48  
CV\_BayerGB2BGR = 47  
CV\_BayerGB2BGR\_VNG = 63  
CV\_BayerGB2RGB = 49  
CV\_BayerGR2BGR = 49  
CV\_BayerGR2BGR\_VNG = 65  
CV\_BayerGR2RGB = 47  
CV\_BayerRG2BGR = 48  
CV\_BayerRG2BGR\_VNG = 64  
CV\_BayerRG2RGB = 46  
CV\_C = 1  
CV\_CALIB\_CB\_ADAPTIVE\_THRESH = 1  
CV\_CALIB\_CB\_FILTER\_QUADS = 4  
CV\_CALIB\_CB\_NORMALIZE\_IMAGE = 2  
CV\_CALIB\_FIX\_ASPECT\_RATIO = 2  
CV\_CALIB\_FIX\_FOCAL\_LENGTH = 16  
CV\_CALIB\_FIX\_INTRINSIC = 256  
CV\_CALIB\_FIX\_K1 = 32  
CV\_CALIB\_FIX\_K2 = 64  
CV\_CALIB\_FIX\_K3 = 128  
CV\_CALIB\_FIX\_PRINCIPAL\_POINT = 4  
CV\_CALIB\_SAME\_FOCAL\_LENGTH = 512  
CV\_CALIB\_USE\_INTRINSIC\_GUESS = 1  
CV\_CALIB\_ZERO\_DISPARITY = 1024  
CV\_CALIB\_ZERO\_TANGENT\_DIST = 8  
CV\_CANNY\_L2\_GRADIENT = -2147483648  
CV\_CAP\_OPENNI = 900  
CV\_CAP\_OPENNI\_BGR\_IMAGE = 5  
CV\_CAP\_OPENNI\_DEPTH\_GENERATOR = -2147483648  
CV\_CAP\_OPENNI\_DEPTH\_GENERATOR\_BASELINE = -2147483546  
CV\_CAP\_OPENNI\_DEPTH\_GENERATOR\_FOCAL\_LENGTH = -2147483545  
CV\_CAP\_OPENNI\_DEPTH\_GENERATOR\_REGISTRATION = -2147483544  
CV\_CAP\_OPENNI\_DEPTH\_MAP = 0  
CV\_CAP\_OPENNI\_DISPARITY\_MAP = 2  
CV\_CAP\_OPENNI\_DISPARITY\_MAP\_32F = 3  
CV\_CAP\_OPENNI\_GRAY\_IMAGE = 6

CV\_CAP\_OPENNI\_IMAGE\_GENERATOR = 1073741824  
CV\_CAP\_OPENNI\_IMAGE\_GENERATOR\_OUTPUT\_MODE = 1073741924  
CV\_CAP\_OPENNI\_POINT\_CLOUD\_MAP = 1  
CV\_CAP\_OPENNI\_SXGA\_15HZ = 1  
CV\_CAP\_OPENNI\_VALID\_DEPTH\_MASK = 4  
CV\_CAP\_OPENNI\_VGA\_30HZ = 0  
CV\_CAP\_PROP\_BRIGHTNESS = 10  
CV\_CAP\_PROP\_CONTRAST = 11  
CV\_CAP\_PROP\_CONVERT\_RGB = 16  
CV\_CAP\_PROP\_EXPOSURE = 15  
CV\_CAP\_PROP\_FORMAT = 8  
CV\_CAP\_PROP\_FOURCC = 6  
CV\_CAP\_PROP\_FPS = 5  
CV\_CAP\_PROP\_FRAME\_COUNT = 7  
CV\_CAP\_PROP\_FRAME\_HEIGHT = 4  
CV\_CAP\_PROP\_FRAME\_WIDTH = 3  
CV\_CAP\_PROP\_GAIN = 14  
CV\_CAP\_PROP\_HUE = 13  
CV\_CAP\_PROP\_MODE = 9  
CV\_CAP\_PROP\_OPENNI\_BASELINE = 102  
CV\_CAP\_PROP\_OPENNI\_FOCAL\_LENGTH = 103  
CV\_CAP\_PROP\_OPENNI\_FRAME\_MAX\_DEPTH = 101  
CV\_CAP\_PROP\_OPENNI\_OUTPUT\_MODE = 100  
CV\_CAP\_PROP\_OPENNI\_REGISTRATION = 104  
CV\_CAP\_PROP\_POS\_AVI\_RATIO = 2  
CV\_CAP\_PROP\_POS\_FRAMES = 1  
CV\_CAP\_PROP\_POS\_MSEC = 0  
CV\_CAP\_PROP\_RECTIFICATION = 18  
CV\_CAP\_PROP\_SATURATION = 12  
CV\_CHAIN\_APPROX\_NONE = 1  
CV\_CHAIN\_APPROX\_SIMPLE = 2  
CV\_CHAIN\_APPROX\_TC89\_KCOS = 4  
CV\_CHAIN\_APPROX\_TC89\_L1 = 3  
CV\_CHAIN\_CODE = 0  
CV\_CHECK\_QUIET = 2  
CV\_CHECK\_RANGE = 1  
CV\_CHOLESKY = 3  
CV\_CLOCKWISE = 1  
CV\_CMP\_EQ = 0  
CV\_CMP\_GE = 2  
CV\_CMP\_GT = 1  
CV\_CMP\_LE = 4  
CV\_CMP\_LT = 3  
CV\_CMP\_NE = 5  
CV\_CN\_MAX = 512  
CV\_CN\_SHIFT = 3  
CV\_COLORCVT\_MAX = 135  
CV\_COMP\_BHATTACHARYYA = 3  
CV\_COMP\_CHISQR = 1  
CV\_COMP\_CORREL = 0  
CV\_COMP\_INTERSECT = 2

```
CV_CONTOURS_MATCH_I1 = 1
CV_CONTOURS_MATCH_I2 = 2
CV_CONTOURS_MATCH_I3 = 3
CV_COUNTER_CLOCKWISE = 2
CV_COVAR_COLS = 16
CV_COVAR_NORMAL = 1
CV_COVAR_ROWS = 8
CV_COVAR_SCALE = 4
CV_COVAR_SCRAMBLED = 0
CV_COVAR_USE_AVG = 2
CV_CVTIMG_FLIP = 1
CV_CVTIMG_SWAP_RB = 2
CV_DIFF = 16
CV_DIFF_C = 17
CV_DIFF_L1 = 18
CV_DIFF_L2 = 20
CV_DIST_C = 3
CV_DIST_FAIR = 5
CV_DIST_HUBER = 7
CV_DIST_L1 = 1
CV_DIST_L12 = 4
CV_DIST_L2 = 2
CV_DIST_MASK_3 = 3
CV_DIST_MASK_5 = 5
CV_DIST_MASK_PRECISE = 0
CV_DIST_USER = -1
CV_DIST_WELSCH = 6
CV_DXT_FORWARD = 0
CV_DXT_INVERSE = 1
CV_DXT_INVERSE_SCALE = 3
CV_DXT_INV_SCALE = 3
CV_DXT_MUL_CONJ = 8
CV_DXT_ROWS = 4
CV_DXT_SCALE = 2
CV_EVENT_FLAG_ALTKEY = 32
CV_EVENT_FLAG_CTRLKEY = 8
CV_EVENT_FLAG_LBUTTON = 1
CV_EVENT_FLAG_MBUTTON = 4
CV_EVENT_FLAG_RBUTTON = 2
CV_EVENT_FLAG_SHIFTKEY = 16
CV_EVENT_LBUTTONDOWNCLK = 7
CV_EVENT_LBUTTONDOWN = 1
CV_EVENT_LBUTTONUP = 4
CV_EVENT_MBUTTONDOWNCLK = 9
CV_EVENT_MBUTTONDOWN = 3
CV_EVENT_MBUTTONUP = 6
CV_EVENT_MOUSEMOVE = 0
CV_EVENT_RBUTTONDOWNCLK = 8
CV_EVENT_RBUTTONDOWN = 2
CV_EVENT_RBUTTONUP = 5
CV_ErrModeLeaf = 0
```

```
CV_ErrModeParent = 1
CV_ErrModeSilent = 2
CV_FILLED = -1
CV_FLOODFILL_FIXED_RANGE = 65536
CV_FLOODFILL_MASK_ONLY = 131072
CV_FM_7POINT = 1
CV_FM_8POINT = 2
CV_FM_LMEDS = 4
CV_FM_LMEDS_ONLY = 4
CV_FM_RANSAC = 8
CV_FM_RANSAC_ONLY = 8
CV_FONT_HERSHEY_COMPLEX = 3
CV_FONT_HERSHEY_COMPLEX_SMALL = 5
CV_FONT_HERSHEY_DUPLEX = 2
CV_FONT_HERSHEY_PLAIN = 1
CV_FONT_HERSHEY_SCRIPT_COMPLEX = 7
CV_FONT_HERSHEY_SCRIPT_SIMPLEX = 6
CV_FONT_HERSHEY_SIMPLEX = 0
CV_FONT_HERSHEY_TRIPLEX = 4
CV_FONT_ITALIC = 16
CV_FONT_VECTORO = 0
CV_FRONT = 1
CV_GAUSSIAN = 2
CV_GAUSSIAN_5x5 = 7
CV_GEMM_A_T = 1
CV_GEMM_B_T = 2
CV_GEMM_C_T = 4
CV_GRAPH_ALL_ITEMS = -1
CV_GRAPH_ANY_EDGE = 30
CV_GRAPH_BACKTRACKING = 64
CV_GRAPH_BACK_EDGE = 4
CV_GRAPH_CROSS_EDGE = 16
CV_GRAPH_FORWARD_EDGE = 8
CV_GRAPH_FORWARD_EDGE_FLAG = 268435456
CV_GRAPH_ITEM_VISITED_FLAG = 1073741824
CV_GRAPH_NEW_TREE = 32
CV_GRAPH_OVER = -1
CV_GRAPH_SEARCH_TREE_NODE_FLAG = 536870912
CV_GRAPH_TREE_EDGE = 2
CV_GRAPH_VERTEX = 1
CV_GRAY2BGR = 8
CV_GRAY2BGR555 = 30
CV_GRAY2BGR565 = 20
CV_GRAY2BGRA = 9
CV_GRAY2RGB = 8
CV_GRAY2RGBA = 9
CV_HAAR_DO_CANNY_PRUNING = 1
CV_HAAR_DO_ROUGH_SEARCH = 8
CV_HAAR_FEATURE_MAX = 3
CV_HAAR_FIND_BIGGEST_OBJECT = 4
CV_HAAR_MAGIC_VAL = 1112539136
```

```
CV_HAAR_SCALE_IMAGE = 2
CV_HIST_ARRAY = 0
CV_HIST_SPARSE = 1
CV_HLS2BGR = 60
CV_HLS2RGB = 61
CV_HOUGH_GRADIENT = 3
CV_HOUGH_MULTI_SCALE = 2
CV_HOUGH_PROBABILISTIC = 1
CV_HOUGH_STANDARD = 0
CV_HSV2BGR = 54
CV_HSV2RGB = 55
CV_IMWRITE_JPEG_QUALITY = 1
CV_IMWRITE_PNG_COMPRESSION = 16
CV_IMWRITE_PXM_BINARY = 32
CV_INPAINT_NS = 0
CV_INPAINT_TELEA = 1
CV_INTER_AREA = 3
CV_INTER_CUBIC = 2
CV_INTER_LINEAR = 1
CV_INTER_NN = 0
CV_KMEANS_USE_INITIAL_LABELS = 1
CV_L1 = 2
CV_L2 = 4
CV_LINK_RUNS = 5
CV_LKFLOW_GET_MIN_EIGENVALS = 8
CV_LKFLOW_INITIAL_GUESSES = 4
CV_LKFLOW_PYR_A_READY = 1
CV_LKFLOW_PYR_B_READY = 2
CV_LMEDS = 4
CV_LOAD_IMAGE_COLOR = 1
CV_LOAD_IMAGE_GRAYSCALE = 0
CV_LOAD_IMAGE_UNCHANGED = -1
CV_LU = 0
CV_Lab2BGR = 56
CV_Lab2RGB = 57
CV_Luv2BGR = 58
CV_Luv2RGB = 59
CV_MAX_ARR = 10
CV_MAX_DIM = 32
CV_MAX_SOBEL_KSIZE = 7
CV_MEDIAN = 3
CV_MINMAX = 32
CV_MOP_BLACKHAT = 6
CV_MOP_CLOSE = 3
CV_MOP_GRADIENT = 4
CV_MOP_OPEN = 2
CV_MOP_TOPHAT = 5
CV_NEXT_AROUND_DST = 34
CV_NEXT_AROUND_LEFT = 19
CV_NEXT_AROUND_ORG = 0
CV_NEXT_AROUND_RIGHT = 49
```

CV\_NORMAL = 16  
CV\_NORM\_MASK = 7  
CV\_NO\_CN\_CHECK = 2  
CV\_NO\_DEPTH\_CHECK = 1  
CV\_NO\_SIZE\_CHECK = 4  
CV\_PCA\_DATA\_AS\_COL = 1  
CV\_PCA\_DATA\_AS\_ROW = 0  
CV\_PCA\_USE\_AVG = 2  
CV\_PI = 3.141592653589793  
CV\_POLY\_APPROX\_DP = 0  
CV\_PREV\_AROUND\_DST = 51  
CV\_PREV\_AROUND\_LEFT = 32  
CV\_PREV\_AROUND\_ORG = 17  
CV\_PREV\_AROUND\_RIGHT = 2  
CV\_PTLOC\_INSIDE = 0  
CV\_PTLOC\_ON\_EDGE = 2  
CV\_PTLOC\_OUTSIDE\_RECT = -1  
CV\_PTLOC\_VERTEX = 1  
CV\_QR = 4  
CV\_RAND\_NORMAL = 1  
CV\_RAND\_UNI = 0  
CV\_RANSAC = 8  
CV\_REDUCE\_AVG = 1  
CV\_REDUCE\_MAX = 2  
CV\_REDUCE\_MIN = 3  
CV\_REDUCE\_SUM = 0  
CV\_RELATIVE = 8  
CV\_RELATIVE\_C = 9  
CV\_RELATIVE\_L1 = 10  
CV\_RELATIVE\_L2 = 12  
CV\_RETR\_CCOMP = 2  
CV\_RETR\_EXTERNAL = 0  
CV\_RETR\_LIST = 1  
CV\_RETR\_TREE = 3  
CV\_RGB2BGR = 4  
CV\_RGB2BGR555 = 23  
CV\_RGB2BGR565 = 13  
CV\_RGB2BGRA = 2  
CV\_RGB2GRAY = 7  
CV\_RGB2HLS = 53  
CV\_RGB2HSV = 41  
CV\_RGB2Lab = 45  
CV\_RGB2Luv = 51  
CV\_RGB2RGBA = 0  
CV\_RGB2XYZ = 33  
CV\_RGB2YCrCb = 37  
CV\_RGBA2BGR = 3  
CV\_RGBA2BGR555 = 27  
CV\_RGBA2BGR565 = 17  
CV\_RGBA2BGRA = 5  
CV\_RGBA2GRAY = 11



```
CV_RGBA2RGB = 1
CV_SCHARR = -1
CV_SHAPE_CROSS = 1
CV_SHAPE_CUSTOM = 100
CV_SHAPE_ELLIPSE = 2
CV_SHAPE_RECT = 0
CV_SORT_ASCENDING = 0
CV_SORT_DESCENDING = 16
CV_SORT_EVERY_COLUMN = 1
CV_SORT_EVERY_ROW = 0
CV_STEREO_BM_BASIC = 0
CV_STEREO_BM_FISH_EYE = 1
CV_STEREO_BM_NARROW = 2
CV_STEREO_BM_NORMALIZED_RESPONSE = 0
CV_STEREO_GC_OCCLUDED = 32767
CV_SUBDIV2D_VIRTUAL_POINT_FLAG = 1073741824
CV_SVD = 1
CV_SVD_MODIFY_A = 1
CV_SVD_SYM = 2
CV_SVD_U_T = 2
CV_SVD_V_T = 4
CV_TERMCRT_EPS = 2
CV_TERMCRT_ITER = 1
CV_TERMCRT_NUMBER = 1
CV_THRESH_BINARY = 0
CV_THRESH_BINARY_INV = 1
CV_THRESH_MASK = 7
CV_THRESH_OTSU = 8
CV_THRESH_TOZERO = 3
CV_THRESH_TOZERO_INV = 4
CV_THRESH_TRUNC = 2
CV_TM_CCOEFF = 4
CV_TM_CCOEFF_NORMED = 5
CV_TM_CCORR = 2
CV_TM_CCORR_NORMED = 3
CV_TM_SQDIFF = 0
CV_TM_SQDIFF_NORMED = 1
CV_WARP_FILL_OUTLIERS = 8
CV_WARP_INVERSE_MAP = 16
CV_WINDOW_AUTOSIZE = 1
CV_WINDOW_FULLSCREEN = 1
CV_WINDOW_NORMAL = 0
CV_XYZ2BGR = 34
CV_XYZ2RGB = 35
CV_YCrCb2BGR = 38
CV_YCrCb2RGB = 39
GC_BGD = 0
GC_EVAL = 2
GC_FGD = 1
GC_INIT_WITH_MASK = 1
GC_INIT_WITH_RECT = 0
```

GC\_PR\_BGD = 2  
GC\_PR\_FGD = 3  
HG\_AUTOSIZE = 1  
IPL\_DEPTH\_16S = 2147483664L  
IPL\_DEPTH\_16U = 16L  
IPL\_DEPTH\_32F = 32L  
IPL\_DEPTH\_32S = 2147483680L  
IPL\_DEPTH\_64F = 64L  
IPL\_DEPTH\_8S = 2147483656L  
IPL\_DEPTH\_8U = 8L  
IPL\_ORIGIN\_BL = 1  
IPL\_ORIGIN\_TL = 0