Content

void calcHist()……………………. 1

void namedWindow()………… 2

int createTrackbar()……………. 3

void calcHist( const Mat\* images, int nimages,

const int\* channels, InputArray mask,

OutputArray hist, int dims, const int\* histSize,

const float\*\* ranges, bool uniform = true, bool accumulate = false );

函數功能:

建立直方圖

參數說明:

const Mat\* images: 原輸入(圖片)陣列

int nimages: 原輸入陣列元素個數(指幾張圖)

const int\* channels: 用來計算直方圖的通道維數陣列, [0]:灰階, [0, 1, 2]:RGB三色

InputArray mask: 若非空(Mat()), 則必須與images的大小相等, 且僅計算mask

中非零的部分

OutputArray hist: 輸出直方圖

int dims: 直方圖維度

const int\* histSize: 指出直方圖陣列每一維bin大小的陣列

const float\*\* ranges: 用來指出直方圖每一維的每個bin上下界範圍陣列的陣列

bool uniform = true: 直方圖是否均勻

bool accumulate = false: 是否累加

void namedWindow(const string& winname,

int flags=WINDOW\_AUTOSIZE )

函數功能:

建立一個視窗

參數說明:

const string& winname: 視窗名稱, 可當成視窗ID使用

int flags=WINDOW\_AUTOSIZE: 常用的2個設定值: WINDOW\_NORMAL:使用

者可自行縮放視窗大小, WINDOW\_AUTOSIZE: 自動調整成適合螢幕大

小, 不能自行縮放

int createTrackbar (conststring& trackbarname,

conststring& winname,

int\* value, int count,

TrackbarCallback onChange=0, void\* userdata=0);

函數功能:

建立一個滑桿

參數說明:

conststring& trackbarname: 滑桿名稱, 用來代表我們創建的這個滑桿

conststring& winname: 視窗名稱, 表示這個滑桿將會附到哪個視窗上, 即對應

創建視窗namedWindow()時所設的某個視窗名稱

int\* value: 一個指向整數的指標, 表示滑桿位置

int count: 滑桿位置的最大值

TrackbarCallback onChange=0: 這是一個指向回調函數的指針, 預設值onchang=0,

每次滑桿改變位置時, 這個函數都會進行回調, 並且這個函數原型必須為

void XXXX(int , void\*), 其中第一個參數為滑桿位置, 第二個參數為用戶數據(看下一個 參數), 如果回調是NULL指針, 表示沒有回調函數的調用, 僅第3個參數value有變化

void\* userdata=0: void\*類型的userdata, 預設值=0, 這個參數是用戶傳給調用函數

的數據, 用來處理滑桿事件, 如果第3個參數是全局變數, 可以不用管這個參數