OpenCV - Bug # 62: Doubts about the tiltedSum in cvIntegral()

Status:	Cancelled	**Priority:**	Normal
Author:	Ryan Lei	**Category:**	imgproc, video
Created:	2010-01-10	**Assignee:**	
Updated:	2010-01-27	**Due date:**	
Affected version:			
Difficulty:			
Pull request:			
Operating System:			
HW Platform:			

Description:

Hi, I tried to use cvIntegral() today, and had some doubts about the tiltedSum result:
After studying the [[OpenCV]] online "docs":http://opencv.willowgarage.com/documentation/miscellaneous_image_transformations.html?highlight=cvintegral#cvIntegral and the "Learning [[OpenCV]] book":http://www.amazon.com/Learning-OpenCV-Computer-Vision-Library/dp/0596516134, I found out the definition of tiltedSum result is like a "triangular sum", starting from point (X, 0), going down 45 degrees to the left and right, but above the line y = Y.
A sample result goes like the attachment of summing an "ones" matrix.

However, in both the original "paper":http://en.wikipedia.org/wiki/Haar-like_features#Tilted_Haar-like_features:
Lienhart, R. and Maydt, J., "An extended set of Haar-like features for rapid object detection", ICIP02, pp. I: 900-903, 2002, and an "Intel document":http://www.scribd.com/doc/4547752/opencv-objectdetection-2007june10 I found on the Internet, the summing result is nothing like such an triangle. Instead, the sum starts at point (X,Y), going 45 degrees upper-left and lower-left.
In fact, the summation limit is quite different in [[OpenCV]] and the paper:
[[OpenCV]]: y <= Y, |x - X| <= y
Lienhart, and Maydt: x <= X, x <= X - |y - Y|

Is that a mistake? Because the tiltedSum here seems meaningless, and cv/cvsumpixels.cpp has not been updated for 7 months!

Associated revisions

2013-08-26 02:49 pm - Roman Donchenko

Merge pull request #62 from jet47:gpu-resize

History

2010-01-13 12:44 pm - anonymous -

- Status changed from Open to Done

- (deleted custom field) set to worksforme

This is not a mistake. The only purpose of the tilted integral image is to compute sums of pixels covered by tilted rectangles efficiently. Both forms of tilted integral images can be used for that. Indeed, it's just a matter of transposition.

[[OpenCV]] uses a different form, because such tilted integral can be computed simultaneously (within the same loop) with normal integral image, while the original form requires 2-pass algorithm over rows or 1-pass algorithm over columns (inefficient).

The author of the mentioned paper, R. Lienhart is aware of this different interpretation and approved it.

2010-01-15 09:48 am - Ryan Lei

- Status changed from Done to Cancelled

- (deleted custom field) deleted (worksforme)

Thanks for the explanation, and sorry for reopening the ticket to raise attention.
I am aware of the 2-pass performance issue in the original paper, but still, I can't figure out how to compute tilted pixel sums using 4 "plus, minus" operations this way.

As simple example, can any one tell me how to compute the tilted rectangle sum formed by the 4 points: (5,3), (3,5), (7,9), (9,7)? Or in Lienhart's notation:
x = 5, y = 3, w = 4, h = 2 ?

Here, the origin is at (0,0), x and w is horizontal, y and h is vertical.
Please answer assuming tiltedSum is of size W x H, not (W+1) x (H+1).

2010-01-15 12:10 pm - anonymous -

tiltedSum(x,y) - tiltedSum(x-h,y+h) - tiltedSum(x+w, y+w) + tiltedSum(x+w-h,y+w+h)

2010-01-16 10:42 am - Ryan Lei

I'm afraid this is not the case in [[OpenCV]]. I know in Lienhart's paper, the result is:
-tiltedSum(x,y) + tiltedSum(x-h,y+h) + tiltedSum(x+w, y+w) - tiltedSum(x+w-h,y+w+h), with the 4 signs opposite from yours.
I verified that the result of my formula is close (with a bit of errors) to a tilted rectangle sum.

However, if tiltedSum(x,y) in [[OpenCV]] is defined like the triangle sum in the previous attachment "cvIntegral.png", then using your formula does not result in a tilted rectangle sum at all, not even close.

Please see my computations in the three pictures in the attachment. The big points are the 4 corners. The green ones represent the '+' terms, red ones the '-' terms. The small points in "summing.png" show the summing process. And finally, the cancellation result is in "result.png". So I can see the final result is FAR from what we intended. In fact, points within the tilted rectangle rarely appear in the result, but so many unwanted points do.

That's my original doubts.

Files

cvIntegral.png	43.9 kB	2010-01-10	Ryan Lei
4_points.png	56.3 kB	2010-01-16	Ryan Lei
summing.png	61.4 kB	2010-01-16	Ryan Lei
result.png	61.4 kB	2010-01-16	Ryan Lei