PARALLELIZED 45 DEGREES ROTATED IMAGE INTEGRATION

M. Schweitzer, H. Jaspers, T. Kubertschak, H.-J. Wuensch
Universität der Bundeswehr München
Technik Autonomer Systeme (TAS)
85577 Neubiberg

ABSTRACT
In [1] Lienhart and Maydt introduced the calculation of rotated Haar-Wavelets by $45^\circ$. They showed that the extended set of possible wavelets improves the object recognition method of Viola et al. [2]. In this paper, we introduce a novel integral image structure which holds the information of a standard and a $45^\circ$ rotated integral image. We use this image structure to improve a simple and efficient corner detector by Schweitzer and Wuensche [3] based on Haar wavelets in terms of rotational invariance and define an orientation function. Additionally, we focus on parallelization and remove the recursive character of [1] to make the method suitable for GPUs.

Index Terms— Integral Image, Feature Extraction, GPU

1. INTRODUCTION
Integral images are widely used in computer vision algorithms because they provide an extremely efficient way of calculating the sum of intensities within an image box independently of its size (see figure 1). Mainly, this has a great impact on implementing scale invariant algorithms like feature extraction, descriptor calculation [4, 5] or object recognition [1, 2] because the complexity remains at $O(1)$ when searching in scale space. On the other side, it is hard to achieve robustness against image rotation by the fact that rotated boxes are not covered by the $O(1)$-scheme shown in figure 1.

\[ \int B = \int (1) - \int (2) - \int (3) + \int (4) \]

Fig. 1. $O(1)$-scheme of calculating the sum of intensities within Box $B$.

While the SURF method [4] only utilizes standard non-rotated boxes, STAR [5] approximates a center-surrounded wavelet of an inner and an outer circle by hexagons using several slanted integral images. We also want to mention feature detection and description methods like BRISK [6] and ORB [7], which show excellent results but are not based on integral images more than using the FAST approach [8]. In our opinion, integral image structures do not only provide a benefit for image feature processing, they also employ a high reusability in subsequent, higher level image processing tasks like pattern learning and object modeling. This must be taken into account when measuring the overall performance of a visual perception system.

To overcome the rotational blindness of a standard integral image, Lienhart and Maydt [1] proposed a $45^\circ$ rotated image which allows the computation of rotated Haar features in the same way as non-rotated features. But we see a drawback in the proposed calculation pattern regarding an efficient parallelized implementation because of its recursive dependencies. We reorganize the calculation pattern and show that rotated features can hold orthogonal information w.r.t. their non-rotated equivalent. Using this orthogonality, we implement fully rotational invariance for a simple and efficient corner detector proposed in [3].

2. RELATED WORK
The first approach to calculate a 45 degrees rotated integral image was given by Lienhart and Maydt [1] in 2002. There are two passes to be done over all pixels. For a mathematical description we use a notation as follows: $I(x, y)$ is the intensity value at the image coordinates $(x, y)$, $\int_0 I(x, y)$ is the non-rotated integral image value and $\int_1 I(x, y)$ is the value of the $45^\circ$ rotated integral image. So the first pass of [1] is given by

\[ \int_1 I(x, y) = I(x, y) + \int_1 I(x-1, y-1) + \int_1 I(x-1, y) - \int_1 I(x-2, y-1) \quad (1) \]
and the second pass is given by
\[
\int_1 I(x, y) = \int_1' I(x, y) + \int_1 I(x - 1, y + 1) - \int_1 I(x - 2, y) \tag{2}
\]
The calculation pattern given here differ to the simple calculation pattern of a non-rotated integral image. \( \int_0 I \) can be calculated in two passes by first integrating the image rows and then the image columns (or vice versa):
\[
\int_0 I = \int_y \circ \int_x I \tag{3}
\]
with
\[
\int_x I(x, y) = I(x, y) + \int_x I(x - 1, y) \tag{4}
\]
\[
\int_y I(x, y) = I(x, y) + \int_y I(x, y - 1) \tag{5}
\]
The equations (4) and (5) contain one constant coordinate (emphasized) in each case and therefore can be forked into separate threads. A similar, rotated pattern is also present (emphasized) in each case and therefore can be forked in-
\[
\begin{align*}
\int_u I(x, y) &= I(x, y) + \int_u I(x - 1, y - 1) \tag{6} \\
\int_v I(x, y) &= I(x, y) + \int_v I(x - 1, y + 1) \tag{7} \\
\int_1 I &= \int_v \circ \int_u I \tag{8}
\end{align*}
\]
where \( u, v \) are now perpendicular diagonal image paths, see figure 2. \( \int_v \) integrates from the upper left to the lower right while \( \int_u \) integrates from the lower left to the upper right. The grey pixels dots outside the image border in figure 2 are virtual and hold redundant information. They can be skipped and there is no need for implementing them with extra memory. This pattern now can easily be vectorized in the same way as the non-rotated pattern of \( \int_0 I \), because the single paths are independent of each other. Be aware that this is not the case when mirroring the \( v \)-path directions. Then there will be a carry over for the virtual pixels on the right side of the image.

At this point it is important to say that equation (2) and equation (8) do not lead to equal integral images. Equation (8) comprises two independent integral images with nearly the same information. This is shown in figure 3. If this is a drawback or not, we will discuss in the next section.

3. SIMPLIFIED PATTERN FOR \( \int_1 I \)

Adapting the pattern of \( \int_0 I \) for \( \int_1 I \) by rotating it by 45° will lead to an equation set as follows:
\[
\int_u I(x, y) = I(x, y) + \int_u I(x - 1, y - 1) \tag{6}
\]
\[
\int_v I(x, y) = I(x, y) + \int_v I(x - 1, y + 1) \tag{7}
\]
\[
\int_1 I = \int_v \circ \int_u I \tag{8}
\]

Whether the reduced resolution as a result of equation (8) is a disadvantage or not, one must take into account at which stage integral images provide an increase of efficiency. This is obviously not the case when integrating a 2×2 image box. So the use of integral images gives a benefit not until higher scales. Another point is, that calculating features at pixel level is mostly disturbed by the image sensor noise. To get rid of the image sensor noise, many algorithms employ a smoothing step as preprocessing. For those reasons a full resolution integral image is not necessary in our opinion. That’s why we are convinced that equation (8) is a valid and beneficial approximation. Basically, we cut off the inefficient parts of integral images and only use the profitable ones.

The next step forward is to pack \( \int_0 I \) and \( \int_1 I \) of lower resolution into a combined image structure \( \int_{01} I \) aiming to calculate both rotated and non-rotated features for specific image positions. It is essential that the image positions where the rotated feature is calculated is exactly the same as for calculating the non-rotated one. In order to do that, we propose not to use the pixel centers as primary image positions more than...
using the intersections of the pixel borders. Thinking of a ge-
dankenexperiment, the borders of the pixels shape a grid and 
the intersections of the grid may be called gridels. Each gridel 
got its value from its neighboring pixel centers (which is 
in fact a $2 \times 2$-smoothing).

Fig. 4. Sampling of gridel positions w.r.t. feature centers ($\circ$), 
$\int_0 I$-gridels (□) and $\int_1 I$-gridels ($\diamond$).

In order to get a combined $\int_01 I$ image structure, we con-
struct a pattern of gridel positions where each gridel is either a 
feature center, a $\int_0 I$-gridel or a $\int_1 I$-gridel. The sampling is 
shown in figure 4.

5. COVERING ROTATION SPACE WITH 
ORTHOGONAL FEATURES

The most important feature of the combined integral image 
structure $\int_01 I$ is, that it is now possible to cover the complete 
$[0, 2\pi]$ rotation space. We demonstrate this by a simple expe-
riment. Consider a gradient Haar feature $D_{xy}$ and its rotated 
 pendant $D_{uv}$ as shown in figure 5. We now rotate the image 
data of a corner right in the center of both features. The result-
ing feature responses are shown in figure 6.

Fig. 5. Experiment: Rotating a corner within the centers of 
$D_{xy}$ and $D_{uv}$.

Discussing figure 6 one can see that each feature $D_{xy}$ 
and $D_{uv}$ got blind spots where the other feature got its maxi-
mums. As a consequence, $D_{xy}$ cannot register anything about 
image structures which are within the domain of $D_{uv}$. This 
causes a loss of present, valuable image information. Due to 
the fact that $D_{xy}$ and $D_{uv}$ got orthogonal responses, the fully 
coverage of rotation space (and image information) is possi-
bile without increasing efforts a lot. In our opinion this is a 
performance improvement for any algorithm or method using 
Haar wavelets.

6. APPLICATION: ENHANCING THE SIDCELL 
CORNER DETECTOR BY $\int_01 I$

The SIDCELL corner detector is a fast and efficient corner 
detector optimized for real-time applications. It basically uti-

lizes a non maximum suppressed $D_{xy}$ feature response image 
to detect corners and was proposed 2009 [3]. Applications 
were published in [9] for a realtime visual odometry solver 
and in [10] for camera stabilization. Coming from an auto-
motive background, there was no need to implement rotational 
invariance within SIDCELL. The main focus was on detec-
ting corners at different scales based on a integral image 
which was still present due to pattern recognition steps based 
on Haar wavelets. Because of its efficiency and parallelization 
abilities, it was able to solve the correspondence task at 200 
fps by the use of a GPU. See figure 8 for tracking the moti-
on disparity of an accelerating car substracted from the ego 
motion.

Fig. 6. Corner responses of $D_{xy}$ and $D_{uv}$ over rotation space.

Fig. 8. Ego motion disparity of an accelerating car using SID-
CELLs (from [9]).

Enhancing the SIDCELL method by $\int_01 I$ with an addi-
tional SIDCELL channel for $D_{uv}$, the rotational invariance of 
the detector plus an orientation measure function can be im-
plemented at the same magnitude of computation speed by 
combining $D_{xy}$ and $D_{uv}$. Though the rotational invariance 
and orientation function cause additional computation time, it 
is compensated by the lower resolution of $\int_01 I$ without loo-
Fig. 7. As a result of figure 5, SIDCELL Keypoints now cover full rotation space in detection and description. (a) and (b) show enhanced SIDCELL keypoints at different rotations. (c) and (d) show the corresponding orientations. Chessbord like features (center dot) are distinguished from corner like features as well as bright from dark features which forms a weak but fast orientation descriptor. Sheared corners (e) are also detected and subpixel accuracy by a second order interpolation (f) compensates the lower resolution of $\int I$.

The orientation function uses $D_x$ and $D_y$ Haar wavelets as well as most of the comparable methods do (e.g. SURF) but additionally employ $D_u$ and $D_v$ wavelets. All these first order derivative wavelets can be computed from the same $\int I$ coordinates as $D_{xy}$ and $D_{uv}$ so that there is no need to read extra memory (see [3] for that). As a consequence, the detector and the descriptor set is the same, which increases efficiency. Details will be shown in a separate contribution and is out of scope here. We want to give an example of the benefit the combined integral image structure $\int I$ provides by applying it to applications. Figure 7 illustrates the results of the enhanced SIDCELL approach. Figure 9 shows the keypoint tracks of different Keypoint detectors caused by a rotating artificial $2 \times 2$ chessboard pattern.

7. CONCLUSION

In this paper we introduce a novel compact integral image structure providing non-rotated and $45^\circ$ rotated integral information. This integral image structure results in a better coverage of image information. By approximating an existing computation pattern, we construct independent image paths making it suitable for parallelization. This is essential because integrating images is time consuming when implemented sequentially. Integral images are used widely so this paper provides a speedup for many applications. We discuss the efficiency of integral images and conclude that they are not efficient at pixel level and hence our approximation is valid, applicable and efficient. We eliminate inefficient parts and the rotational blindness of a standard integral image. To demonstrate the power of this novel integral image structure, we enhance SIDCELL corners by fully rotational invariance and a more powerful orientation descriptor. Especially when focusing on local descriptors we expect future improvements by adding orthogonal rotated integral image information.
8. REFERENCES


