# A Note on the Computation of Haar-Based Features for High-Dimensional Images

Ernesto Tapia
Freie Universität Berlin, Institut für Informatik
Takustr. 9, 14195 Berlin, Germany
ernesto.tapia@fu-berlin.de

May 2009

**Abstract**

The integral image approach allows the optimal computation of Haar-based features for real-time recognition of objects in image sequences. This short note gives a direction to generalize the approach to high-dimensional images. We offer a formula for optimal computation of sums on high-dimensional rectangles.

Integral Image - Haar-Based Features - High-Dimensional Image - Möbius Inversion Formula

## 1 Introduction

Many computer vision applications require quick feature extraction and classification, in particular real-time location of objects in image sequences. A solution for such problem, though first introduced for face detection, is the approach introduced by Viola and Jones [6]. They construct a boosted cascade of simple classifiers based on Haar-similar features that measure vertical, horizontal, central, and diagonal variations of pixel intensities. These features are defined as the difference between sums of image values on two, three and four rectangles, see Fig. 1.

The sum of image values $i(x',y')$ on a rectangle $(x_0,y_0]\times(x_1,y_1]$

$$A=\sum_{x_0<x'\leq x_1}\sum_{y_0<y'\leq y_1} i(x',y') \tag{1}$$

is computationally expensive, since its complexity depends on the rectangle's size. Viola and Jones use the *integral image* as an intermediate array representation to optimally compute $A$. The integral image value at the pixel $(x,y)$ is defined as

$$I(x,y)=\sum_{0\leq x'\leq x}\sum_{0\leq y'\leq y} i(x',y'), \tag{2}$$

i.e. the sum of the original image values on the rectangle $[0,0]\times[x,y]$. The integral image is computed *in one pass* over the image using the recurrence

$$c(x,y) = c(x,y-1)+i(x,y), \tag{3}$$
$$I(x,y) = I(x-1,y)+c(x,y), \tag{4}$$

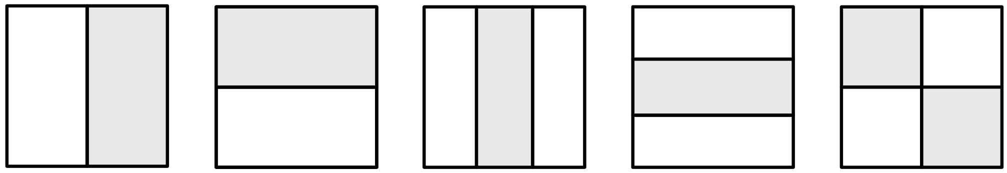

Figure 1: Haar-based rectangular features used for face recognition. The features are the sum on the values on the gray region minus the sum on the white region.

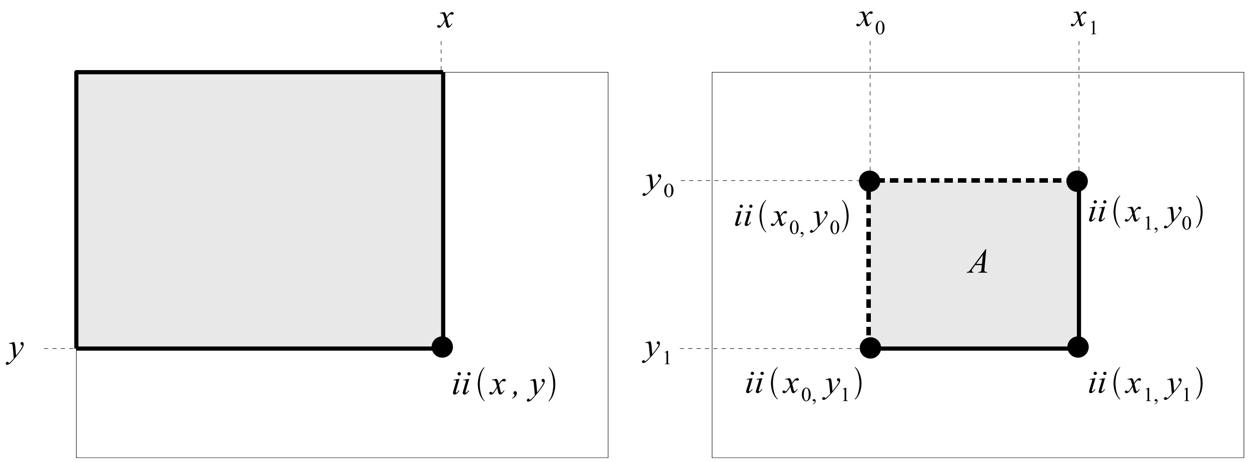


Figure 2: Left: Integral image representation. Right: The four references used to compute the image values on the gray area.

with

$$c(x, -1) = I(-1, y) = 0, \tag{5}$$

where $c(x, y)$ is called the cumulative row sum. See Fig. 2. Thus, one can compute $A$ in *constant time* using only *four references* to the integral image:

$$A = I(x_1, y_1) - I(x_1, y_0) - I(x_0, y_1) + I(x_0, y_0). \tag{6}$$

The integral image approach only uses the spatial domain, excluding the extra information given by the time dimension. By contrast, Ke et.al [3] include the time domain to detect motion events and persons' activities in videos. They extend the approach defining the *integral video* to efficiently compute volumetric features from the video's optical flow, using eight references to the integral video to compute sums on parallelepipeds, see Fig. 3.

Many other image structures could profit from this approach, such as *three dimensional images* that represent ultrasound sequences in medical applications or flow in porous media in experimental fluid dynamics [4]. These structures are not only static but dynamic high-dimensional images $i(\mathbf{p}, t)$, where $t \in [0, T]$ and $\mathbf{p} \in \mathbb{R}^n$ are the indexes in the time and spatial domain with $n = 2$, 3. Thus, a natural question is how we can generalize this method for high-dimensional images.

We noted that generalizing the approach consist basically in adapting two main steps. One that computes in *one pass* an *integral array*, and other that computes in *constant time* the sum of pixels included in an *hyper-rectangle* using only few references to the integral array.

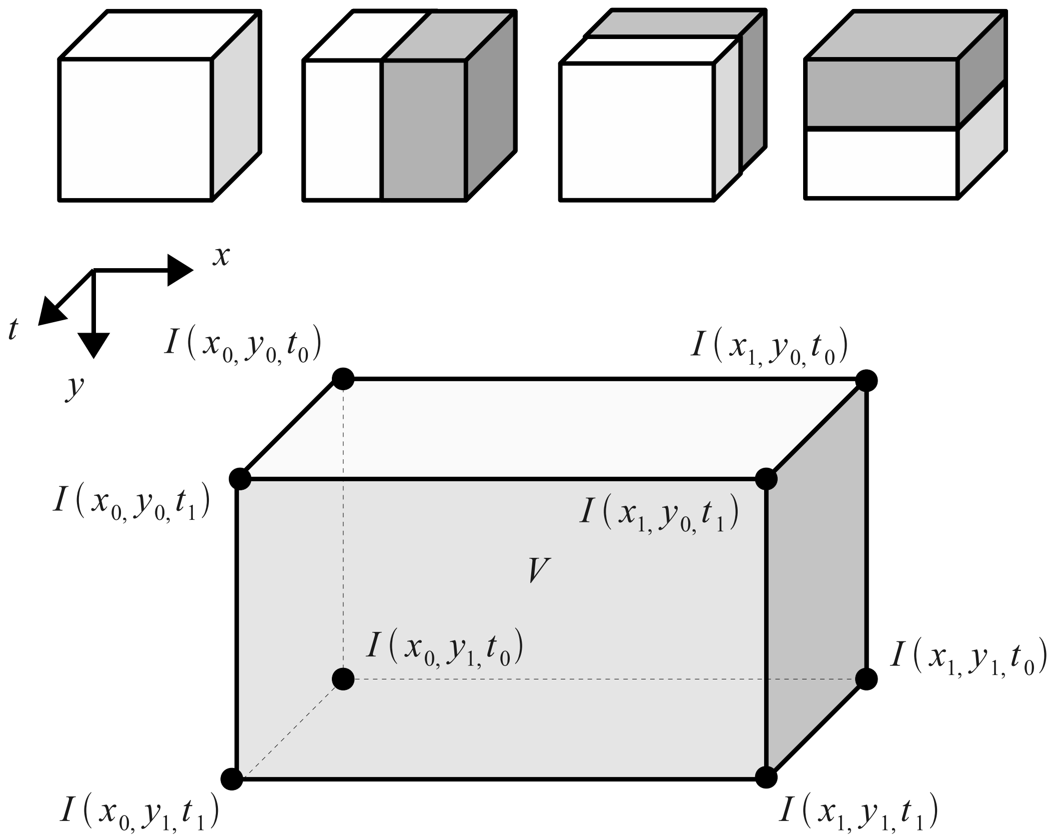


Figure 3: Above: Volumetric features computed by the integral video. Below: The black circles are the references used to compute the sum of the optical flow on the volumen $V$.

The next section states these generalization steps and begins with some useful notation and definitions.

## 2 Integral Representation in Higher Dimensions

We denote vectors with the usual notation

$$\mathbf{x} = (x_1, \ldots, x_d). \tag{7}$$

Bold-faced scalars denote vectors whose entries are equal to the scalar.

Superindexes in vectors represent a labeling, which can be a scalar $m$ such as

$$\mathbf{x}^m = (x_1^m, \ldots, x_d^m), \tag{8}$$

or a vector $\mathbf{n}$ like

$$\mathbf{x}^{\mathbf{n}} = (x_1^{n_1}, \ldots, x_d^{n_d}). \tag{9}$$

Note that, if $m$ is a number, then $x^{\mathbf{m}} = x^m$. As usual, $\mathbf{e}^m$ is a vector of the canonical basis, where $e_m^m = 1$ and $e_n^m = 0$ for $n \neq m$.

We define below a relation that plays a relevant role in this work.

**Definition 1.** *The* partial order $\preceq$ *on the vectors is defined as*

$$\mathbf{x} \preceq \mathbf{y} \Leftrightarrow x_i \leq y_i,\ i = 1, \ldots, d. \tag{10}$$

**Remark 1.** *The partial order lets us consider intervals similar to the one dimensional case, such as the semi-closed interval*

$$[\mathbf{x}, \mathbf{y}) = \{\mathbf{z} : \mathbf{x} \preceq \mathbf{z} \prec \mathbf{y}\}. \tag{11}$$

*Note that the interval defines geometrically a semi-closed hyper-rectangle. We will use without distinction interval or rectangle to denote such a set.*

**Definition 2.** *A* $d$-dimensional image *is a real-valued function*

$$i : [\mathbf{0}, \mathbf{u}] \rightarrow \mathbb{R}. \tag{12}$$

*The* integral image $I : [\mathbf{0}, \mathbf{u}] \rightarrow \mathbb{R}$ *of the image* $i$ *is defined as*

$$I(\mathbf{x}) = \sum_{\mathbf{z} \in [\mathbf{0}, \mathbf{x}]} i(\mathbf{z}). \tag{13}$$

## 2.1 Optimal Computation of Integral Images

The first step in the approach is the computation of the integral image in one pass. This step is relatively easy to generalize: if the array has dimension $d$, then one has only to maintain $d-1$ extra arrays and define a recursion similar to (3)-(5). We state formally this idea:

**Proposition 1.** *The integral image* $I$ *is computed* in one pass *over the image* $i$ *using the arrays* $c_m$, $m = 1, \ldots, d-1$, *and the recurrence*

$$\begin{aligned} I(\mathbf{x}) &= I(\mathbf{x} - \mathbf{e}^1) + c_1(\mathbf{x}), & (14)\\ c_1(\mathbf{x}) &= c_1(\mathbf{x} - \mathbf{e}^2) + c_2(\mathbf{x}) & (15)\\ &\vdots \\ c_{d-1}(x) &= c_{d-1}(\mathbf{x} - \mathbf{e}^d) + i(\mathbf{x}), & (16) \end{aligned}$$

*with*

$$c_m(\mathbf{x}) = I(\mathbf{x}) = 0, \tag{17}$$

*when* $x_n < 0$ *for* $m = 1, \ldots, d-1$ *and* $n = 1, \ldots, d$.

*Proof.* Reordering the sum in the integral image we have:

$$\begin{aligned} I(\mathbf{x}) &= \sum_{\mathbf{0} \preceq \mathbf{z} \preceq \mathbf{x}} i(\mathbf{z}) & (18)\\ &= \sum_{0 \leq z_1 \leq x_1} \cdots \sum_{0 \leq z_d \leq x_d} i(z_1, \ldots, z_d) & (19)\\ &= \sum_{0 \leq z_1 \leq x_1 - 1} \cdots \sum_{0 \leq z_d \leq x_d} i(z_1, \ldots, z_d) + & (20)\\ &\quad \sum_{0 \leq z_2 \leq x_2} \cdots \sum_{0 \leq z_d \leq x_d} i(x_1, z_2, \ldots, z_d) & (21) \end{aligned}$$

If we define

$$c_1(\mathbf{x}) = \sum_{0 \le z_2 \le x_2} \cdots \sum_{0 \le z_d \le x_d} i(x_1, z_2, \ldots, z_d), \tag{22}$$

then we have

$$I(\mathbf{x}) = I(\mathbf{x} - \mathbf{e}^1) + c_1(\mathbf{x}). \tag{23}$$

Similarly, we define for $n = 1, \ldots, d-1$

$$c_n(\mathbf{x}) = c_n(\mathbf{x} - \mathbf{e}^{n+1}) + c_{n+1}(\mathbf{x}), \tag{24}$$

with

$$c_{n+1}(\mathbf{x}) = \sum_{0 \le z_{n+1} \le x_{n+1}} \cdots \sum_{0 \le z_d \le x_d} i(x_1, \ldots, x_n, z_{n+1}, \ldots, z_d), \tag{25}$$

where $c_d(\mathbf{x}) = i(\mathbf{x})$. □

The second step in the approach is the optimal computation of

$$A = \sum_{\mathbf{z} \in (\mathbf{x}^0, \mathbf{x}^1]} i(\mathbf{z}), \tag{26}$$

given the image $i$, its integral image $I$, and the rectangle of interest $(\mathbf{x}^0, \mathbf{x}^1]$. This step is much more difficult to generalize than it seems. For two-dimensional rectangles, one can derive an optimal expression using, for example, visual inspection of Fig. 2. However, using only visual inspection to compute sums on a rectangle in dimension higher than two is very difficult if not impossible. Actually, one needs a *general expression* to compute sums on the rectangles in terms of the integral array. The key we found is relating the sum on the rectangle of interest with the partial ordering on vectors, the integral image and the concepts we define next.

**Definition 3.** *The* corners *of the rectangle* $(\mathbf{x}^0, \mathbf{x}^1]$ *are the vectors*

$$\mathbf{x}^{\mathbf{q}} = (x_1^{q_1}, \ldots, x_d^{q_d}), \tag{27}$$

*where* $\mathbf{q} \in \{0,1\}^d$.

**Definition 4.** *The* binary representation *of the rectangle* $[\mathbf{x}^0, \mathbf{x}^1)$ *are the sums on its corners*

$$S(\mathbf{q}) = \sum_{\mathbf{z} \in [\mathbf{0}, \mathbf{x}^{\mathbf{q}}]} i(\mathbf{z}), \tag{28}$$

$$A(\mathbf{q}) = \sum_{\mathbf{z} \in (\mathbf{x}^{\mathbf{q}-\mathbf{1}}, \mathbf{x}^{\mathbf{q}}]} i(\mathbf{z}), \tag{29}$$

*where* $x_n^{-1} = 0$ *for* $n = 1, \ldots, d$.

Figure 4 compares the concepts defined above with the original definitions of Viola and Jones.

The binary representation offers a useful way to prove the generalization of the second step, using the following result of combinatorial theory:

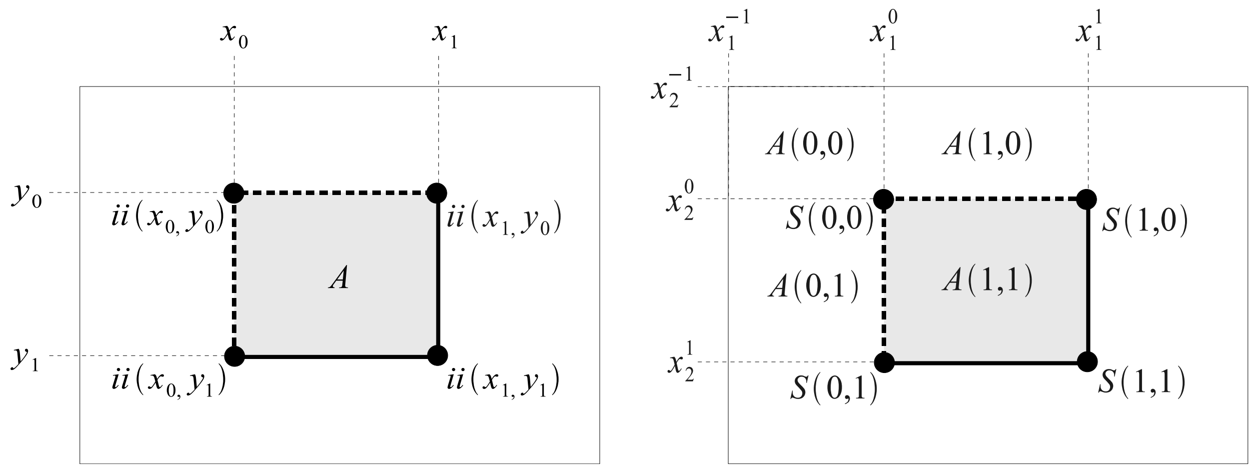


Figure 4: Notation using binary labels for integral representations.

**Proposition 2.** *[Möbius Inversion Formula] Let $f(q)$ be a real-valued function, defined for $q$ ranging in a locally finite partially ordered set $Q$. Let an element $m$ exist with the property that $f(q) = 0$ unless $q \geq m$. Suppose that*

$$g(q) = \sum_{p \leq q} f(p). \tag{30}$$

*Then*

$$f(q) = \sum_{p \leq q} \mu(p, q) g(p), \tag{31}$$

*where the function $\mu$ is called the* Möbius function *of the partially ordered set $Q$. The value $\mu(p, q)$ is* computed recursively *for $p \leq q$ as*

$$\mu(p, q) = \begin{cases} 1, & p = q \\ -\sum_{p \leq r < q} \mu(p, r), & p \neq q. \end{cases} \tag{32}$$

We refer the interested reader to [5] for the proof of the Möbius Inversion Formula.

Now, we state an important result of this section:

**Proposition 3.** *We can express a binary representation of the rectangle $[\mathbf{x}^0, \mathbf{x}^1)$ as*

$$S(\mathbf{q}) = \sum_{\mathbf{p} \preceq \mathbf{q}} A(\mathbf{p}), \tag{33}$$

$$A(\mathbf{q}) = \sum_{\mathbf{p} \preceq \mathbf{q}} (-1)^{\ell(\mathbf{q}) - \ell(\mathbf{p})} S(\mathbf{p}), \tag{34}$$

*where*

$$\ell(\mathbf{q}) = \sum_{i=1}^{d} q_i. \tag{35}$$

*Proof.* Equation (33) is easily proved using

$$[\mathbf{0}, \mathbf{x}^{\mathbf{q}}] = \bigcup_{\mathbf{p} \preceq \mathbf{q}} (\mathbf{x}^{\mathbf{p}-\mathbf{1}}, \mathbf{x}^{\mathbf{p}}]. \tag{36}$$

Now, we prove (34). Observe that the element $m$ mentioned in Prop. 2 guarantees that the sum (30) is well defined. In our case, the sum (33) runs over a finite number of indexes, so it is already well defined. Thus, the partial order $\preceq$ and (33) satisfy the hypothesis of the the Möbius Inversion Formula, then we conclude

$$A(\mathbf{q}) = \sum_{\mathbf{p}\preceq\mathbf{q}} \mu(\mathbf{p},\mathbf{q})S(\mathbf{p}). \tag{37}$$

We have only to prove

$$\mu(\mathbf{p},\mathbf{q}) = (-1)^{\ell(\mathbf{q})-\ell(\mathbf{p})}. \tag{38}$$

For such purpose, we use the definition (32) of the Möbius function. Suppose first that $\mathbf{p}=\mathbf{q}$. Then we have $\ell(\mathbf{p}) = \ell(\mathbf{q})$ and thus

$$\mu(\mathbf{p},\mathbf{q}) = 1 = (-1)^0 = (-1)^{\ell(\mathbf{q})-\ell(\mathbf{p})}. \tag{39}$$

Suppose that $\mathbf{p} \neq \mathbf{q}$ and that (38) is valid for $\mu(\mathbf{p},\mathbf{r})$, with $\mathbf{p} \preceq \mathbf{r} \prec \mathbf{q}$. Using the definition (32), we have

$$\begin{aligned}
\mu(\mathbf{p},\mathbf{q}) &= -\sum_{\mathbf{p}\preceq\mathbf{r}\prec\mathbf{q}} \mu(\mathbf{p},\mathbf{r}) && (40)\\
&= -\sum_{i=0}^{\ell(\mathbf{q})-\ell(\mathbf{p})-1} \sum_{\substack{\mathbf{p}\preceq\mathbf{r}\prec\mathbf{q}\\ \ell(\mathbf{r})=\ell(\mathbf{p})+i}} \mu(\mathbf{p},\mathbf{r}), && (41)\\
&= -\sum_{i=0}^{\ell(\mathbf{q})-\ell(\mathbf{p})-1} \sum_{\substack{\mathbf{p}\preceq\mathbf{r}\prec\mathbf{q}\\ \ell(\mathbf{r})=\ell(\mathbf{p})+i}} (-1)^{\ell(\mathbf{r})-\ell(\mathbf{p})} && (42)\\
&= -\sum_{i=0}^{\ell(\mathbf{q})-\ell(\mathbf{p})-1} \sum_{\substack{\mathbf{p}\preceq\mathbf{r}\prec\mathbf{q}\\ \ell(\mathbf{r})=\ell(\mathbf{p})+i}} (-1)^{i} && (43)\\
&= -\sum_{i=0}^{\ell(\mathbf{q})-\ell(\mathbf{p})-1} (-1)^i \left|\{\mathbf{p}\preceq\mathbf{r}\prec\mathbf{q} : \ell(\mathbf{r}) = \ell(\mathbf{p})+i\}\right| && (44)\\
&= -\sum_{i=0}^{\ell(\mathbf{q})-\ell(\mathbf{p})-1} (-1)^i \binom{\ell(\mathbf{q})-\ell(\mathbf{p})}{i} && (45)\\
&= -\sum_{i=0}^{\ell(\mathbf{q})-\ell(\mathbf{p})} (-1)^i \binom{\ell(\mathbf{q})-\ell(\mathbf{p})}{i} + (-1)^{\ell(\mathbf{q})-\ell(\mathbf{p})} && (46)\\
&= (-1)^{\ell(\mathbf{q})-\ell(\mathbf{p})}. && (47)
\end{aligned}$$

□

The above result lets us conclude the second step in the generalization of the approach:

**Proposition 4.** *Given an image* $i$*, its integral image* $I$ *and the rectangle* $[\mathbf{x}^0, \mathbf{x}^1)$*, we can compute the sum* $A$ *of the image values on the rectangle using* $2^d$ *references to the integral image with the formula*

$$A = \sum_{\mathbf{p} \in \{0,1\}^d} (-1)^{d-\ell(\mathbf{p})} I(\mathbf{x}^{\mathbf{p}}). \tag{48}$$

*Proof.* Equation (48) is an immediate consequence of Prop. 3 and

$$\begin{aligned} S(\mathbf{q}) &= I(\mathbf{x}^{\mathbf{q}}), & (49) \\ A(\mathbf{1}) &= A, & (50) \end{aligned}$$

which are derived from the definition of binary representation. □

# 3 Concluding Remarks

This short note gives a direction to generalize the Integral Image approach for $d$-dimensional images. The generalization consists in the computation of a integral array in one pass and the optimal computation of sums on rectangles using $2^d$ references to the integral array.

The generalization has some drawbacks for high $d$. The computation of the integral array uses $d-1$ extra arrays, that means a memory increase that many personal computers could not support for large images. Another problem is the *curse of dimensionality*. The boosting method used by Viola and Jones selects the best feature from all possible ones generated by scaling, rotating and translating a base feature through the image. If we consider, for example, the first (and simplest) volumetric feature in Fig. 3, then the number of features is $O(n^{2d})$ for an image of dimension $n \times n \times n$. Despite these drawbacks, the results presented in this note seem to us an attractive starting point for future applications.

By the other hand, there is another generalization not related directly with object recognition. If we use integral instead of sums in Prop. 4, then we can informally say that (48) offers a generalization of the *Fundamental Theorem of Calculus* [1]. Remember that this theorem states that, if $f$ is a continuous function on the interval $[x^0, x^1]$ and $F$ is an antiderivative of $f$, then

$$\int_{x^0}^{x^1} f(x)\, dx = F(x^1) - F(x^0). \tag{51}$$

Note that the integral image can be regarded as an “antiderivative” of the original image, since it is an integral of the image with a variable upper limit, similar to the antiderivative of a function of one variable. Using this analogy, the generalization to several variables computes the integral of a function $f$ on the interval using its antiderivative $F$ evaluated at the rectangle’s corners:

$$\int_{[\mathbf{x}^0, \mathbf{x}^1]} f(x)\, dx = \sum_{\mathbf{p} \in \{0,1\}^d} (-1)^{d-\ell(\mathbf{p})} F(\mathbf{x}^{\mathbf{p}}). \tag{52}$$

This generalization is a “direct” generalization of the Fundamental Theorem of Calculus as compared to Stoke’s Theorem, which involves specialized concepts such as *manifolds* and *differential forms* [2]. A formal statement of this generalization needs the defintion of integrability and antiderivative for functions of several variables, among other concepts. We are sure, however, the proof of the generalization could follow the procedure we developed in this note to demonstrate Prop. 4.

## Acknowledgments

Special thanks go to Marte Ramírez-Ortegón for his helpful comments about the preliminary results of this note. I am very grateful to Dr. Fernando Galaz-Fontes for his comments which helped improve this note.

## References

[1] T.M. Apostol. *Calculus: One-Variable Calculus with an Introduction to Linear Algebra*, volume 1. John Wiley & Sons Inc, 1967.

[2] V.J. Katz. The history of Stokes' Theorem. *Mathematics Magazine*, pages 146–156, 1979.

[3] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *IEEE International Conference on Computer Vision*, volume 1, pages 166–173, Washington, DC, USA, 2005.

[4] T. Preusser and M. Rumpf. Extracting motion velocities from 3d image sequences and coupled spatio-temporal smoothing. In *SPIE Converence on Visualization and Data Analysis*, volume 5009, pages 181–192, 2003.

[5] G.C. Rota. On the Foundations of Combinatorial Theory I - Theory of Möbius Functions. *Probability Theory and Related Fields*, 2(4):340–368, 1964.

[6] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.