Tracking Library for the Web

Eduardo A. Lundgren Melo

Universidade Federal de Pernambuco

Centro de Informática U·F·P·E

Master of Science in Computer Science

Silvio de Barros Melo (*Advisor*)

Veronica Teichrieb (*Co-Advisor*)

Outline

Outline

1 Introduction

2 Basic concepts
- Web
- Visual tracking

3 Tracking library for the web

Motivation

- The web browser environment is evolving fast

Motivation

- The web browser environment is evolving fast
- Phones and notebooks devices have embedded web browser

Motivation

- The web browser environment is evolving fast
- Phones and notebooks devices have embedded web browser
- Entertainment solutions are gaining space on the web

Motivation

- The web browser environment is evolving fast
- Phones and notebooks devices have embedded web browser
- Entertainment solutions are gaining space on the web
- Vision is an accurate and low-cost solution

Problem definition

- JavaScript is a language interpreted by all web browsers

Problem definition

- JavaScript is a language interpreted by all web browsers
- Interpreted languages have limited computational power

Problem definition

- JavaScript is a language interpreted by all web browsers
- Interpreted languages have limited computational power
- Modern web browsers can natively capture the user media

Problem definition

- JavaScript is a language interpreted by all web browsers
- Interpreted languages have limited computational power
- Modern web browsers can natively capture the user media
- Capturing and processing user media are required steps for visual tracking

Objectives

- Facilitate user interaction with the web browser

Objectives

- Facilitate user interaction with the web browser
- Accelerate the use of visual tracking in commercial products

Objectives

- Facilitate user interaction with the web browser
- Accelerate the use of visual tracking in commercial products
- Provide a cross-platform tracking library

Objectives

- Facilitate user interaction with the web browser
- Accelerate the use of visual tracking in commercial products
- Provide a cross-platform tracking library
- Design and implement a tracking library for the web

Outline

World Wide Web

The World Wide Web is a shared information system operating on top of the Internet

The beggining of the web

- Plain text and images were the most advanced features

The beggining of the web

- Plain text and images were the most advanced features
- In 1994, the World Wide Web Consortium (W3C) was founded

The beggining of the web

- Plain text and images were the most advanced features
- In 1994, the World Wide Web Consortium (W3C) was founded
- Companies were able to contribute to the W3C specifications

The beggining of the web

- Plain text and images were the most advanced features
- In 1994, the World Wide Web Consortium (W3C) was founded
- Companies were able to contribute to the W3C specifications
- Today's web is a result of the ongoing efforts of an open web

The modern web

- Contributions transformed the web in a growing universe

The modern web

- Contributions transformed the web in a growing universe
- Videos, audio, photos, interactive content, 3D graphics

The modern web

- Contributions transformed the web in a growing universe
- Videos, audio, photos, interactive content, 3D graphics
- Processed by the Graphics Processing Unit (GPU)

The modern web

- Contributions transformed the web in a growing universe
- Videos, audio, photos, interactive content, 3D graphics
- Processed by the Graphics Processing Unit (GPU)
- Without requiring any third-party plugins installation

Browser technologies

- Web pages are written using HyperText Markup Language

Browser technologies

- Web pages are written using HyperText Markup Language
- Web can be augmented with other technologies

Browser technologies

- Web pages are written using HyperText Markup Language
- Web can be augmented with other technologies
- JavaScript is the main programming language

Browser technologies

- Web pages are written using HyperText Markup Language
- Web can be augmented with other technologies
- JavaScript is the main programming language
- Layout and style information uses Cascading Style Sheets

Browser architecture

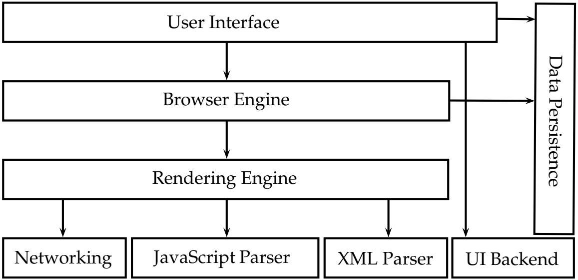

Figure : Reference architecture for web browsers

Audio and video

Figure : Video and audio HTML5 elements

Audio and video

```
<video autoplay></video>
<script>
  var video = document.querySelector('video');
  navigator.getUserMedia({video: true, audio: true}, function(localMediaStream) {
      video.src = window.URL.createObjectURL(localMediaStream);
      video.onloadedmetadata = function(e) { alert('Ready to go.') };
  }, onFail);
</script>
```

Listing 1: Capturing browser microphone and camera

Canvas element

- HTML5 element

Canvas element

- HTML5 element
- Resolution-dependent bitmap canvas

Canvas element

- HTML5 element
- Resolution-dependent bitmap canvas
- Two-dimensional grid, computer graphics coordinate system

Canvas element

- HTML5 element
- Resolution-dependent bitmap canvas
- Two-dimensional grid, computer graphics coordinate system
- Can render graphs, game graphics, art, or other visual images

Canvas element

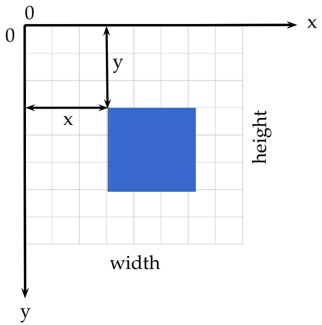

Figure : The canvas coordinate space

JavaScript typed arrays

- In the past, raw data was accessed as a string

JavaScript typed arrays

- In the past, raw data was accessed as a string
- Browsers need to quickly manipulate raw binary data

JavaScript typed arrays

- In the past, raw data was accessed as a string
- Browsers need to quickly manipulate raw binary data
- Typed data structures were added to JavaScript

JavaScript typed arrays

- In the past, raw data was accessed as a string
- Browsers need to quickly manipulate raw binary data
- Typed data structures were added to JavaScript
- JavaScript-typed arrays access raw binary more efficiently

Typed arrays performance benchmark

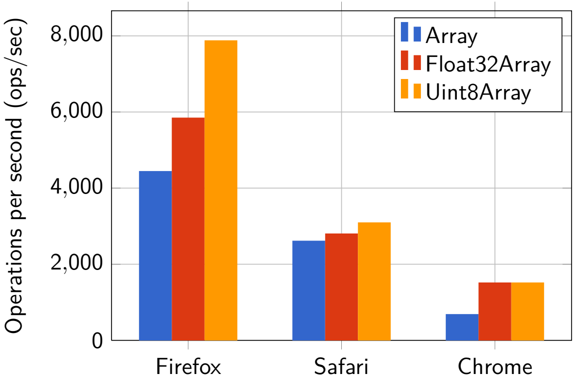

Figure : Regular *vs* typed arrays performance benchmark

What is the relation between typed arrays and canvas?

- Videos and images pixels can be drawn on a canvas bitmap

What is the relation between typed arrays and canvas?

- Videos and images pixels can be drawn on a canvas bitmap
- Canvas raw binary data can be accessed from JavaScript

What is the relation between typed arrays and canvas?

- Videos and images pixels can be drawn on a canvas bitmap
- Canvas raw binary data can be accessed from JavaScript
- Canvas array of pixels, is in row-major order

What is the relation between typed arrays and canvas?

- Videos and images pixels can be drawn on a canvas bitmap
- Canvas raw binary data can be accessed from JavaScript
- Canvas array of pixels, is in row-major order
- Consider the 2×3 array $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$, in row-major order it is laid out contiguously in linear memory as $\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$.

What is the relation between typed arrays and canvas?

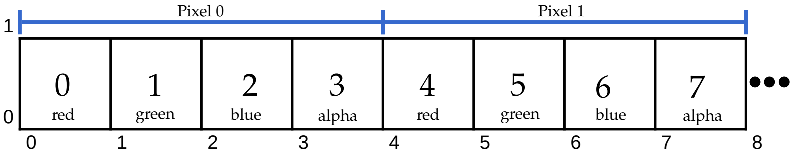

Figure : The canvas image data array of pixels

What is the relation between typed arrays and canvas?

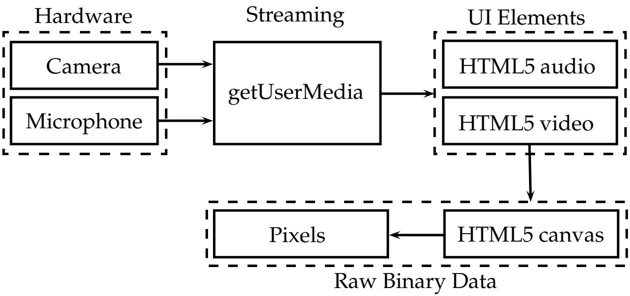

Figure : Access flow of raw binary data captured from videos on modern browsers

Visual tracking

Tracking an object in a video sequence means continuously identifying its location when either the object or the camera are moving

Figure : Example of an accurate object tracking robust to occlusion

Visual tracking

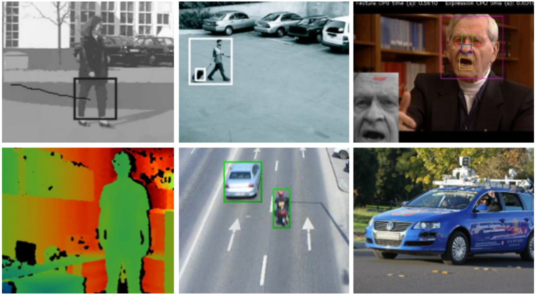

Figure : Computer vision applications: motion-based recognition (top left); automated surveillance (top center); video indexing (top right); human-computer interaction (bottom left); traffic monitoring (bottom center); vehicle navigation (bottom right).

Which devices could use tracking.js?

Different devices such as mobile phones, notebooks, and even head-worn (Google Project Glass), provide an embedded web browser capable to run JavaScript and HTML5.

Outline

tracking.js

Tracking library for the web aiming to provide a common infrastructure to develop applications and to accelerate the use of those techniques on the web in commercial products

tracking.js

Tracking library for the web aiming to provide a common infrastructure to develop applications and to accelerate the use of those techniques on the web in commercial products

It runs on native web browsers without requiring third-party plugins installation

Related work

- FLARToolKit: a port of the well-known ARToolKit marker tracking library to ActionScript

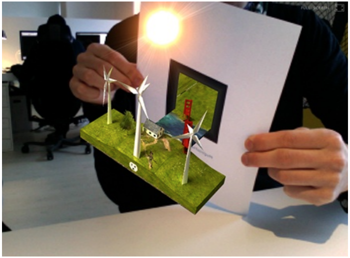

Figure : Marker based AR for the web using FLARToolKit

Related work

- JSARToolkit: is a JavaScript port of FLARToolKit

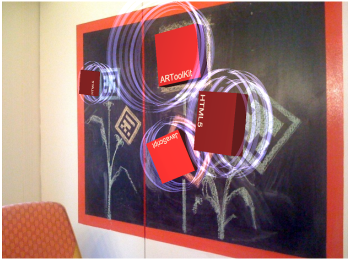

Figure : Marker-based AR for the web using JSARToolKit

Related work

- Unifeye Viewer: from Metaio company, it offers a robust markerless tracking solution for the web to ActionScript

Figure : Markerless example of image projected over a magazine cover using Unifeye Viewer solution

Library modules

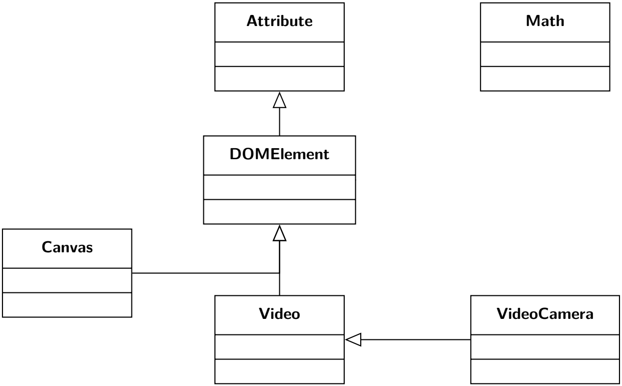

Figure : Base classes of tracking.js library

Library modules

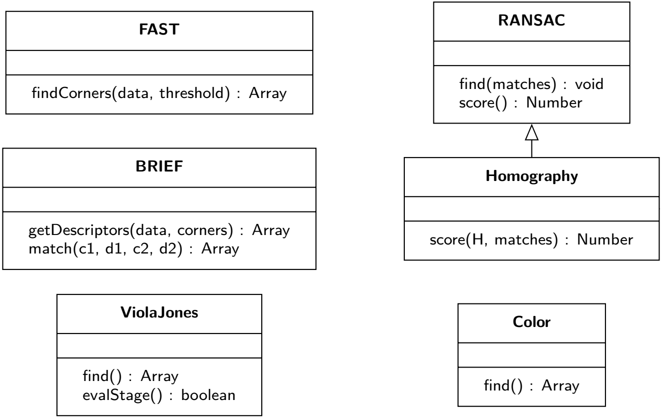

Figure : Visual tracking classes of tracking.js library

Feature detector

- Detects individual features (*keypoints*) across images

Feature detector

- Detects individual features (*keypoints*) across images
- Robustness against partial occlusions or matching errors

Feature detector

- Detects individual features (*keypoints*) across images
- Robustness against partial occlusions or matching errors
- Used as the first step of many vision tasks such as tracking

Feature detector

- Detects individual features (*keypoints*) across images
- Robustness against partial occlusions or matching errors
- Used as the first step of many vision tasks such as tracking
- Features from Accelerated Segment Test (FAST)

Feature detector

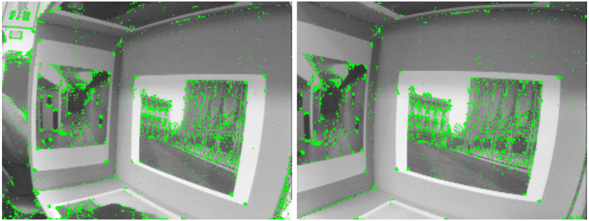

Figure : Image features detected on two different frames, green pixels represents found keypoints

Feature detector

Features from Accelerated Segment Test (FAST)

It works by testing a small patch of an image to see if it could be a corner

Feature detector

Features from Accelerated Segment Test (FAST)

It works by testing a small patch of an image to see if it could be a corner

The detector is evaluated using a circle surrounding the candidate pixel

Feature detector

Features from Accelerated Segment Test (FAST)

It works by testing a small patch of an image to see if it could be a corner

The detector is evaluated using a circle surrounding the candidate pixel

The test is based on whether the concentric contiguous arcs around the pixel are significantly different from the central pixel p

Feature detector

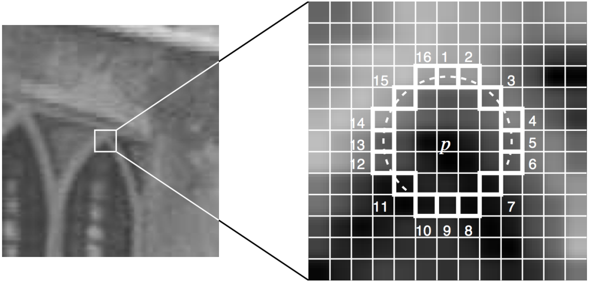

Figure : FAST: point segment test corner detection in an image patch

Feature detector

- FAST technique was chosen for feature detector

Feature detector

- FAST technique was chosen for feature detector
- Robust against occlusions, matching errors and illumination

Feature detector

- FAST technique was chosen for feature detector
- Robust against occlusions, matching errors and illumination
- Has excellent repeatability

Feature detector

- FAST technique was chosen for feature detector
- Robust against occlusions, matching errors and illumination
- Has excellent repeatability
- Computational complexity of the technique is low

Feature extractor

- To estimate motion, match sets of features is required

Feature extractor

- To estimate motion, match sets of features is required
- For each point $\{m_i\}$ in the first image, search in a region of the second image around location $\{m_i\}$ for point $\{m'_j\}$

Feature extractor

- To estimate motion, match sets of features is required
- For each point $\{m_i\}$ in the first image, search in a region of the second image around location $\{m_i\}$ for point $\{m'_j\}$
- The search is based on the similarity of the local image windows

Feature extractor

- To estimate motion, match sets of features is required
- For each point $\{m_i\}$ in the first image, search in a region of the second image around location $\{m_i\}$ for point $\{m'_j\}$
- The search is based on the similarity of the local image windows
- Binary Robust Independent Elementary Features (BRIEF)

Feature extractor

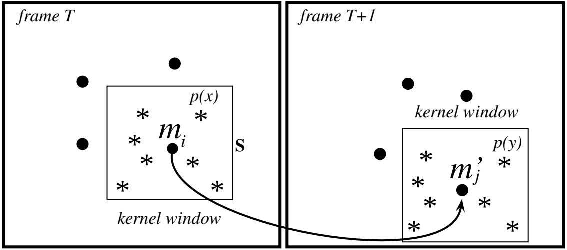

Figure : Feature extractor

Feature extractor

Binary Robust Independent Elementary Features (BRIEF)

Searches for correspondent points in the current frame and only points that are highly descriptive invariant features, called keypoints, are tested

Feature extractor

Binary Robust Independent Elementary Features (BRIEF)

Searches for correspondent points in the current frame and only points that are highly descriptive invariant features, called keypoints, are tested

After the keypoints are detected they need to be described and the respective matching point should be found

Feature extractor

Binary Robust Independent Elementary Features (BRIEF)

Searches for correspondent points in the current frame and only points that are highly descriptive invariant features, called keypoints, are tested

After the keypoints are detected they need to be described and the respective matching point should be found

BRIEF uses a binary string to describe the keypoints and having local descriptors that are fast to compute, to match and being memory efficient are important aspects

Feature extractor

To generate the binary strings it is defined the test τ on patch $\mathbf{p}$ of size $\mathbf{S} \times \mathbf{S}$ as

$$\tau(\mathbf{p}; x, y) := \begin{cases} 1 & \text{if} \quad \mathbf{p(x)} < \mathbf{p(y)}, \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{p(x)}$ is the pixel intensity. The set of binary tests is defined by the n_d ($\mathbf{x}$, $\mathbf{y}$)-location pairs uniquely chosen during the initialization

Feature extractor

The n_d-dimensional bit-string is our BRIEF descriptor for each keypoint

$$f_{n_d}(\mathbf{p}) := \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(\mathbf{p}; x, y).$$

In this work, $n_d = 128$ was used, since it presented good matching results and performance. The number of bytes required to store the descriptor can be calculated by $k = n_d/8$

Feature extractor

- Each keypoint is described with its binary string

Feature extractor

- Each keypoint is described with its binary string
- Each keypoint is compared with the closest matching point

Feature extractor

- Each keypoint is described with its binary string
- Each keypoint is compared with the closest matching point
- Distance metric is critical to the performance

Feature extractor

- Each keypoint is described with its binary string
- Each keypoint is compared with the closest matching point
- Distance metric is critical to the performance
- Using binary strings reduces the size of the descriptor

Feature extractor

Given two image patches x and y, denote their binary descriptors as $b(x) \in \{0,1\}^n$ and $b(y) \in \{0,1\}^n$ respectively, the Hamming distance is computed by

$$Ham(x,y) = \sum_{i=1}^{n} b_i(x) \otimes b_i(y)$$

From the hamming distance, the Hamming weight can be calculated

$$WHam(x,y) = \sum_{i=1}^{n} w_i(b_i(x) \otimes b_i(y))$$

Feature extractor

- BRIEF technique was chosen for feature extractor

Feature extractor

- BRIEF technique was chosen for feature extractor
- Use binary strings to describe the keypoints

Feature extractor

- BRIEF technique was chosen for feature extractor
- Use binary strings to describe the keypoints
- Fast to compute, to match and is memory efficient

Feature extractor

- BRIEF technique was chosen for feature extractor
- Use binary strings to describe the keypoints
- Fast to compute, to match and is memory efficient
- Computational complexity of the technique is low

Homography estimation

Homographies are estimated between images by finding feature correspondences on them

Homography estimation

Homographies are estimated between images by finding feature correspondences on them

A 2D point (x, y) in an image can be represented as a 3D vector $\mathbf{x} = (x_1, x_2, x_3)$ where $x = \frac{x_1}{x_3}$ and $y = \frac{x_2}{x_3}$

Homography estimation

Homography is a mapping from $P^2 \rightarrow P^2$ which is a projectivity if and only if there exists a non-singular 3×3 matrix H such that for any point in P^2 represented by vector $\mathbf{x}$ it is true that its mapped point equals $H\mathbf{x}$

$$c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad H = \begin{pmatrix} h1 & h2 & h3 \\ h4 & h5 & h6 \\ h7 & h8 & h9 \end{pmatrix},$$

where c is any non-zero constant, $(\ u\ v\ 1\ )^T$ represents $\mathbf{x'}$, $(\ x\ y\ 1\ )^T$ represents $\mathbf{x}$

Random sample consensus (RANSAC)

RANSAC (Random Sample Consensus) is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers

Random sample consensus (RANSAC)

RANSAC (Random Sample Consensus) is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers

It is the most commonly used robust estimation method for homographies

Random sample consensus (RANSAC)

The idea of the algorithm is pretty simple

Random sample consensus (RANSAC)

The idea of the algorithm is pretty simple

- For N iterations, a random sample of 4 correspondences is selected and H is computed

Random sample consensus (RANSAC)

The idea of the algorithm is pretty simple

- For N iterations, a random sample of 4 correspondences is selected and H is computed
- Each other correspondence is classified as an inlier or outlier

Random sample consensus (RANSAC)

The idea of the algorithm is pretty simple

- For N iterations, a random sample of 4 correspondences is selected and H is computed
- Each other correspondence is classified as an inlier or outlier
- The iteration containing more inliers is selected

Random sample consensus (RANSAC)

The idea of the algorithm is pretty simple

- For N iterations, a random sample of 4 correspondences is selected and H is computed
- Each other correspondence is classified as an inlier or outlier
- The iteration containing more inliers is selected
- The matrix H is recomputed from all inliers in that iteration

Rapid object detection (Viola Jones)

- Robust and extremely rapid object detection

Rapid object detection (Viola Jones)

- Robust and extremely rapid object detection
- Became popular mainly because rapid face detection

Rapid object detection (Viola Jones)

- Robust and extremely rapid object detection
- Became popular mainly because rapid face detection
- A training phase is required

Rapid object detection (Viola Jones)

- Robust and extremely rapid object detection
- Became popular mainly because rapid face detection
- A training phase is required
- A scanning detector is what makes the detection

Rapid object detection (Viola Jones)

Features

Images are classified based on the value of simple features reminiscent of Haar basis functions

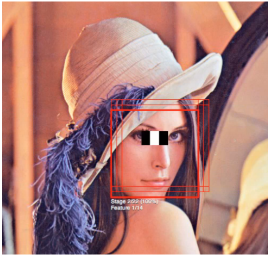

Figure : Example rectangle features shown relative to the enclosing detection window

Rapid object detection (Viola Jones)

- Three kinds of features are used

Rapid object detection (Viola Jones)

- Three kinds of features are used
- Two-rectangle feature is the difference between the sum of the pixels within two rectangular regions

Rapid object detection (Viola Jones)

- Three kinds of features are used
- Two-rectangle feature is the difference between the sum of the pixels within two rectangular regions
- Three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle

Rapid object detection (Viola Jones)

- Three kinds of features are used
- Two-rectangle feature is the difference between the sum of the pixels within two rectangular regions
- Three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle
- Four-rectangle feature computes the difference between diagonal pairs of rectangles

Rapid object detection (Viola Jones)

Integral Image

Rectangle features can be computed very rapidly using an intermediate representation for the image which we call the integral image

The integral image at location x, y contains the sum of the pixels above and to the left of x, y, inclusive

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Rapid object detection (Viola Jones)

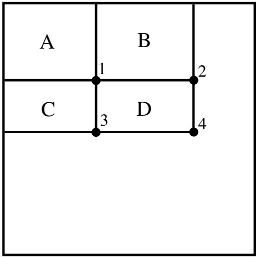

Figure : The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is $A + B$, at location 3 is $A + C$ and at location 4 is $A + B + C + D$

Rapid object detection (Viola Jones)

Scanning detector algorithm

Rapid object detection (Viola Jones)

Scanning detector algorithm

1. Create or scale a 20×20 squared block by 1.25 per iteration

Rapid object detection (Viola Jones)

Scanning detector algorithm

1. Create or scale a 20×20 squared block by 1.25 per iteration
2. Loop the block by Δ pixels over the image

Rapid object detection (Viola Jones)

Scanning detector algorithm

1. Create or scale a 20×20 squared block by 1.25 per iteration
2. Loop the block by Δ pixels over the image
3. For each block location, loop the tree and evaluate each stage

Rapid object detection (Viola Jones)

Scanning detector algorithm

1. Create or scale a 20×20 squared block by 1.25 per iteration
2. Loop the block by Δ pixels over the image
3. For each block location, loop the tree and evaluate each stage
4. Positive stage evaluate next stage, otherwise stops the loop

Rapid object detection (Viola Jones)

Scanning detector algorithm

1. Create or scale a 20×20 squared block by 1.25 per iteration
2. Loop the block by Δ pixels over the image
3. For each block location, loop the tree and evaluate each stage
4. Positive stage evaluate next stage, otherwise stops the loop
5. If all stages were positive store the rectangle

Rapid object detection (Viola Jones)

Scanning detector algorithm

1. Create or scale a 20×20 squared block by 1.25 per iteration
2. Loop the block by Δ pixels over the image
3. For each block location, loop the tree and evaluate each stage
4. Positive stage evaluate next stage, otherwise stops the loop
5. If all stages were positive store the rectangle
6. Once the tree is done, group the overlapping rectangles

Rapid object detection (Viola Jones)

Scanning detector algorithm

1. Create or scale a 20×20 squared block by 1.25 per iteration
2. Loop the block by Δ pixels over the image
3. For each block location, loop the tree and evaluate each stage
4. Positive stage evaluate next stage, otherwise stops the loop
5. If all stages were positive store the rectangle
6. Once the tree is done, group the overlapping rectangles
7. Find the best rectangle of each the group (merging phase)

Rapid object detection (Viola Jones)

Optimized merging phase

Rectangles are used partitioned into a disjoint set data structure that was replaced by an alternative that is called Minimum Neighbor Area Grouping

Rapid object detection (Viola Jones)

Optimized merging phase

Rectangles are used partitioned into a disjoint set data structure that was replaced by an alternative that is called Minimum Neighbor Area Grouping

Minimum Neighbor Area Grouping

Simple loop trough the possible rectangles comparing the current rectangle with all other not yet compared. If their area overlaps by $\eta = 0.5$ the smallest rectangle of the set is selected