The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

An Efficient Hardware Architecture of the Optimised SIFT Descriptor Generation

Wenjuan Deng, Dr. Yiqun Zhu

Email: eexwd2@nottingham.ac.uk, Yiqun.zhu@nottingham.ac.uk

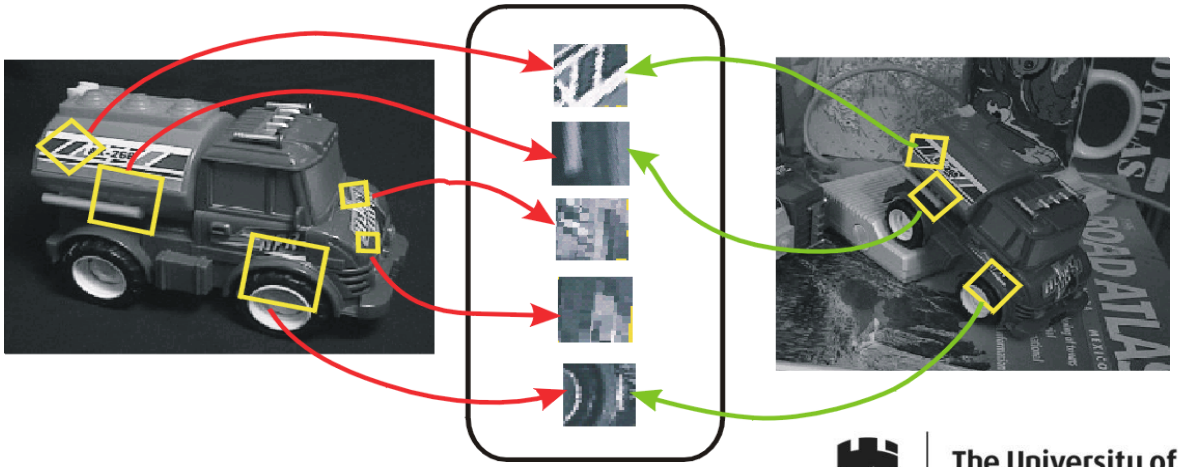
University of Nottingham

Overview

- Background Introduction
- Motivation of Work
- SIFT Algorithm
- Hardware Design
- Performance Evaluation
- Conclusions and Further Work

Background Introduction

- Image Matching
- Applications
  - Object or scene recognition
  - Object localization and mapping
  - Image fusion and registration
  - 3D reconstruction



Motivation of Work

- Challenges
  - Descriptor Generation has become the bottleneck
  - High throughput requirement for stand-alone real-time applications
- Objectives
  - Improve descriptor generation efficiency
    --- by descriptor dimension reduction
  - Computational complexity simplification
    --- by LookUp Table Technique

SIFT Algorithm

- Stands for **S**cale **I**nvariant **F**eature **T**ransform
- Extract highly distinctive invariant features

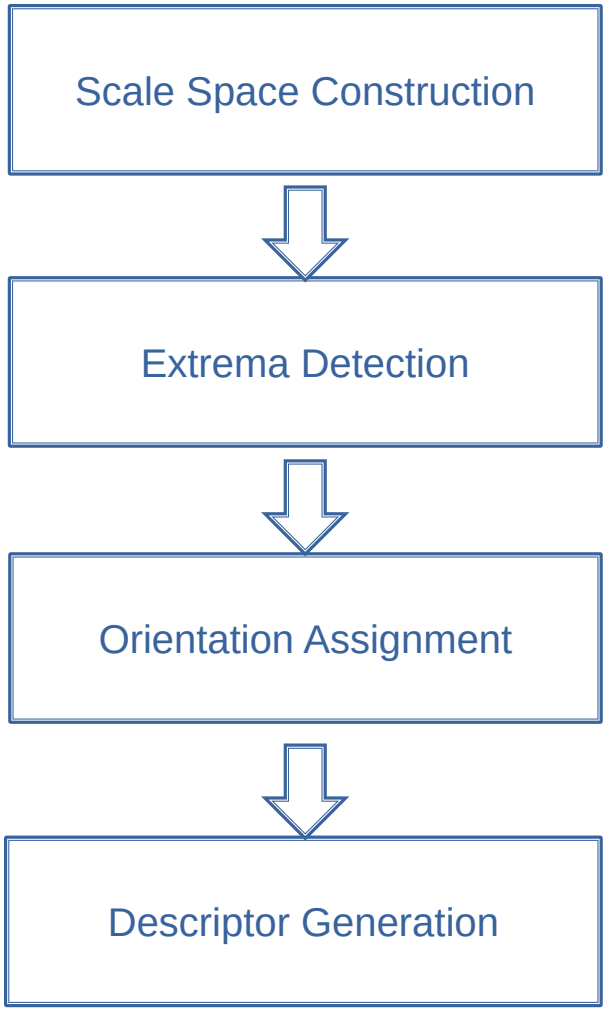
Scale and Rotation

Illumination

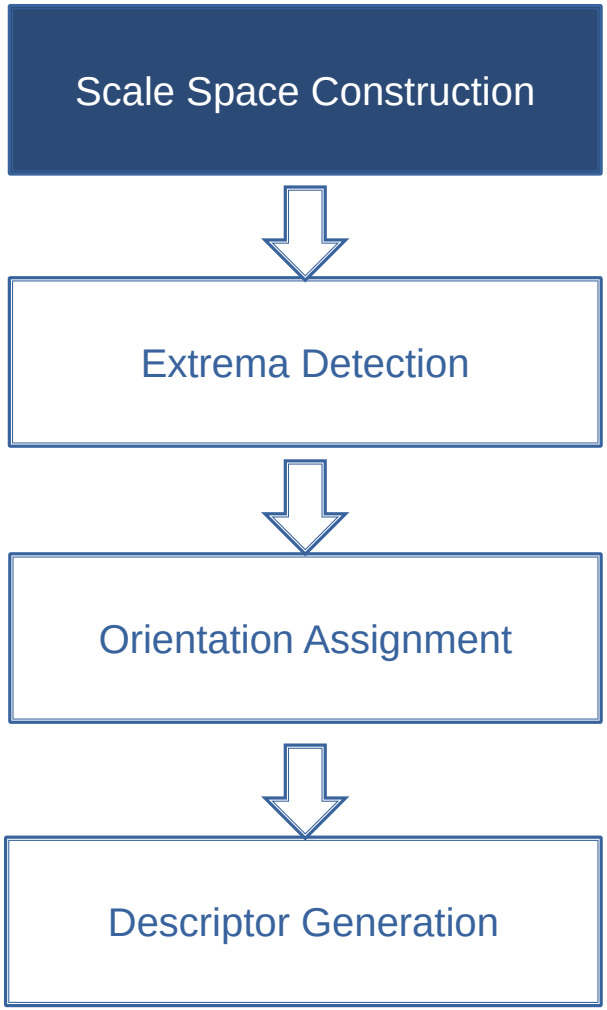
3D viewpoint

- Computation is highly complicated

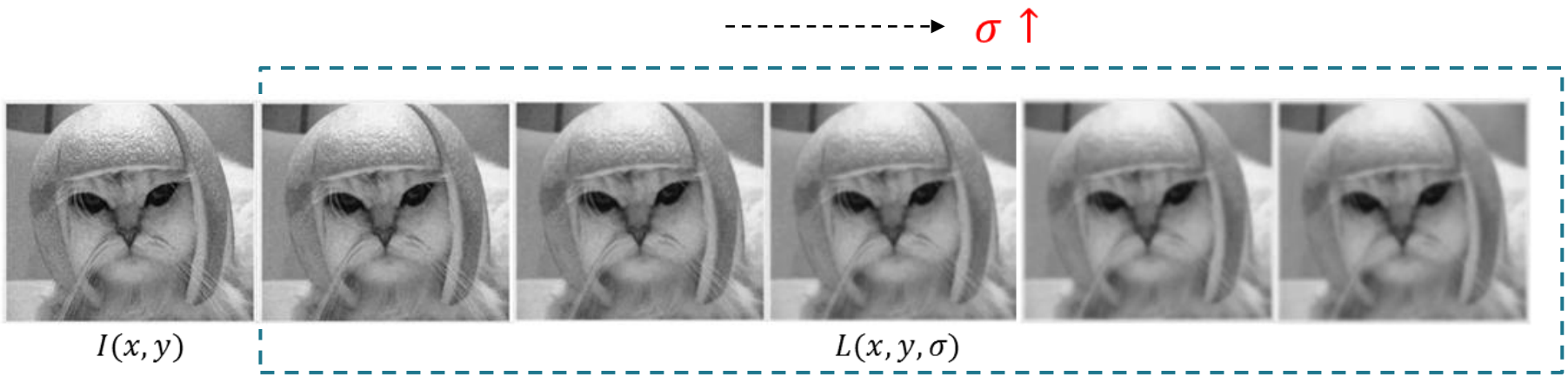
SIFT Algorithm



SIFT Algorithm



Scale Space Construction



Gaussian Scale space image

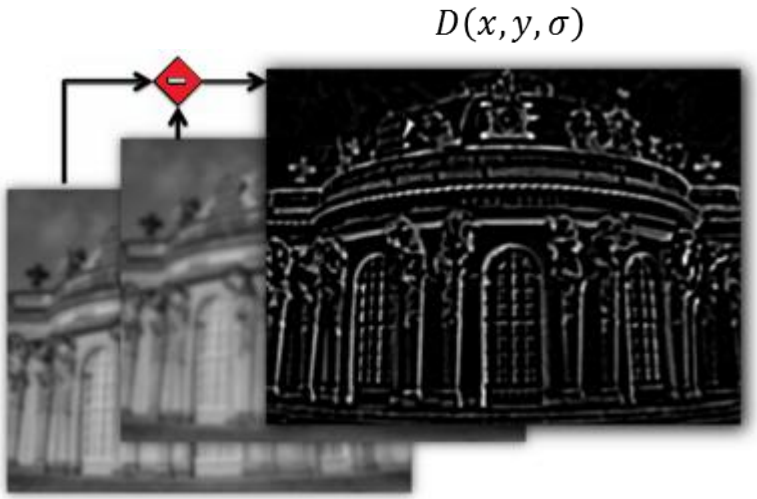
Gaussian kernel

Input image

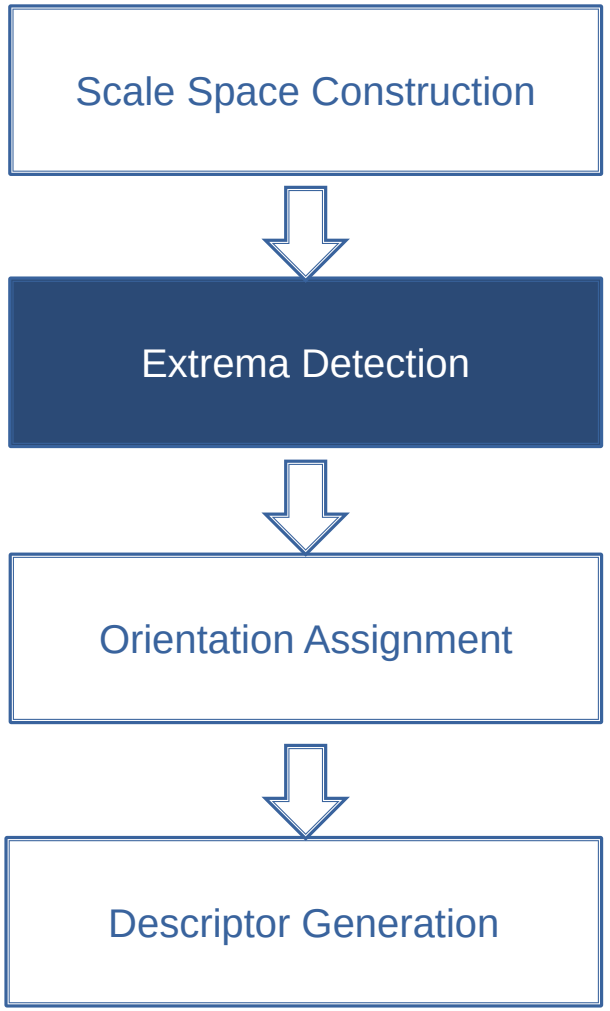
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

Difference-of-Gaussian image

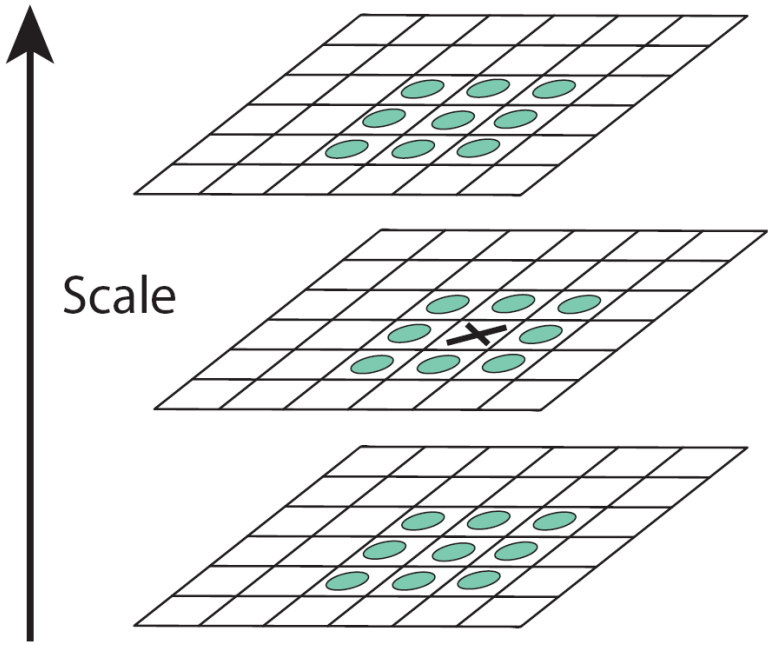
$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2)$$



SIFT Algorithm



- Extrema Detection



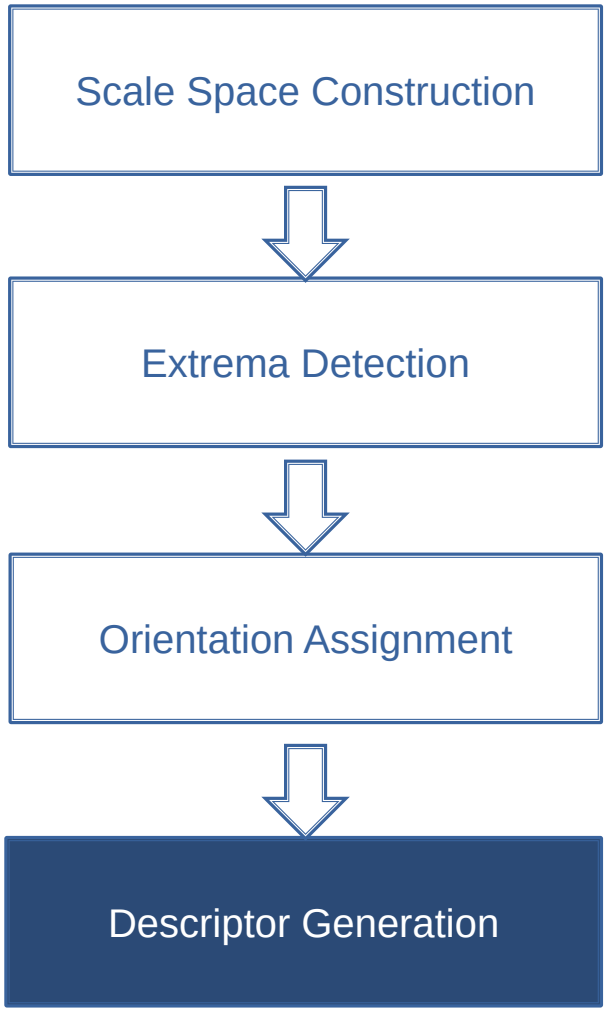
Maxima and Minima in a 3x3x3 neighbourhood

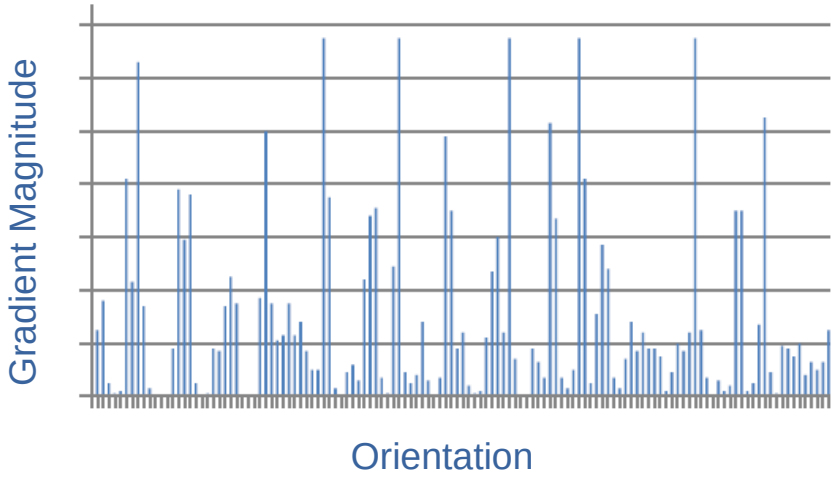
'X' represents the pixel to be compared with surrounding 26 pixels

SIFT Algorithm

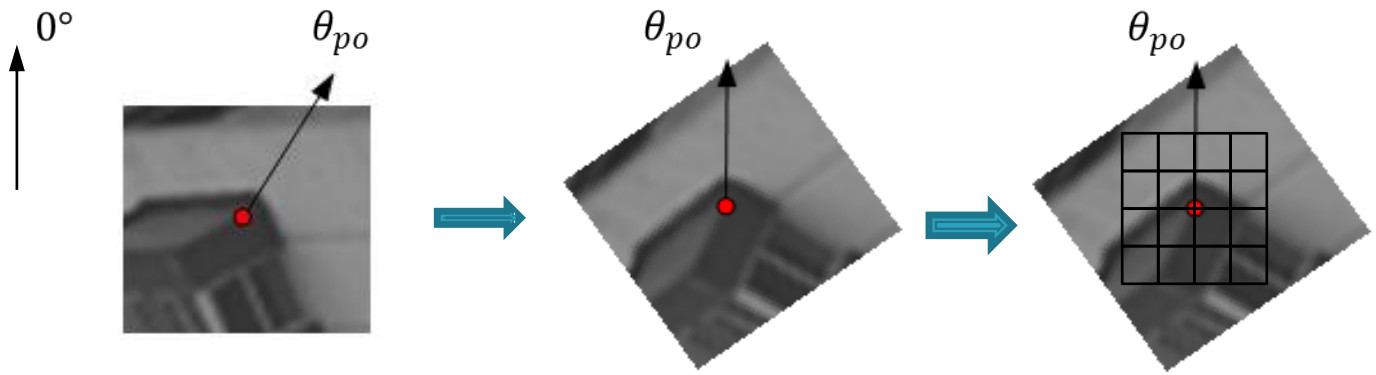
- Scale Space Construction
- Extrema Detection
- **Orientation Assignment**
- Descriptor Generation

SIFT Algorithm





Descriptor Generation of Original SIFT Algorithm



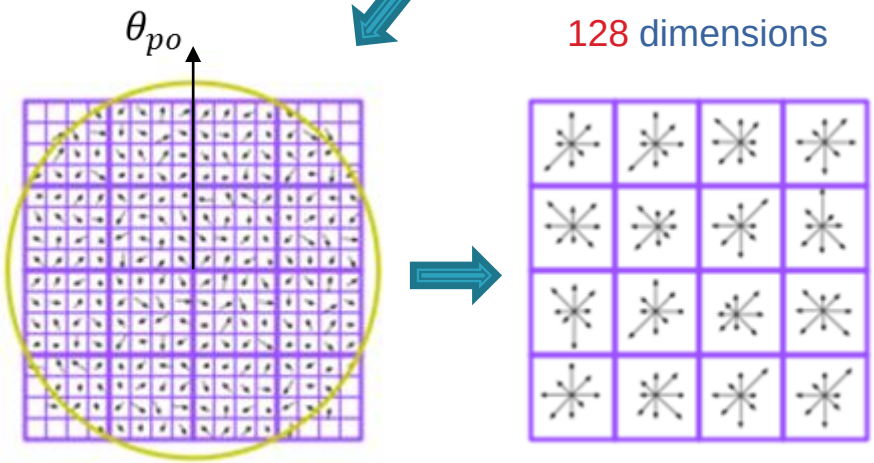
θ_{po} is the **Principal Orientation**

- Rotate Coordinates

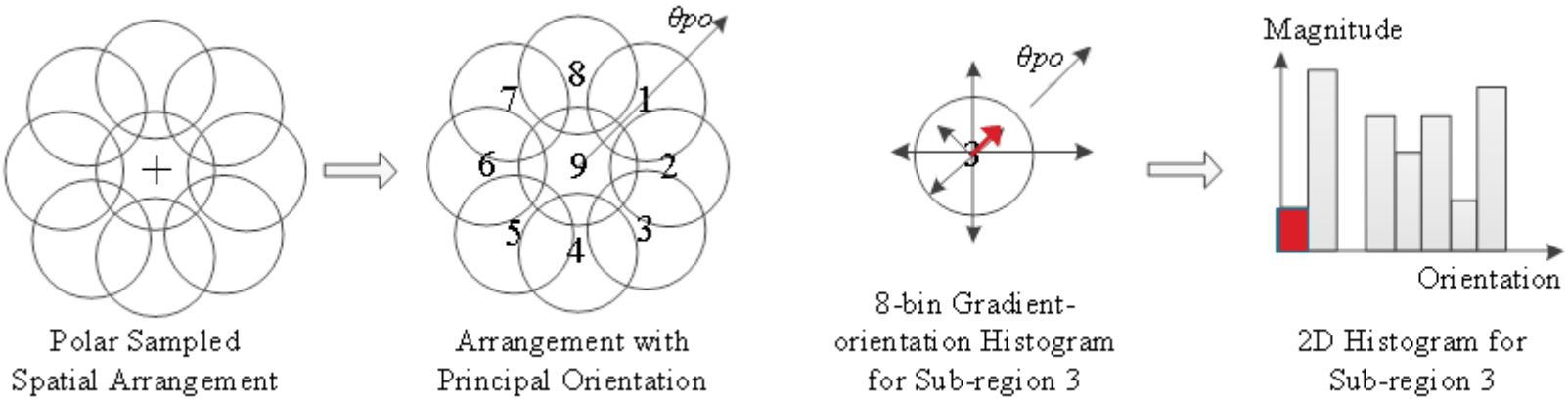
$$x' = x\cos(\theta_{po}) - y\sin(\theta_{po})$$

$$y' = y\cos(\theta_{po}) + x\sin(\theta_{po})$$

- Rotate Gradient Orientation



Descriptor Generation of our Optimised SIFT Algorithm



Polar Sampled Spatial Arrangement

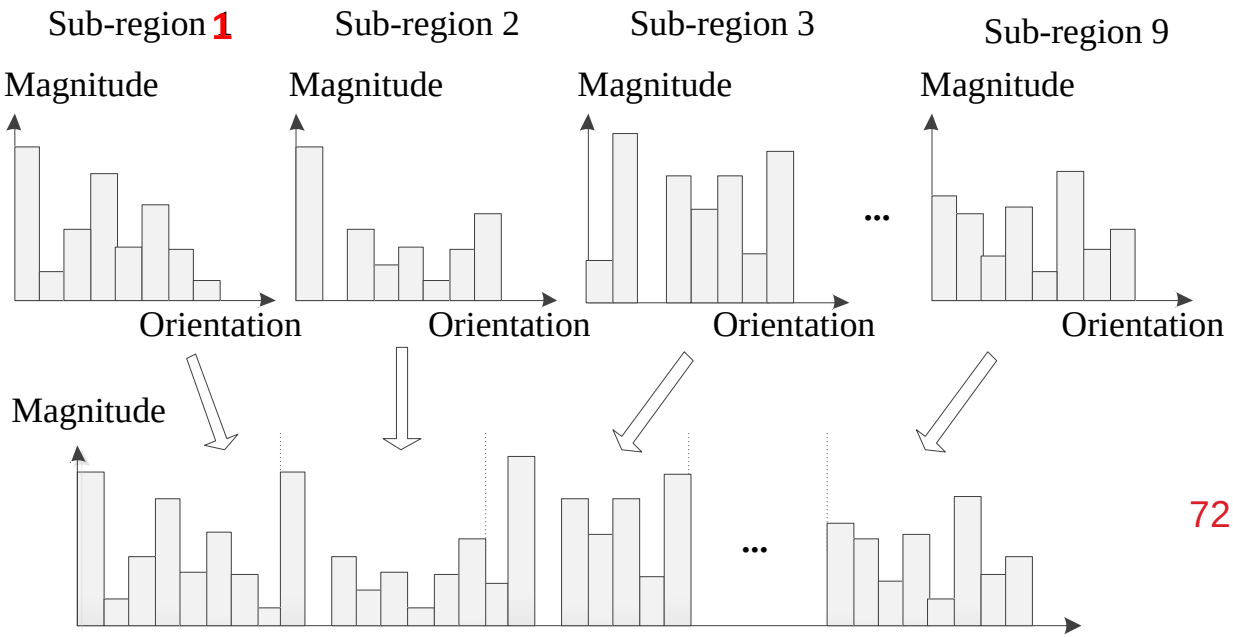
Arrangement with Principal Orientation

8-bin Gradient-orientation Histogram for Sub-region 3

2D Histogram for Sub-region 3

- Step 1: Arrangement of surrounding sub-regions
- Step 2: Arrangement of gradient histogram of each sub-region

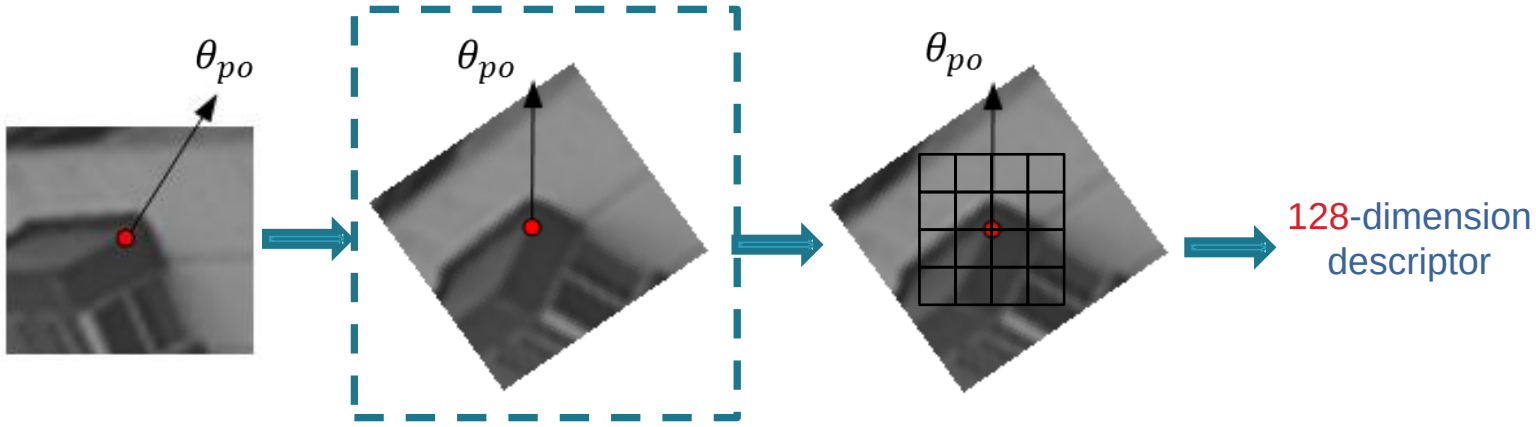
Descriptor Generation of our Optimised SIFT Algorithm



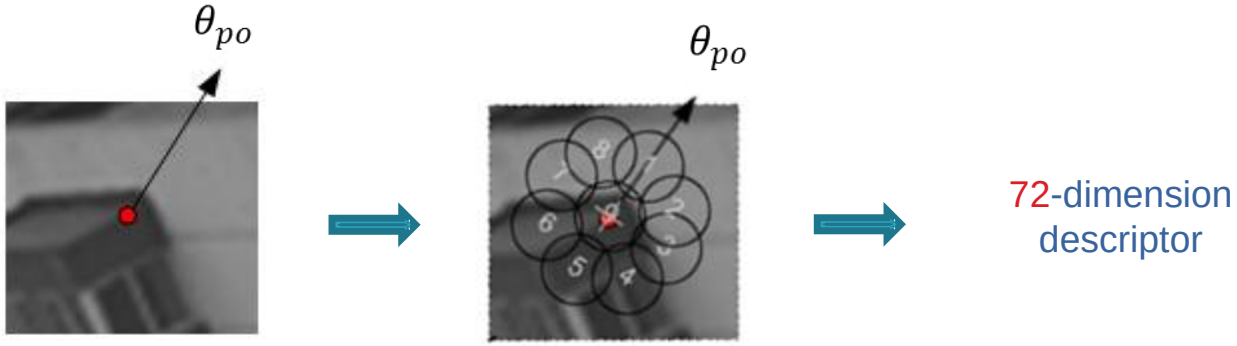
72 dimensions

- Step 3: Linking together the histograms of 9 sub-regions to generate a descriptor

- Original SIFT



- Optimised SIFT



Hardware Design

- Overall Hardware Architecture
- Key Optimisation
  - LookUp Table

Overview

Background Introduction

Motivation of Work

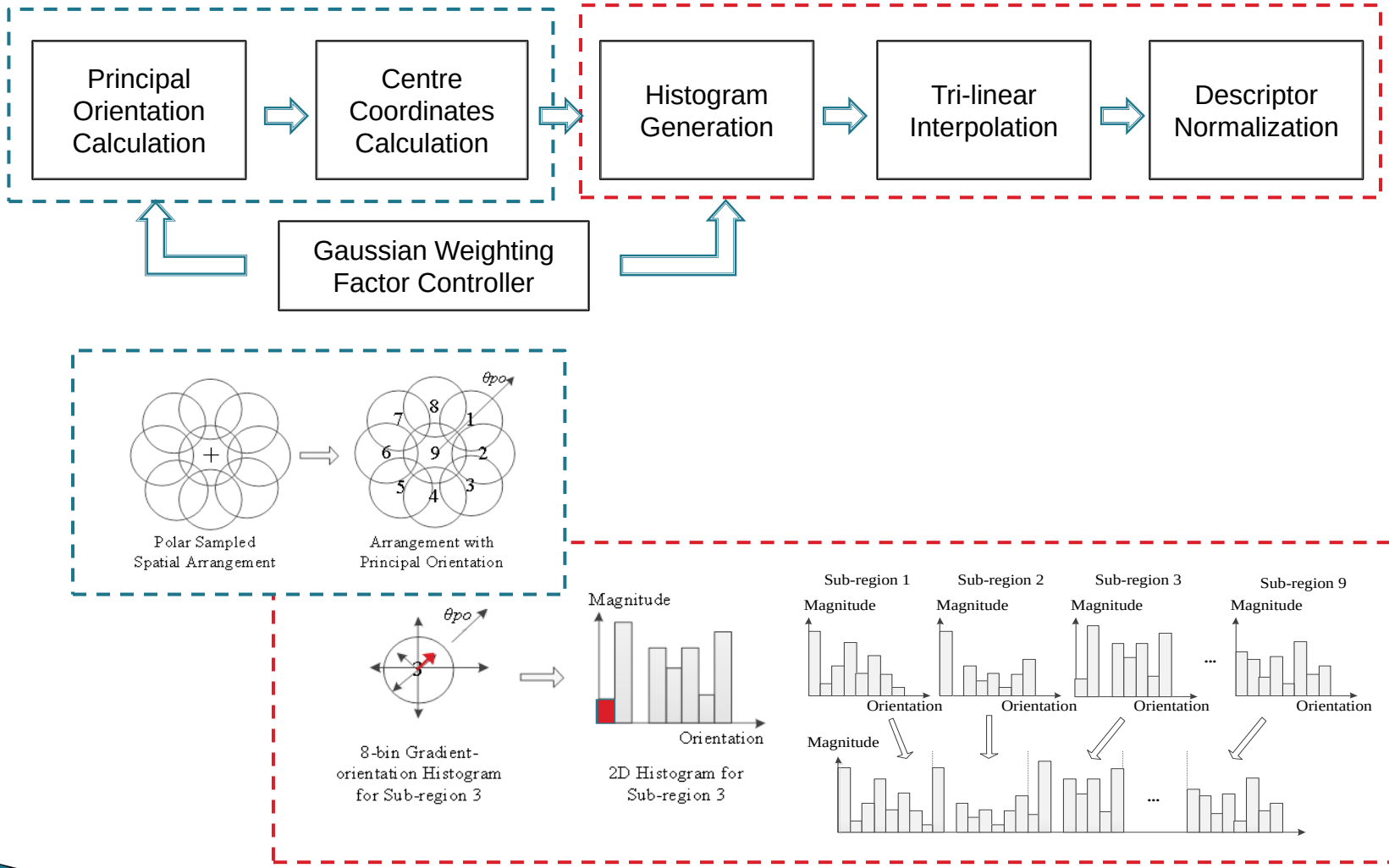
SIFT Algorithm

Hardware Design

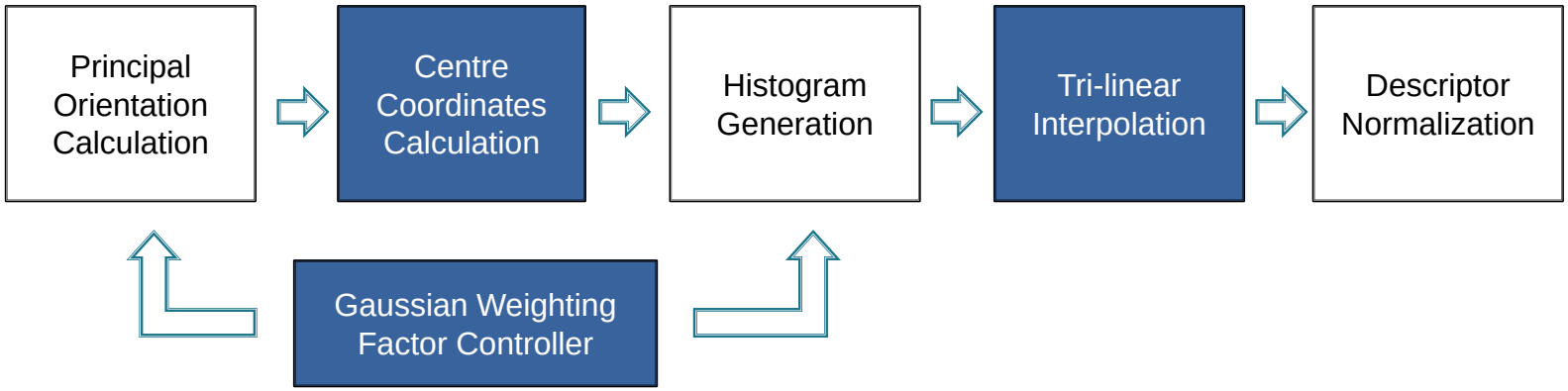
Performance Evaluation

Conclusions and Further Work

Overall Hardware Architecture

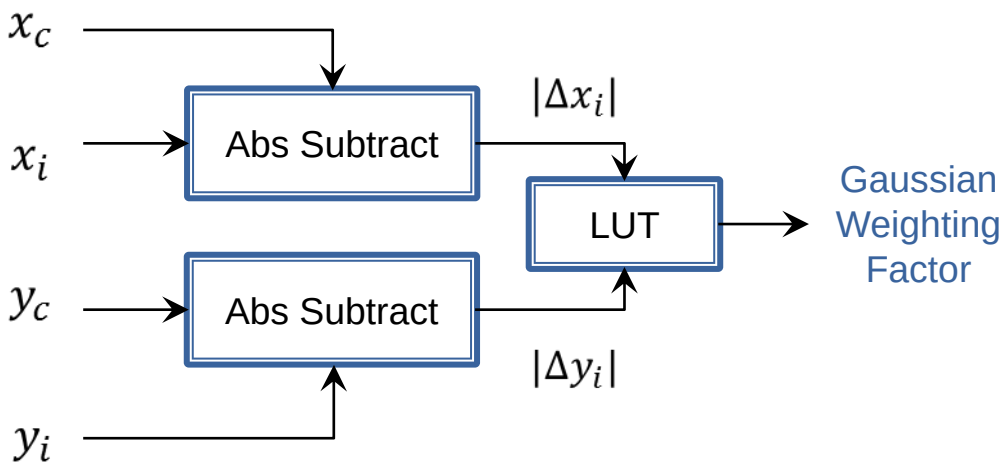


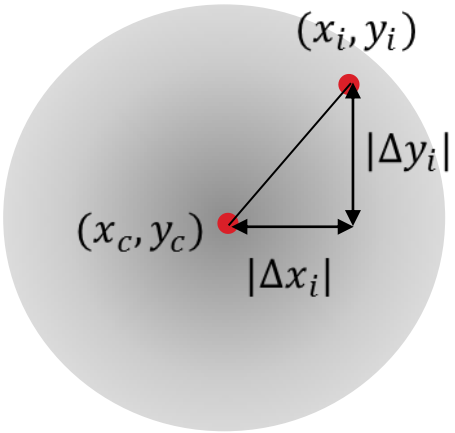
- Key Optimisation
  - LookUp Table



Runtime Computation ⇒ Array Indexing

Gaussian Weighting Factor Controller





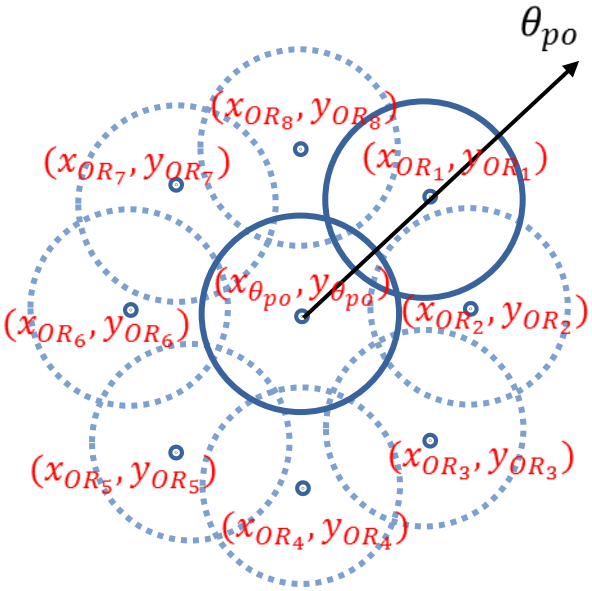
(x_c, y_c) : coordinates of centre pixel

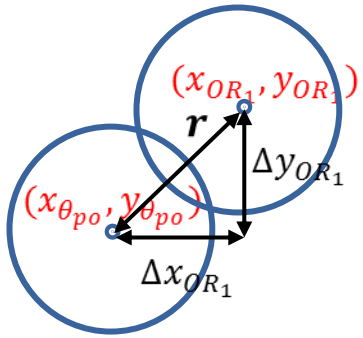
(x_i, y_i) : coordinates of a pixel to be weighted

Centre Coordinate Calculation

Table 1. Offsets from the centre coordinates of sub-regions to the feature point in both x and y directions

Sub-region	Δx_{OR_i}	Δy_{OR_i}
1	$\mathrm{r} * \cos(\theta_{po})$	$\mathrm{r} * sin(\theta_{po})$
2	$\frac{\sqrt{2}}{2} r * (\cos(\theta_{po}) - \sin(\theta_{po}))$	$\frac{\sqrt{2}}{2} r * (\cos(\theta_{po}) + \sin(\theta_{po}))$
3	$-\mathrm{r} * sin(\theta_{po})$	$\mathrm{r} * \cos(\theta_{po})$
4	$-\frac{\sqrt{2}}{2} r * (\cos(\theta_{po}) + \sin(\theta_{po}))$	$\frac{\sqrt{2}}{2} r * (\cos(\theta_{po}) - \sin(\theta_{po}))$
5	$-\mathrm{r} * \cos(\theta_{po})$	$-\mathrm{r} * sin(\theta_{po})$
6	$\frac{\sqrt{2}}{2} r * (\sin(\theta_{po}) - \cos(\theta_{po}))$	$-\frac{\sqrt{2}}{2} r * (\cos(\theta_{po}) + \sin(\theta_{po}))$
7	$\mathrm{r} * sin(\theta_{po})$	$-\mathrm{r} * \cos(\theta_{po})$
8	$\frac{\sqrt{2}}{2} r * (\cos(\theta_{po}) + \sin(\theta_{po}))$	$\frac{\sqrt{2}}{2} r * (\sin(\theta_{po}) - \cos(\theta_{po}))$
9	0	0

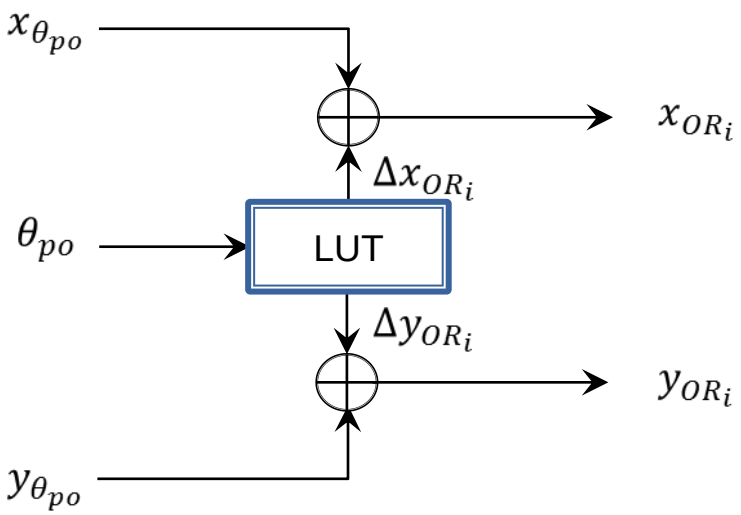




$(x_{\theta_{po}}, y_{\theta_{po}})$: coordinates of the feature point

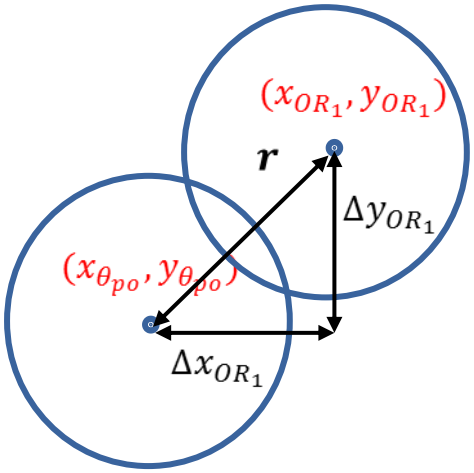
(x_{OR_i}, y_{OR_i}) : centre coordinates of surrounding sub-regions

- Centre Coordinate Calculation



$$x_{OR_i} = x_{\theta_{po}} + \Delta x_{OR_i}$$

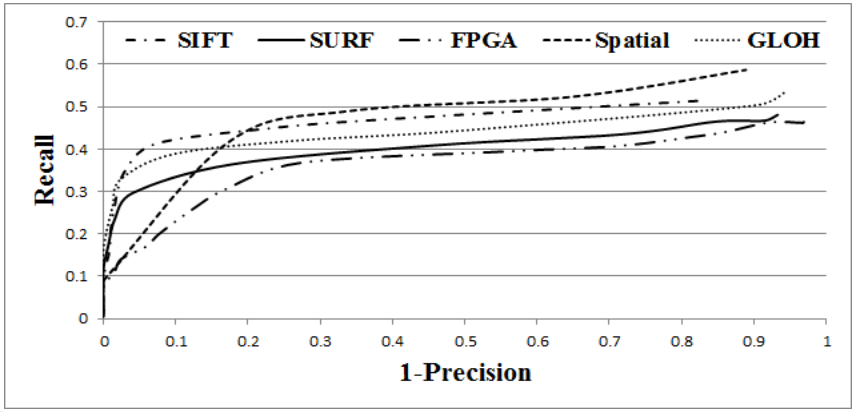
$$y_{OR_i} = y_{\theta_{po}} + \Delta y_{OR_i}$$



Performance Evaluation

- Matching Performance
  - Matching results

  - Matching performance comparison



Hardware Efficiency

Table 1. Hardware Resource Usage and Throughput Comparison of Different Hardware Designs

	Y. Lin's [2010]	K. Mizuno's [2010]		F. –C. Huang's [2012]	Our design
		High-accuracy Mode	High-speed Mode		
Frame Size	N/A	VGA			
Frequency (MHz)	200	50		100	**100**
Registers	N/A	23,247		N/A	**29,453**
LUTs		32,592			**64,701**
DSP		258			**107**
Memory Usage (Mb)		0.87	0.67	4.86	**3.84**
Time Consumption per Descriptor (us)	15.315	N/A		33.1	**7.57**
Frame Rate (fps)	N/A	32	56	30	**at least 60**

Conclusions and Further Work

- Conclusions
  - 7.57 us per descriptor @100MHz
  - 132,100 descriptors per second
  - 2,200 descriptors/frame @60 VGA fps
  - Real-time processing of even higher resolution images
- Further Work
  - FPGA prototype of a dual-camera image matching system
  - A real-time vision system for visual prosthesis simulator

Thank you!