Adaptive Image Compression Using Saliency and KAZE Features

Authors: Siddharth Srivastava, Prerana Mukherjee, Dr. Brejesh Lall

Department of Electrical Engineering
Indian Institute of Technology, Delhi

Overview

- **Introduction**
- **Background**
- **Proposed Methodology**
- **Experimental Results and Discussions**
- **Conclusion**

Introduction

- **Primary Objectives**
  - To achieve image compression based on importance of the contents in the image.
  - Utilize properties from local regions to identify region importance.

Introduction

- **Underlying Principles**
  - **Differed Importance**: The human eye gives more importance to salient regions
  - **Transition**: The surrounding regions around salient objects make them distinguishable
  - **Rejection**: Perceptually less significant Pixels/Regions can be made further insignificant
  - **Object Characterization**
    - Well defined boundary
    - Distinctive appearance
    - Uniqueness

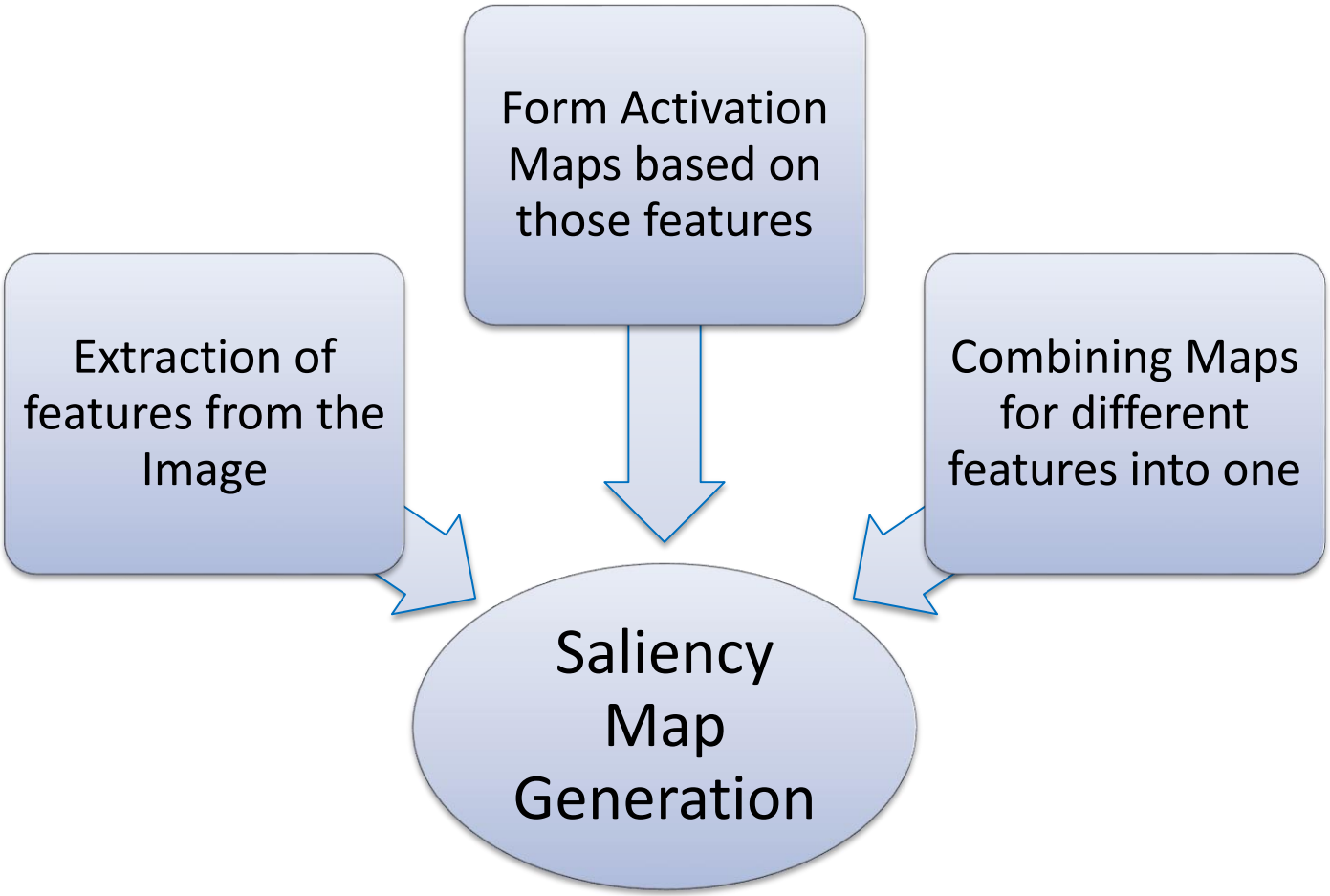
Introduction

- **Core Methodology**
  - We adapt the quality factor in JPEG compression scheme for each block instead of a global quality factor
  - This adaptation of quality parameter is based on saliency map and strength of KAZE keypoints

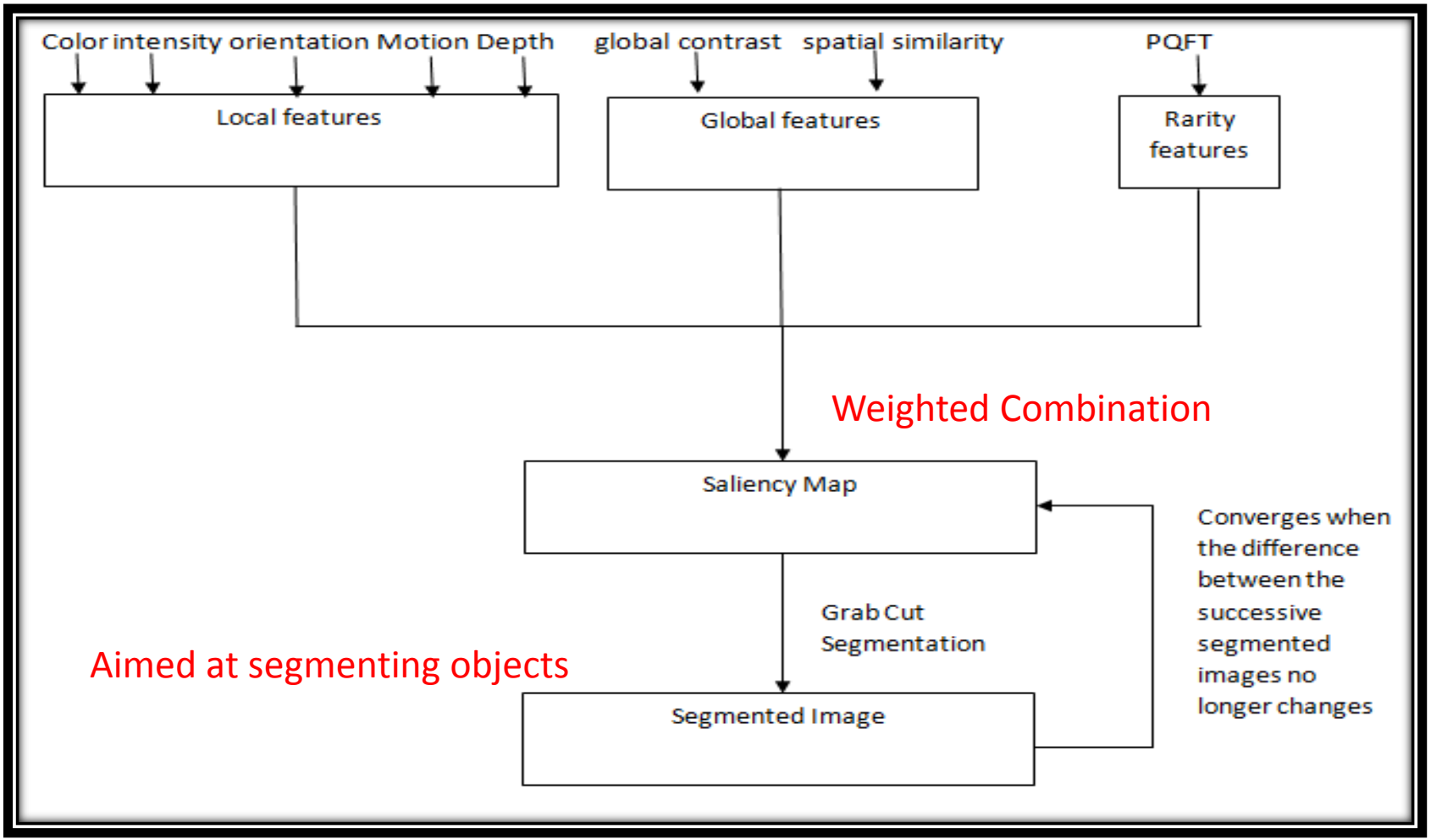
Introduction

- **Key Contributions:**
  - JPEG Compatible
  - First to introduce KAZE for image compression
  - Maintains better perceptual quality at high compression ratios

Background: Saliency Map



Background: Saliency Map Computation



P. Mukherjee, B. Lall, and A. Shah, "Saliency map based improved segmentation," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015.

Background: KAZE

- KAZE is a recent feature detection technique which exploits the non linear scale space to detect keypoints along edges and sharp discontinuities.
- Non linear diffusion filtering allows KAZE to achieve less blurring on edges as compared to Gaussian Blurring.

Saliency Map and KAZE

a) b) c)

Fig. 1: (a) Original Image (b) Saliency Map (c) KAZE keypoints

Proposed Methodology

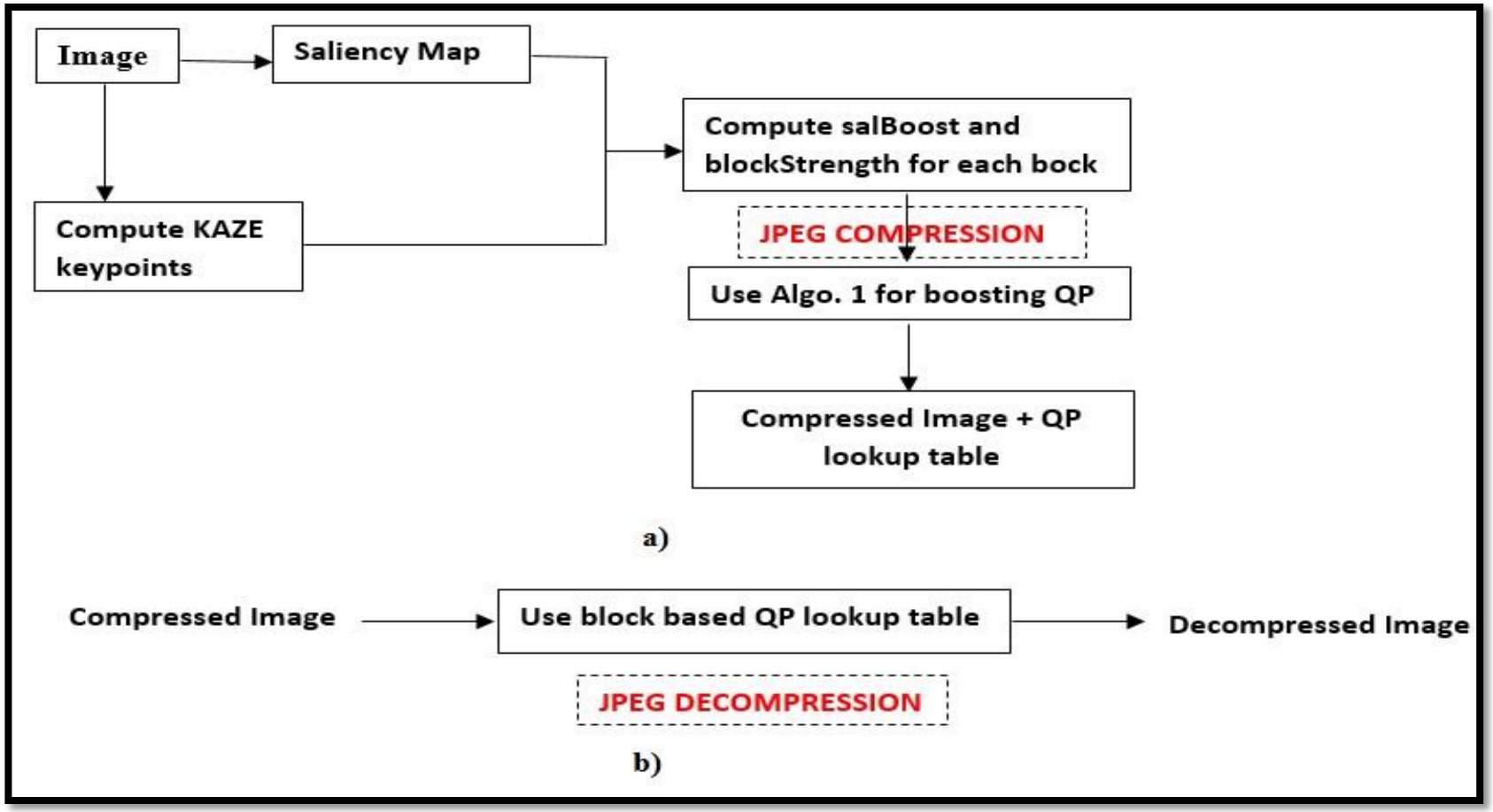


Fig. 2: Architecture of (a) Compression and (b) Decompression

Adapting Quality Parameter (QP) with Saliency Response

$$salBoost(i) = \sqrt{\frac{mean(block\_sal(i))}{mean(salmap)}}$$

$$Q_{Sal}(i) = Q_{JPEG} * salBoost(i)$$

i: the 8x8 block in the image

M. T. Khanna, K. Rai, S. Chaudhury, and B. Lall, "Perceptual depth preserving saliency based image compression," in *Proceedings of the 2nd International Conference on Perception and Machine Intelligence*. ACM, 2015, pp. 218–223.

Adapting Quality Parameter (QP) with Keypoint Response

$$blockStrength(i) = \sqrt{\frac{mean(block\_response(i))}{mean(image\_response)}}$$

i: the 8x8 block in the image

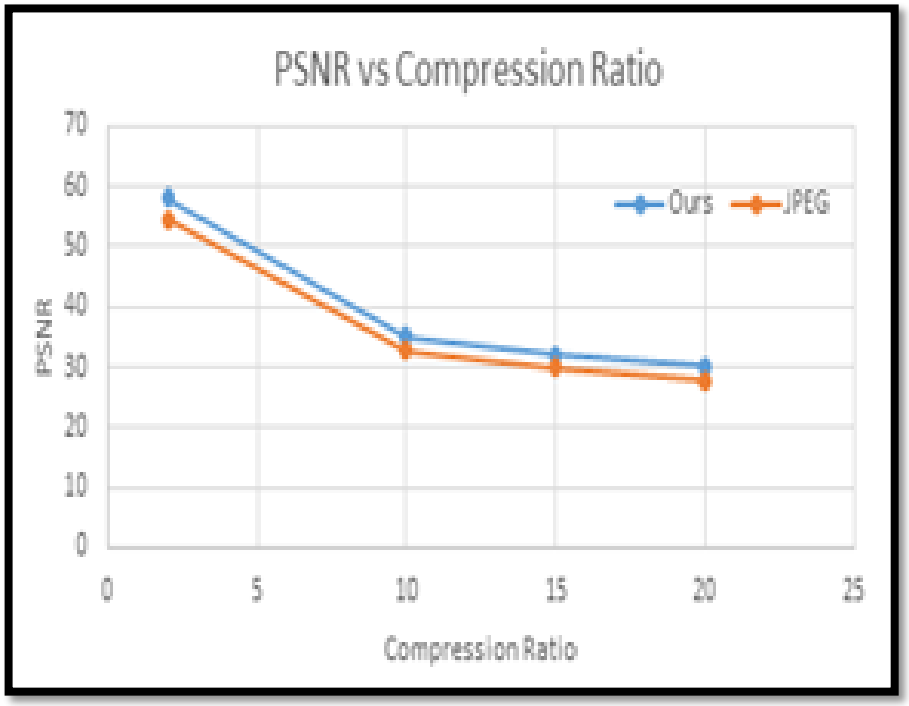
Algorithm I: Algorithm for Piecewise Adaptive QP

- $Q_{Sal} = Q_{JPEG} * salBoost$
- $M_R = mean(image\_response)$
- $B_S = blockStrength$

$$boostedQP = \begin{cases} \beta 1 * Q_{Sal} & \text{if } B_S = 0 \text{ and } salBoost < 1 \\ Q_{Sal} & \text{if } B_S = 0 \text{ and } salBoost \geq 1 \\ & \text{or } B_S \geq \alpha * M_R \text{ and } B_S < (1-\alpha) * M_R \\ (1-\alpha) * Q_{Sal} & \text{if } \alpha * M_R \\ (1+\alpha) * Q_{Sal} & \text{if } B_S < (1-\alpha) * M_R \text{ and } B_S \leq (1+\alpha) * M_R \\ \beta 2 * Q_{Sal} & \text{otherwise} \end{cases}$$

*$0 < \alpha \leq 0.5$, $\beta 1 < 1$ and $\beta 2 > 1$

Results



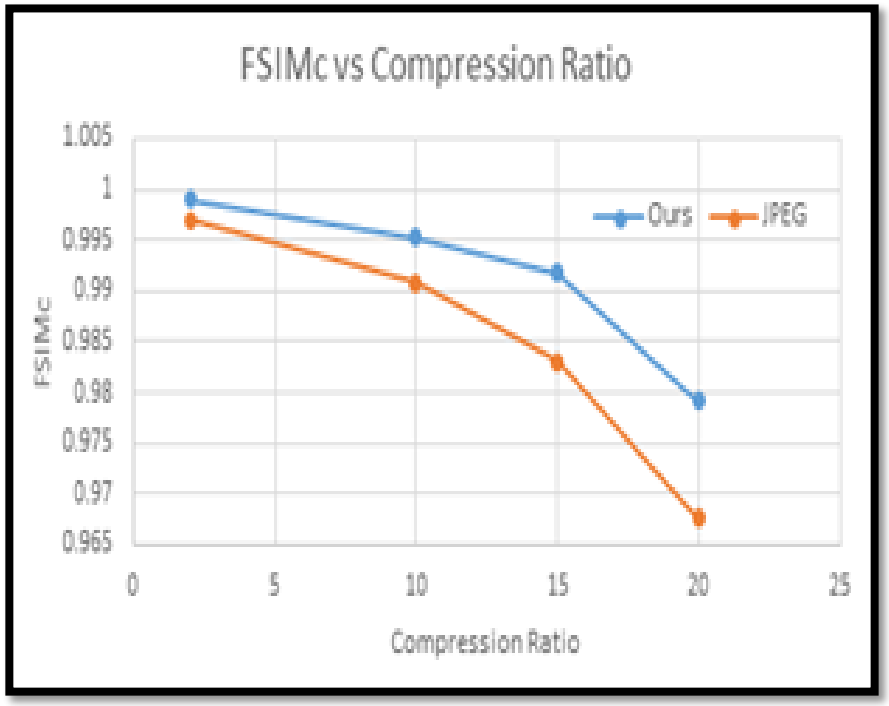


Fig. 3: (a) Plot showing the change in PSNR rate as the compression ratio changes (b) Plot between FSIMc with the varying compression ratio

Results

a) b) c)

Fig. 4: (a) Original Image (b) Results after JPEG compression (c) Results after Adaptive Compression (Proposed Approach)

References

- P. Mukherjee, B. Lall, and A. Shah, “Saliency map based improved segmentation,” in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015.
- M. T. Khanna, K. Rai, S. Chaudhury, and B. Lall, “Perceptual depth preserving saliency based image compression,” in *Proceedings of the 2nd International Conference on Perception and Machine Intelligence*. ACM, 2015, pp. 218–223.
- F. Alcantarilla, A. Bartoli, and A. J. Davison, “Kaze features,” in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 214–227.

Questions ?

THANK YOU

KAZE: Background

Nonlinear Diffusion Filtering

- **Nonlinear diffusion approaches** describe the **evolution of the luminance of an image** through **increasing scale levels** as the divergence of a **flow** function that controls the diffusion process

$$\frac{\partial L}{\partial t} = div(c(x, y, t) \cdot \nabla L) \quad ($$

- The function c depends on the **local image differential structure**, and this function can be either a **scalar** or a **tensor**
- The time t is the **scale parameter**, and larger values lead to simpler image representations

KAZE: Background

Perona and Malik Diffusion Equation

- Function c dependent on the **gradient magnitude**
- **Reduce diffusion at edges location**, encouraging smoothing within a region instead of smoothing across boundaries

$$c(x, y, t) = g(|\nabla L_\sigma(x, y, t)|)$$

- Two different formulations for the **conductivity function** g
  - g_1 promotes **high-contrast edges**
  - g_2 promotes **wide regions over smaller ones**

$$g_1 = \exp\left(-\frac{|\nabla L_\sigma|^2}{k^2}\right)$$

$$g_2 = \frac{1}{1+\frac{|\nabla L_\sigma|^2}{k^2}}$$

KAZE: Background

- The **contrast factor** k is computed **empirically** as the 70% percentile of the gradient histogram of a smoothed version of the original images
  - It can be also set by hand or by some learning
- If the **conductivity function** c is **constant**, we obtain the **heat equation**, i.e. **linear diffusion**
- **Rapidly decreasing diffusivities**
  - Smoothing on both sides of an edge is much stronger than smoothing across it

$$g_3 = \begin{cases} 1 & , \quad |\nabla L_\sigma|^2 = 0 \\ 1 - \exp\left(-\frac{3.315}{(|\nabla L_\sigma|/k)^8}\right) & , \quad |\nabla L_\sigma|^2 > 0 \end{cases} \tag{9}$$

KAZE: Background

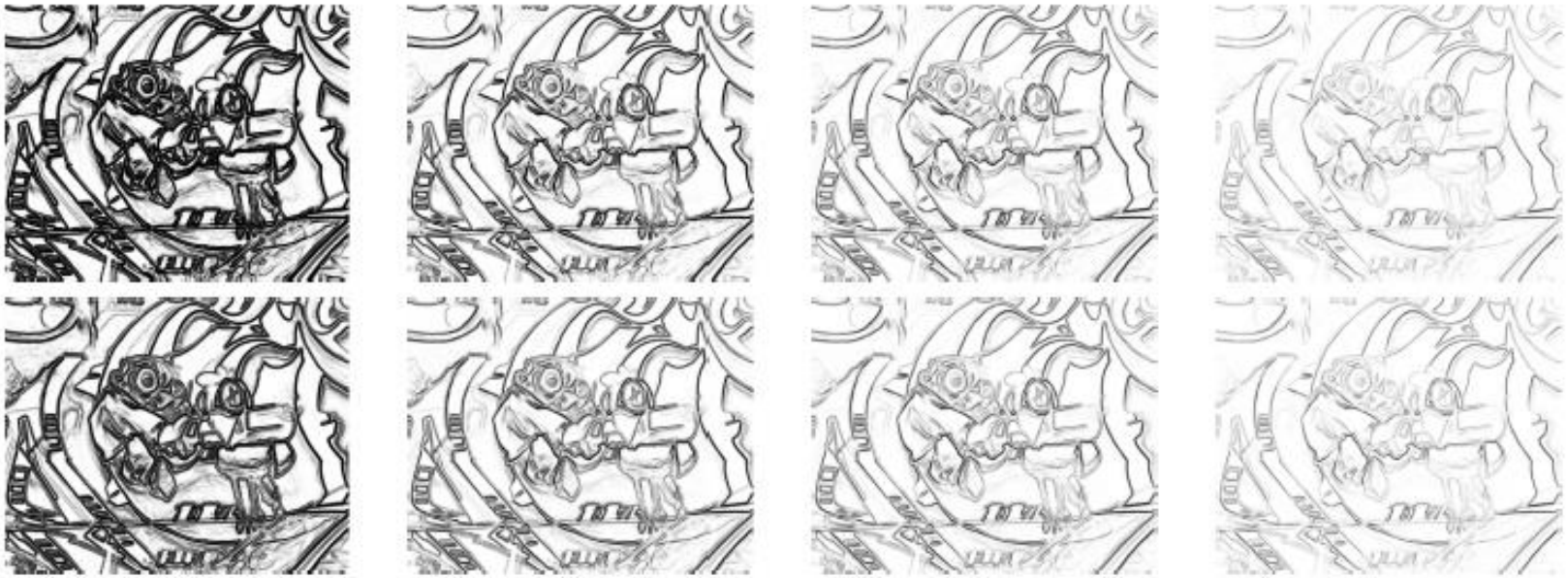


Figure: First row: g_1 conductivity function. Second row: g_2 conductivity function. Notice that for increasing values of k only higher gradients are considered.

KAZE: Background

Additive Operator Splitting (AOS)

- Modification of the **semi-implicit scheme**

$$\frac{L^{i+1} - L^i}{\tau} = \sum_{l=1}^{m} A_l(L^i) L^{i+1} ,$$

- The solution L^{i+1} can be obtained as:

$$L^{i+1} = \left(I - \tau \sum_{l=1}^{m} A_l(L^i) \right)^{-1} L^i$$

- Now the total diffusion is the **addition of two 1D diffusion processes**
- The matrix A_l encodes the diffusivities for each image dimension

KAZE: Keypoint Detection

- Steps in **KAZE Detection**:
  - Build **nonlinear scale space** using **AOS** and a set of **octaves** O and **sublevels** S

$$\sigma_i(o,s) = \sigma_0 2^{o+s/S},\ o \in [0 \dots O-1],\ s \in [0 \dots S-1],\ i \in [0 \dots N]$$

  - We need to map **scale units** to **time units**:

$$t_i = \frac{1}{2}\sigma_i^2,\ i = \{0 \dots N\}$$

$$L^{i+1} = \left(I - (t_{i+1} - t_i) \cdot \sum_{l=1}^{m} A_l\left(L^i\right)\right)^{-1} L^i\ .$$

equation for building non linear scale space using AOS

Non linear vs linear scale space

Comparison between gaussian blurring and nonlinear diffusion

KAZE: Keypoint Detection

$$L_{Hessian} = \sigma^2 \left(L_{xx} L_{yy} - L_{xy}^2 \right)$$

we analyze the detector response at different scale levels σ_i. We search for maxima in scale and spatial location. The search for extrema is performed in all the filtered images except $i = 0$ and $i = N$. Each extrema is searched over a rectangular window of size $\sigma_i \times \sigma_i$ on the current i, upper $i + 1$ and lower $i - 1$ filtered images.

The set of first and second order derivatives are approximated by means of 3×3 Scharr filters of different derivative step sizes σ_i. Second order derivatives are approximated by using consecutive Scharr filters in the desired coordinates of the derivatives.

KAZE: Keypoint Detection

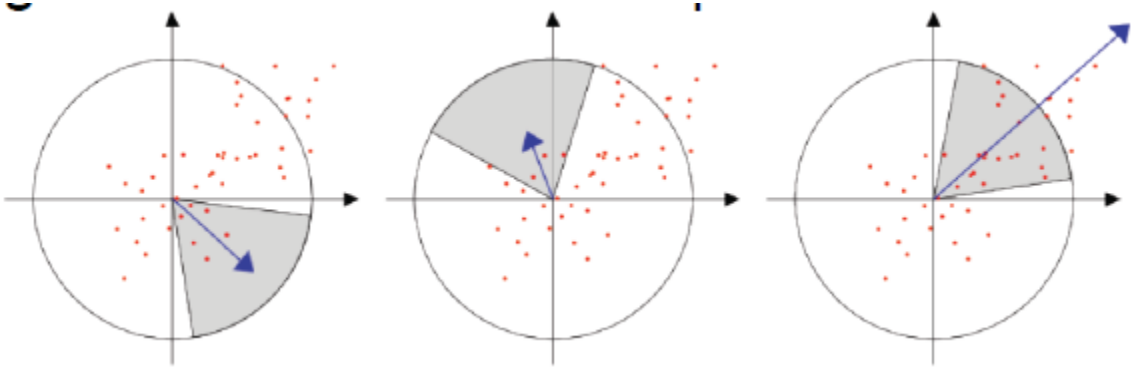
Scharr edge filter

The Scharr operator is the most common technique with two kernels used to estimate the two dimensional second derivatives horizontally and vertically. The operator for the two direction is given by the following formula:

$$K_x = \begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}, K_y = \begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}.$$

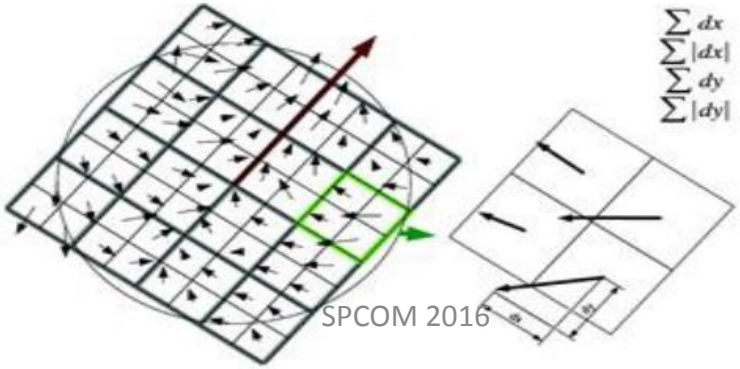
KAZE: Keypoint Description

Similar to SURF, we find the dominant orientation in a circular area of radius $6\sigma_i$ with a sampling step of size σ_i. For each of the samples in the circular area, first order derivatives L_x and L_y are weighted with a Gaussian centered at the interest point. Then, the derivative responses are represented as points in vector space and the dominant orientation is found by summing the responses within a sliding circle segment covering an angle of $\pi/3$. From the longest vector the dominant orientation is obtained.



KAZE: Keypoint Description

Building the Descriptor. We use the M-SURF descriptor adapted to our nonlinear scale space framework. For a detected feature at scale σ_i, first order derivatives L_x and L_y of size σ_i are computed over a $24\sigma_i \times 24\sigma_i$ rectangular grid. This grid is divided into 4×4 subregions of size $9\sigma_i \times 9\sigma_i$ with an overlap of $2\sigma_i$. The derivative responses in each subregion are weighted with a Gaussian ($\sigma_1 = 2.5\sigma_i$) centered on the subregion center and summed into a descriptor vector $d_v = (\sum L_x, \sum L_y, \sum |L_x|, \sum |L_y|)$. Then, each subregion vector is weighted using a Gaussian ($\sigma_2 = 1.5\sigma_i$) defined over a mask of 4×4 and centered on the interest keypoint. When considering the dominant orientation of the keypoint, each of the samples in the rectangular grid is rotated according to the dominant orientation. In addition, the derivatives are also computed according to the dominant orientation. Finally, the descriptor vector of length 64 is normalized into a unit vector to achieve invariance to contrast.



Piecewise function for further adapting QP

Algorithm 1 Algorithm for Piecewise Adaptive QP

procedure BOOSTING–QP[a]
  $Q_{Sal} = Q_{JPEG} * salBoost$
  $M_R = mean(image\_response)$ [ref. Sec III-C]
  $B_S = blockStrength$ [ref. Equation 6]
  if ($B_S == 0$ and $salBoost < 1$)
    boostedQP = $\beta 1 * Q_{Sal}$;
  else if ($B_S == 0$ and $salBoost \geq 1$)
    boostedQP = Q_{Sal};
  else if ($B_S < \alpha * M_R$)
    boostedQP = $(1 - \alpha) * Q_{Sal}$;
  else if ($B_S \geq \alpha * M_R$ and $B_S < (1 - \alpha) * M_R$)
    boostedQP = Q_{Sal};
  else if($B_S \leq (1 - \alpha) * M_R$ and $B_S \leq (1 + \alpha) * M_R$)
    boostedQP = $(1 + \alpha) * Q_{Sal}$;
  else
    boostedQP = $\beta 2 * Q_{Sal}$;
  end if
end procedure

[a] $0 < \alpha \leq 0.5$, $\beta 1 < 1$ and $\beta 2 > 1$