Vision Images/videos

Image Features

NGUYEN Hong Thinh

FET-UET-VNU

Ngày 28 tháng 3 năm 2020

Today:

- What is features?
- Global features vs Local Features
- Global features: Color features
- Local features: Image Matching
- HOG feature
- SIFT-BOW method for image classification
- Conclusion

Large volume of image database

Content-based Image Retrieval

Given a query image, try to find visually similar images from an image database

Comparing 2 images: challenging problems?

- Different size/resolution
- Different viewpoint/pose
- Different light condition
- Occlusion (Che khuất)
- Intra class variation
- Inter class variation

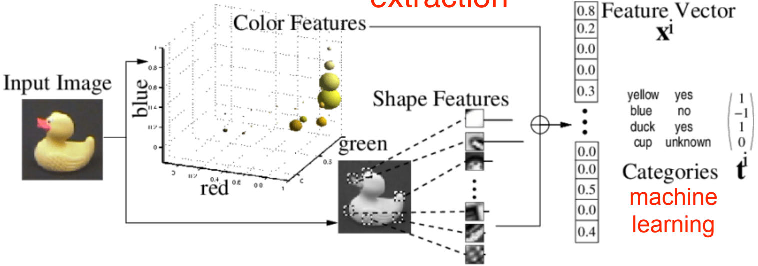
Không thể so sánh trực tiếp 2 ảnh ⇒ So sánh dựa trên đặc điểm /đặc trưng trong ảnh

-tiêu chuẩn

Image features?

- Features (đặc trưng): là tất cả các đặc điểm, đặc trưng có thể sử dụng để phân biệt các đối tượng với nhau (vd color, shape)
- Features vector: vector chứa thông tin của tất cả đặc trưng của ảnh.

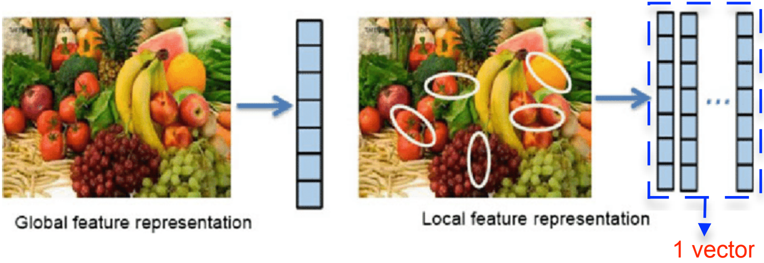
features extraction



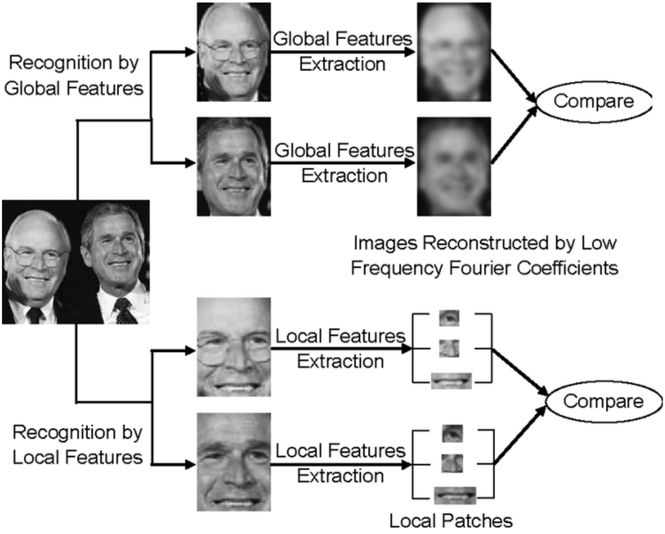
machine learning

Phân loại đặc trưng ảnh

- Global features: One image => 1 feature
- Local featuers: Block based features: Images= set of blocks => features



Phân loại đặc trưng ảnh



Popular features

- Global features:
  - Color based features: CH, CM, Corr, CN
  - Shape features
  - GIST features
- Local features:
  - Interesting point detectors
  - Dense sampling
  - Descriptors: SIFT, SUFT, HOG, BRIEF, ORB...

open-cv

Global features: Colors features

Color histogram

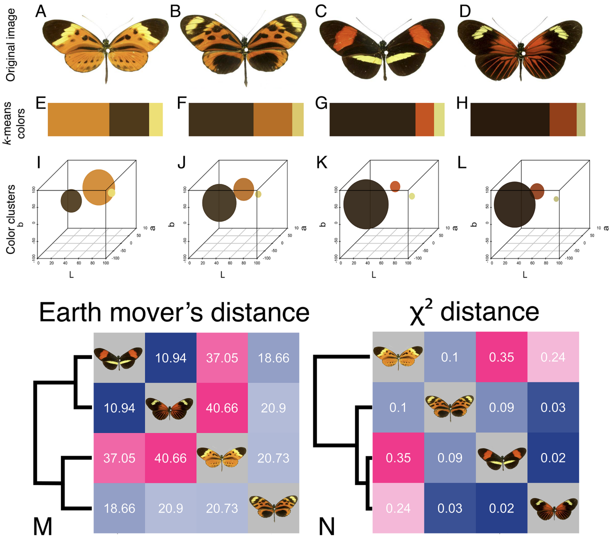
Cam Đen Vàng Đỏ

50. 30. 20 0

35 65 15 0

0 60. 15 25

0 55 10 35

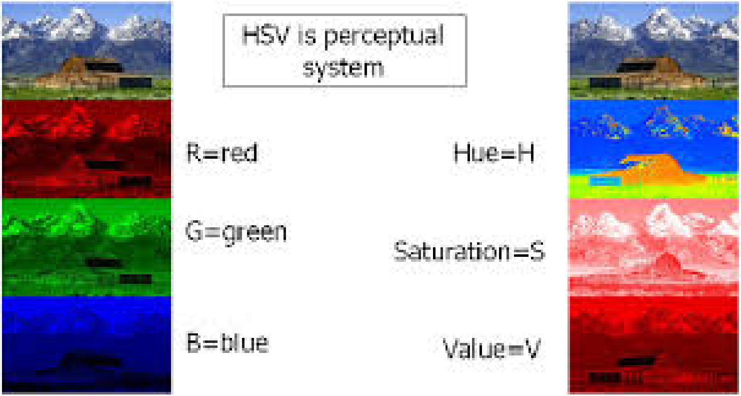


Colors based features

- An image maybe a gray (intensity image) or colored image
- Normally, using RGB (8bit = 256 values for R,G,B color) to encode color of image
- R,G,B color channels have high correlations
- $\Rightarrow$ using other color space
  - YCrCb
  - Lab (Light, Red/Green Value, Blue/Yellow Value)
  - HSV (Hue, Saturation, Values)

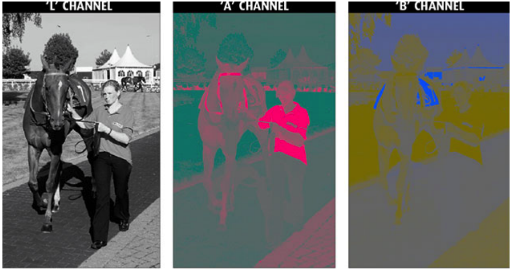
Colors based features

HSV color space



Colors based features

LAB color space



Colors based features

Compute color features?

- Step 1: $\Rightarrow$ Convert RGB to other color space (HSV, LAB...)
- Step 2: Calculate mean, standard deviation, Skewness for each color of images $\Rightarrow$ Color Moment (**CM**) features:
  - Mean: $E_c = \frac{1}{N} \sum_{allpixels} I_{i,j}$, where N is number pixels
  - Std: $\sigma_c = \sqrt{\frac{1}{N} \sum_{allpixels} (I_{i,j} - E)^2}$
  - Skewness: $s_c = \sqrt[3]{\frac{1}{N} \sum_{allpixels} (I_{i,j} - E)^3}$
- RGB image $\Rightarrow$ CM features as: $[E_R, E_G, E_B, \sigma_R, \sigma_G, \sigma_B, s_R, s_G, s_B]$

Colors based features

Calculate Normalized Color Histogram (**CH**):

- Compute histogram for each color channel (3 bins for R,G, 2 bin for B)
- Or compute histogram: 8 bins for R+G+B, 16 bin for R-G and 2B-R-G
- Concatenate histograms above
- Normalized
- Using histogram intersection distance to compare histogram

COLOR MODULE

COLOR SPACE

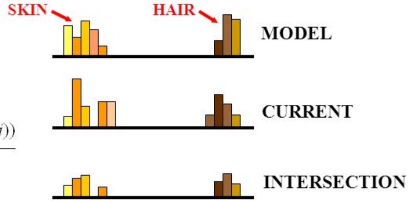
B+G+R (4 bins) 8 bin

B-G (8 bins) 16 G-R (8 bins) 16

HISTOGRAM INTERSECTION

[Swain & Ballard 1991]

$$\phi_c(\mathbf{s}) = \frac{\sum_{i=1}^{N} \min(I_s(i), M(i))}{\sum_{i=1}^{N} I_s(i)}$$



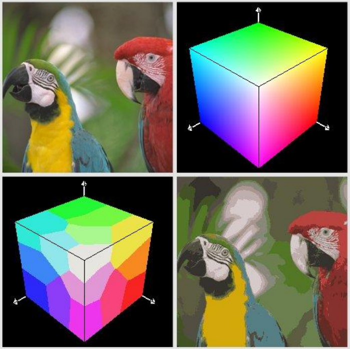
NORMALIZATION

$$\bar{\phi}_c(\mathbf{s}) = \frac{\phi_c(\mathbf{s}) - \min_{s_i \in S} \phi_c(\mathbf{s}_i)}{\max_{s_i \in S} \phi_c(\mathbf{s}_i) - \min_{s_i \in S} \phi_c(\mathbf{s}_i)}$$

Colors based features

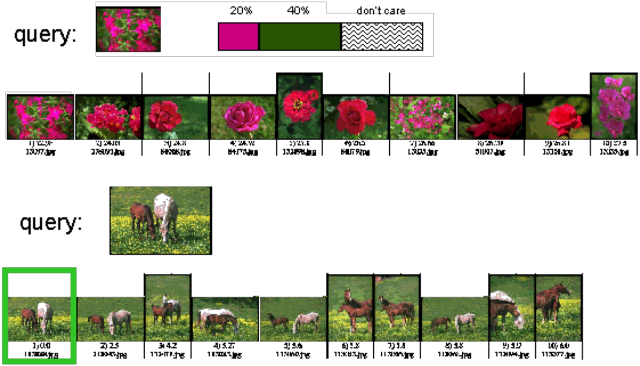
Compute color features?

- Too much colors: for example 256R * 256G * 256 B =16.777.216 colors
- $\Rightarrow$ Color Quantizations (Kmean, Meanshift, ColorName)



- $\Rightarrow$ Histogram on Quantized Image

Examples: Color-based Retrieval 2005

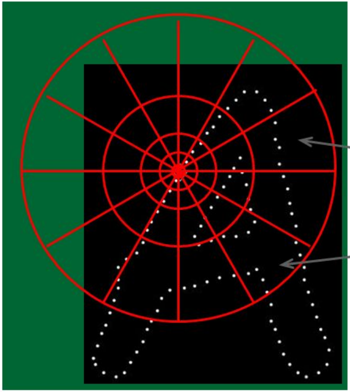


Shape based features

Image=> colors, shape, freq

w1 color +w2 shape + w3 freq

Shape context descriptor [Belongie et al '02]



Count the number of points inside each bin, e.g.:

Count = 4

⋮

Count = 10

Compact representation of distribution of points relative to each point

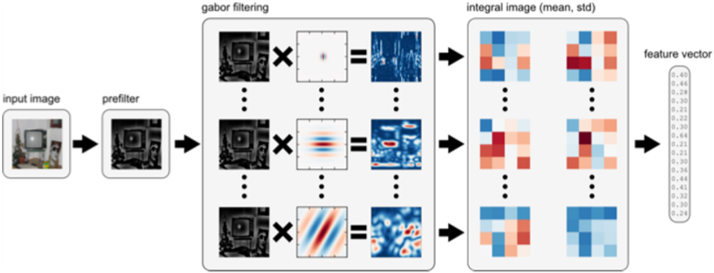
$$h_i(k) = \#\{q \neq p_i \ : \ (q - p_i) \in \text{bin}(k)\}.$$

Shape context slides from Belongie et al.

25

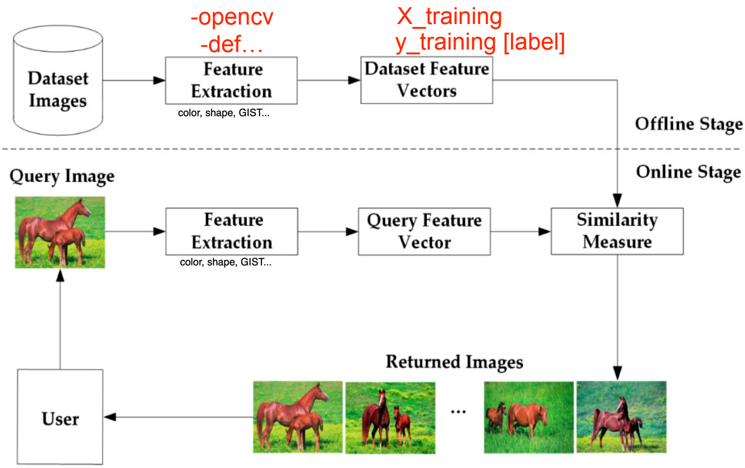
Global features-GIST

GIST:



Source - http://cybertron.cg.tu-berlin.de/pdci10/art2real/Art2Real/Image_Descriptor.html

Global features

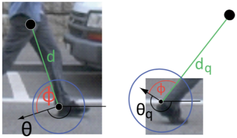


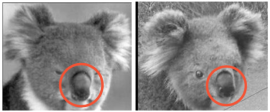
Limitations?

Different images content may have same color histogram

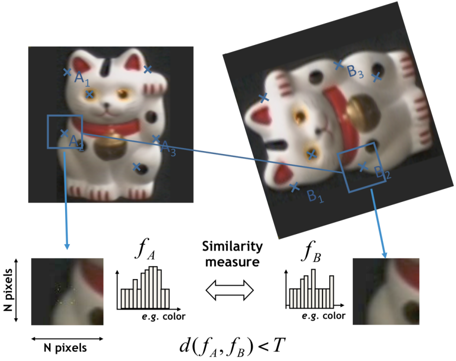
Motivation for using local features

- Global representations have major limitations
- Instead, describe and match only local regions
- Increased robustness to
  - Occlusions
  - Articulation
  - Intra-category variations





Motivation for using local features

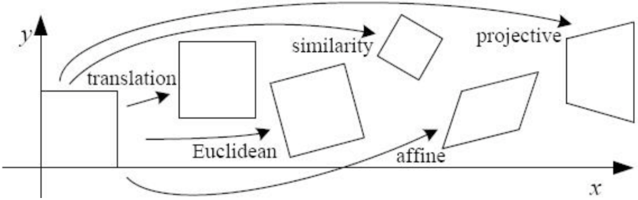


1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

Slide credit: Bastian Leibe

Local features

(Invariant) Local Features: Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



Invariance: Geometric Transformations

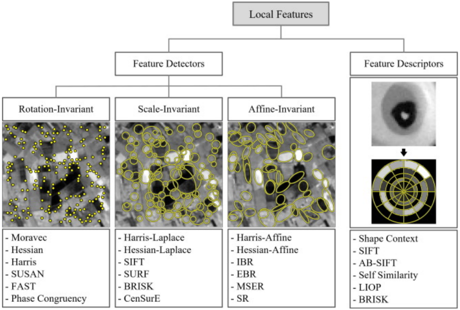
Slide credit: Steve Seitz

Local features

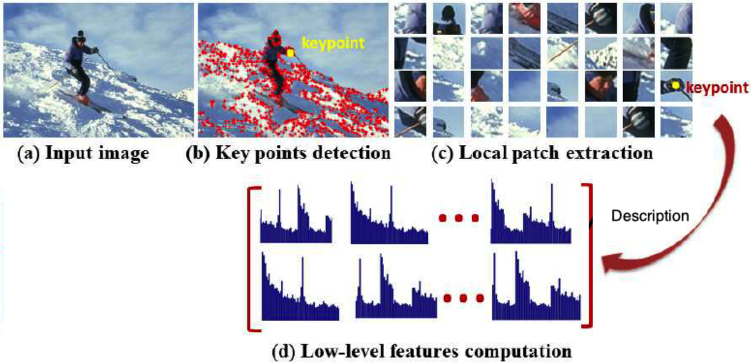
- Advantages of invariant local features:
  - Locality: features are local, so robust to occlusion and clutter
  - Distinctiveness: individual features can be matched to a large database of objects
  - Quantity: many features can be generated for even small objects
  - Efficiency: close to realtimeme performance
  - Extensibility: can easily be extended to wide range of differing feature types, with each adding robustness

Local features

- Feature Detector: Detect interesting point/location in images
  Dense sampling: A special detector, sample image by a fix sliding window with fix overlap
- Feature Descriptor: an encoder method, to convert an image block into a vector, with some properties.

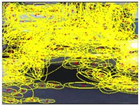


Local features



Intersting point detector or dense sampling?

- To detect intersting regions in image:



Sparse, at interest points

Dense, uniformly

Randomly

- Requirement:
  - Region extraction needs to be repeatable and accurate
    - Invariant to translation, rotation, scale changes
    - Robust or covariant to out-of-plane (≈affine) transformations
    - Robust to lighting variations, noise, blur, quantization
  - Locality: Features are local, therefore robust to occlusion and clutter.
  - Quantity: We need a sufficient number of regions to cover the object.
  - Distinctivenes : The regions should contain "interesting" structure.
  - Efficiency: Close to real-time performance.

e credit: Bastian Leibe

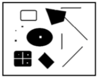
Many Existing Detectors Available

- Hessian & Harris [Beaudet '78], [Harris '88]
- Laplacian, DoG [Lindeberg '98], [Lowe '99]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid '01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid '04]
- EBR and IBR [Tuytelaars & Van Gool '04]
- MSER [Matas '02]
- Salient Regions [Kadir & Brady '01]
- Others...

- *Those detectors have become a basic building block for many recent applications in Computer Vision.*

Slide credit: Bastian Leibe

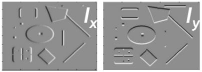
Summary: Harris Detector [Harris88]



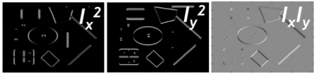
- Compute second moment matrix (autocorrelation matrix)

$$M(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

1. Image derivatives



2. Square of derivatives



3. Gaussian filter *g(σ$_I$)*

4. Cornerness function – two strong eigenvalues

$$\theta = \det[M(\sigma_I,\sigma_D)] - \alpha[\mathrm{trace}(M(\sigma_I,\sigma_D))]^2$$

$$= g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

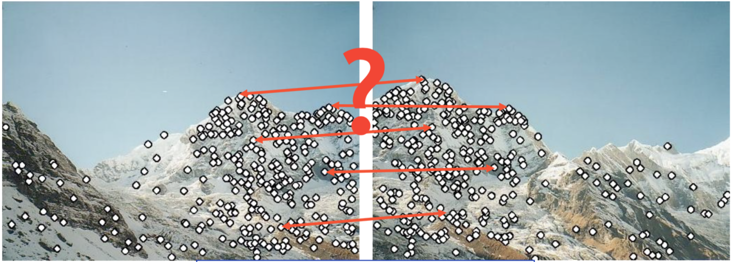
5. Perform non-maximum suppression

Slide credit: Krystian Mikolajczyk

Descriptor

- We know how to detect points
- Next question:

How to *describe* them for matching?

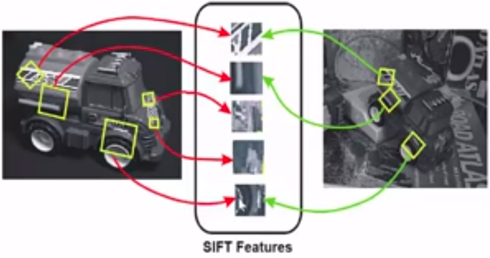


Point descriptor should be:

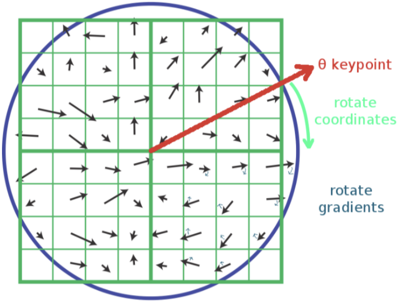
1. **Invariant**
2. **Distinctive**

SIFT-Description:

SIFT (Scale Invariance Features Transform): Bất biến với scale, rotation and translation
1 block ảnh A => scale, rotate, translate nó, tạo ra block B; thì SIFT(A) = SIFT(B).

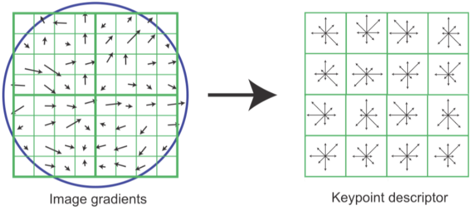


SIFT descriptor formation



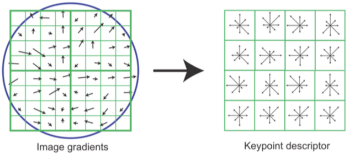
- Use the blurred image associated with the keypoint's scale
- Take image gradients over a 16x16 array of locations.
- To become rotation invariant, rotate the gradient directions AND locations by (-keypoint orientation)
  - Now we've cancelled out rotation and have gradients expressed at locations **relative** to keypoint orientation θ
  - We could also have just rotated the whole image by -θ, but that would be slower.

SIFT descriptor formation



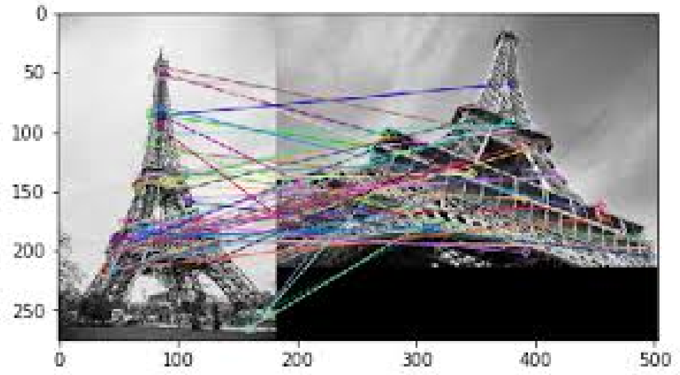
- Using precise gradient locations is fragile. We'd like to allow some "slop" in the image, and still produce a very similar descriptor
- Create array of orientation histograms (a 4x4 array is shown)
- Put the rotated gradients into their local orientation histograms
  - A gradients's contribution is divided among the nearby histograms based on distance. If it's halfway between two histogram locations, it gives a half contribution to both.
  - Also, scale down gradient contributions for gradients far from the center
- The SIFT authors found that best results were with 8 orientation bins per histogram, and a 4x4 histogram array.

SIFT descriptor formation



- 8 orientation bins per histogram, and a 4x4 histogram array, yields 8 x 4x4 = 128 numbers.
- So a SIFT descriptor is a length 128 vector, which is invariant to rotation (because we rotated the descriptor) and scale (because we worked with the scaled image from DoG)
- We can compare each vector from image A to each vector from image B to find matching keypoints!
  - Euclidean "distance" between descriptor vectors gives a good measure of keypoint similarity

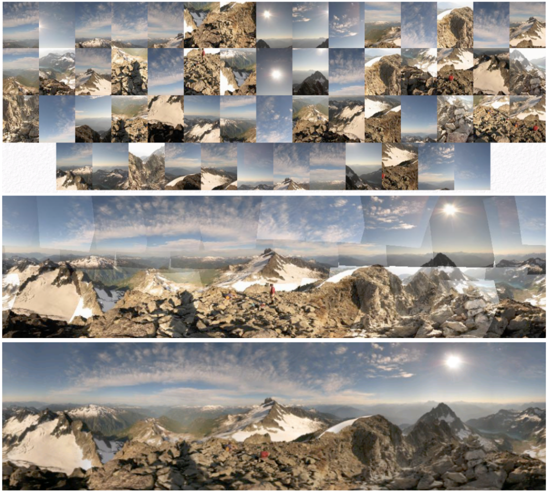
SIFT-Description: Image matching



SIFT-Description

Stitching to create panorama picture

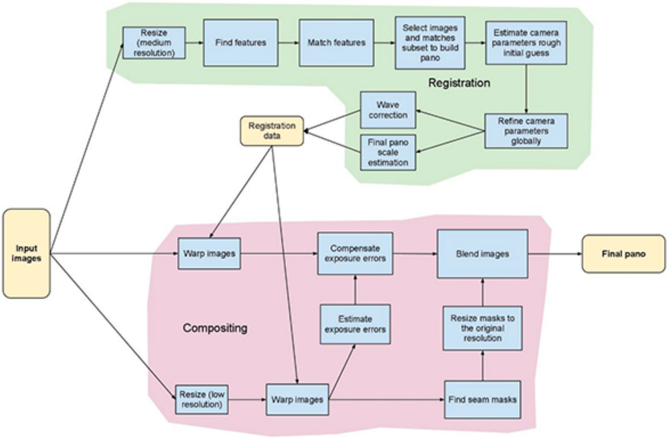
Automatic mosaicing



http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

LAB

Create your panorama: (Brown and Lowe: Automatic Panoramic Image Stitching with Invariant Features.)



OpenCV's image stitching algorithm