ASSINMENT-03

1. Background subtraction is a widely used approoch for detecting moving objects from static cameras. Many different methods were present but still researcher are confused to choose the best one among all. Generally, these procedures are normally utilized in traffic surveillance or video surveillance where the camera is static. The methods reviewed in the following are:

- **Running Gaussian average :**
  [1] have proposed to model the background independently at each (x,y) pixel location. The model is based on fitting a Gaussian pdf on last *n's* pixel values. In order to avoid fitting a Guassian pdf from scratch at each level at time t, we consider a average instead:

  $$\mu_t = \alpha I_t + (1-\alpha)\mu_{t-1}$$

  where I_t is the present pixel value and *u* be the previous average. There always exists a tradeoff between stability and frequent update. Similarly, we can compute standard deviation. In addition to speed, it also required less memory utilization. Since for each pixel it consists only of two parameters i.e (mean, sd) instead of buffer of last *n's* pixel values computed so far.
  At each frame time *t* , the pixel as classified as foreground and background if the euqality holds:

  $$|I_t - \mu_t| > k\sigma_t$$

  otherwise it will be considered as background. *'Background Substraction'* is the name that was commonly used to indicate this set of teachniques (2).
- Later, there's an update in an equation where binary value *M* is introduced that corresponds to 1 if foreground and 0 if background. Formula given below:

  $$\mu_t = M\mu_{t-1} + (1-M)(\alpha I_t + (1-\alpha)\mu_{t-1}) \quad (3),$$

  Since the model was proposed for the intensity of the images, extension can be considered or made for colour space images aldo i.e (R,G,B), (H,S,V) and so on. Moreover if we consider real time scenario and requirements constratins then update rate of (mean) can be set to less than the value of sample frame rate. Thus, the lower the update rate of background model, the less a system will be able to responsd to the actual dynamic background.

- **Mixture of Gaussians:**
  This implementation is particularly for indoor scenes and has been modified to be adapted for various other applications.
  A (x,y) pixel value is modelled using *n* Gaussian distributions. These distributions are compared with the new input value. If the value doesn't match the existing distribution value then a new component is added, otherwise the backgroud pair i.e (mean, sd) gets updated. Thus to become a model of background alone, a

crietia is set to distinguish between background and foreground distribution. First all the distributions are ranked based on the ratios their peak amplitude *w* and standard deviation

w : standard deviation

The assumption is that, higher and compact distribution are more likely to belong to the background. Then, the first B distributions in the ranking order stastifying

$$\sum_{i=1}^{B} \omega_i > T$$

- with T assigned threshold, considered as background.

- **Kernel Density Estimation (KDE):**
  As discussed above in this technique also we consider Gaussian pdf for the reconstruction of foreground and backgroud using kernel function. An approximation of background pdf can be a histogram given by most recent values which are classified as backgrund values. Since samples are less in number hence, this approximation has some drawbacks:

  i. histogram is a step function, which might provide poor modeling of true, unknow pdf.

  Thus, to overcome this problem a new model was proposed considering non-parametric model based on KDE on the buffer of last *n* pixel values. Which further gurantees the smooth and continous histogram.
  The background pdf is given as a sum of Gaussian kemels centered in the most recent n background values, *xi*:

$$P(x_t) = \frac{1}{n}\sum_{i=1}^{n}\eta(x_t - x_i, \Sigma_t)$$

  where *n* is the number of previous frames and (eta) is the kernel function, both of which are used to estimate the Parzen window P.

- **Eigenbackgrounds:**
  This approach is based on eigen values decomposition, but here it is applied to whole image instead of blocks image only. Such extension of spatial domain can be extensively explore spatial correlation and avoid tilling effects of block partioning.

  i. *Learning phase*
  In this approach, a sample of *n* images is taken into account where each image p pixels value are considered, then the average mean of p pixels value are computed and all images mean-subtracted
  Futher, a covariance matrix is computed and top M values are stored in eigenvector matrix. Hence, size of matri is (M *x* p).

  ii. *Classification phase*

Everytime a new image I appeared it is projected to eigenspace as $I' =$ eigenvectormatrix(I-mean).
I' is then back projected onto the image space as I''. As eigen values is a good model for static parts of the scene, but not for small movinf objects. Therefore, I'' will not have any such objects.
Foreground points are detected at locations where $|I – I''| > T$.

- **Frame differencing:**
  It is one of the easiset and simplest approach for background subtraction which is why it is widely used by a lot of researchers. In this technique, the previous frame is considered as the background model. Thus, absolute difference is computed between the present frame and the present background model (which is essentially the previous frame). Then, a threshold is set to find the foreground in the current frame.

$$\left| frame_i - frame_{i-1} \right| > threshold$$

- **Other commonly used strategies could be:**
  i. The initial frame is subtracted from a frame where the object is present, there will be a change in intensity value (non-zero) at pixels value where object is present and if only background is present then we will get 0 values.
  ii. The subraction of the current frame by a genric frame that is developed by using the mean, median or mode of all the frames. But problem encountered here is that it requires a lot of memory.

2. ***Motion History Image:*** The MHI is a static image where pixel intensity is a function of the recency of motion in a sequence.
MHI approach is a view-based temporal template method which is simple and robust in representing movements and is widely used by many researchers for action recognition, motion analysis and other related applications. Basically in simple terms, it is a template matching approach.
This algorithm was developed to represent the motion that occur in video recording with a single image. The algo produces a grayscale image, in which the white pixels shows most recent movements and darkest gray shows the earliest motion ekements. Black pixels shows absence of movement. This algorithm was widely used in the field of human action recognition.

Applications of MHI are as follows:

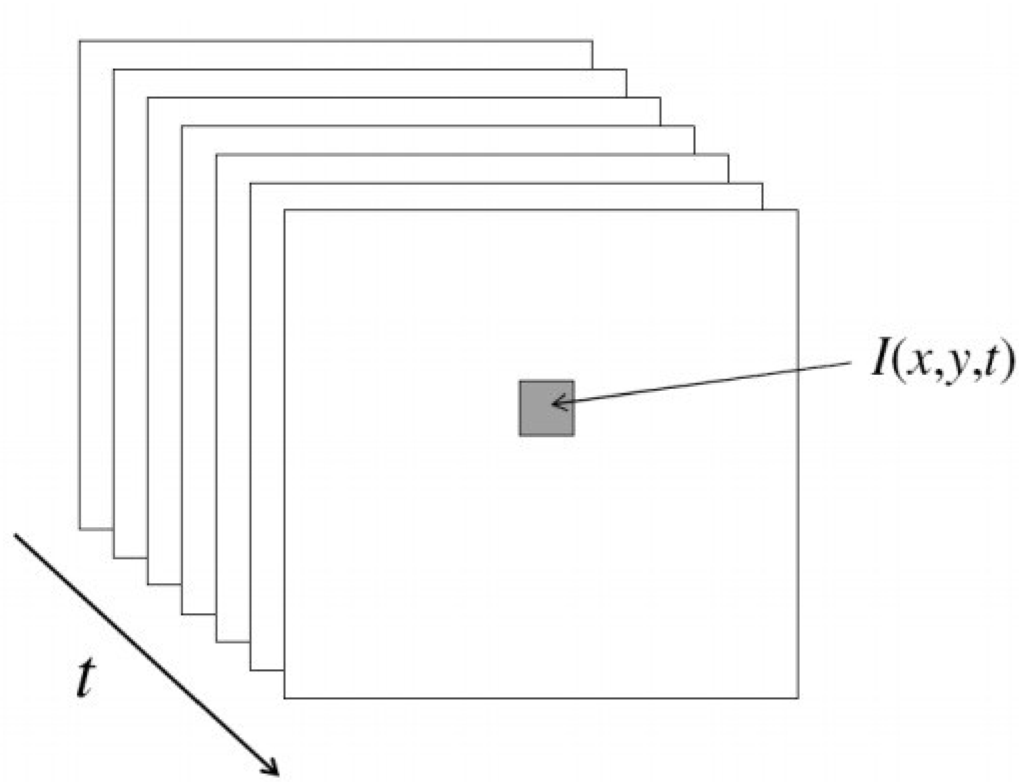
i. MHI template is not much susciptible to silhouette noises, like holes, shadows, and missing parts.Thus make the templates a suitable candidate for motion and gait analysis.
ii. It is widely used for motion analysis and gesture recognition.

For motion analysis: MHI is employed for various motion detection and localization, for automatic video surveillance and other purposes. It is also used for automatically localizing and tracking moving person or vehicle for an automatic visual surveillance system.

For gesture recognition: MHI is used to recognize different human movements and moving objects tracking.

iii. MHI has been modified to perform better analysis. Using Hierarchical MHIs, motion orientation histograms can be calculated that can capture subtle differences between similar motions caused by reflection. The motion orientation histogram representation quite distinctly captures the differences resulting from the view-based changes.
iv. Another application of MHI is a area of video surveillance is to analyse human behaviour and detect unusual behavior. Since, falls are one of the greatest risks for the elderly living at home. This approach is used to classify different variantion in human shapes which can establship the human pose hence the fall events also. Thus, this application is very uselful for old age people.
v. Others may be detecting depressions in human through video, which can further help in clinical analysis of a depressive disorder.
vi. The motion history image (MHI), used to visualize the decreased blood flow during an acute cerebral ischemic event in a mouse brain. This was implemented on the dynamic fluorescent (DF) data images and visualize the regions where perfusion evolves with time.
vii. Motion History for detecting facial action in video.

3. A video is a sequence of frames captured over time, Now our image is function f space(x,y) at time t.



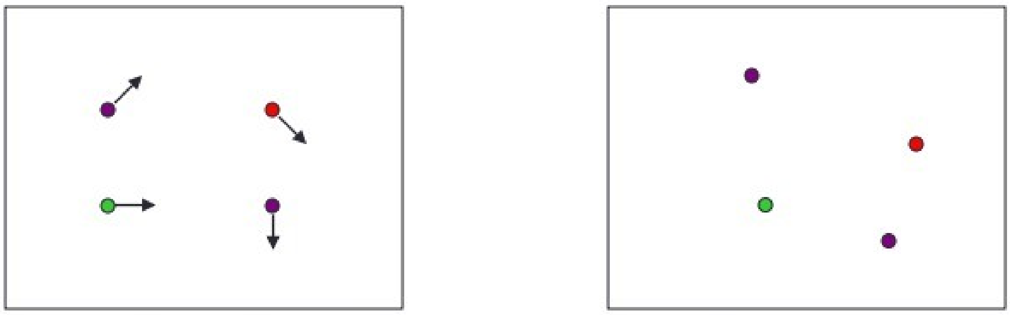
Optical Flow:

Optical flow is the apparent motion of brightness patterns in the image. Apparent motion can be caused by lighting changes without any actual motion.

For example, consider a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination.

Aim: Recover image motion at each pixel from optical flow.

Estimating optical flow



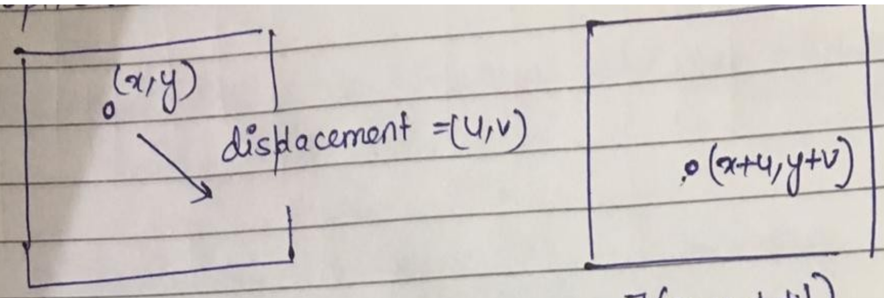
$I(x, y, t)$ $\qquad$ $I(x, y, t+1)$

Given two subsequent frames, estimate the apparent motion field *u(x,y), v(x,y)* between them.
Estimate pixel motion from image I(x,y,t) to I(x,y,t+1) by solving pixel correspondance problem, given a pixel in I(x,y,t), consider for nearby pixels of the exact same color in I(x,y,t+1).
In simple terms, Given two consecutive image frames, estimate the motion of each pixel.

Key Assumptions:

- *color constancy:* a point in I(x,y, consider exact same in I(x,y,t+1)
  – For grayscale images, this is *brightness constancy*
- small motion: points do not move very far.

Page 1:



$I(x,y,t)$ $\qquad\qquad$ $I(x,y,t+1)$

Brightness Constancy:

$$I(x,y,t) = I(x+u(x,y),\ y+(x,y)+t+1)$$

Linearizing the right side equ^n using Taylor expansion.

$$I(x+u, y+u, t+1) \approx I(x,y,t) + I_x \cdot u(x,y) + I_y \cdot v(x,y) + I_t$$

$I_x \rightarrow$ Image derivative along x.
$I_y \rightarrow$ Image derivative along y.
$I_t \rightarrow$ Difference over the frames

$$I(x+u, y+u, t+1) - I(x,y,t) = I_x \cdot u(x,y) + I_y \cdot v(x,y) + I_t$$

Hence, $I_x \cdot u + I_y \cdot v + I_t = 0$ (shorthand form)

$$\underset{(x\text{-flow})}{I_x \cdot u} + \underset{(y\text{-flow})}{I_y \cdot v} + I_t = 0$$

$$\underset{(1\times2)}{\nabla I} \cdot \underset{(2\times1)}{[u, v]^T} + I_t = 0 \quad \text{(vector form)}$$

$I_x, I_y \rightarrow$ Image gradients

$u, v \rightarrow$ flow velocities

$I_t \rightarrow$ temporal gradient

$$\left. I_x = \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y} \right\} \text{Spatial derivative}$$

$$\left. I_t = \frac{\partial I}{\partial t} \right\} \text{temporal derivative}$$

eg: frame difference.

3x3 patch

t

1	1	1	1	1
1	1	1	1	1
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10

−

$t+1$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	10	10	10
1	1	10	10	10
1	1	10	10	10

=

$I_t = \frac{\partial I}{\partial t}$

0	0	0	0	0
0	0	0	0	0
0	9	9	9	9
0	9	0	0	0
0	9	0	0	0
0	9	0	0	0

Page 3:

Now,

$I_x = \frac{\partial I}{\partial x}$ $\qquad$ $I_y = \frac{\partial I}{\partial y}$

-	0	0	0	-
-	0	0	0	-
-	9	0	0	-
-	9	0	0	-
-	9	0	0	-
-	9	0	0	-

$\rightarrow x$, $\downarrow y$

$-1 \quad 0 \quad 1$

-	-	-	-	-
0	0	0	0	0
0	9	9	9	9
-	0	0	0	0
0	0	0	0	0
-	-	-	-	-

$\rightarrow x$, $\downarrow y$

$\begin{matrix} -1 \\ 0 \\ 1 \end{matrix}$

$I_t = \frac{\partial I}{\partial t}$

0	0	0	0	0
0	0	0	0	0
0	9	9	9	9
0	9	0	0	0
0	9	0	0	0
0	9	0	0	0

$\rightarrow x$, $\downarrow y$

Now, $I_x u + I_y v + I_t = 0$

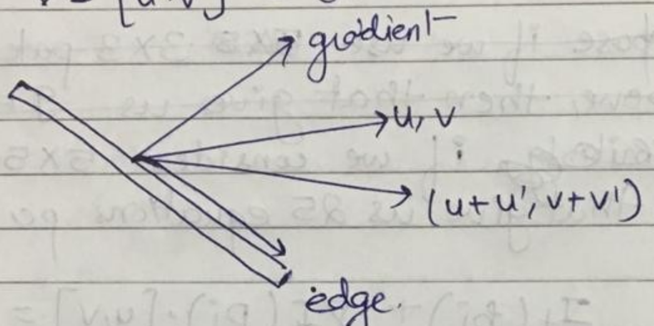
we've two unknowns and 1 equ^n (scalar equ^n)

Page 4:

The component of flow is $\perp r$ to the gradient which means that it is parallel to the edge) cannot be measured.

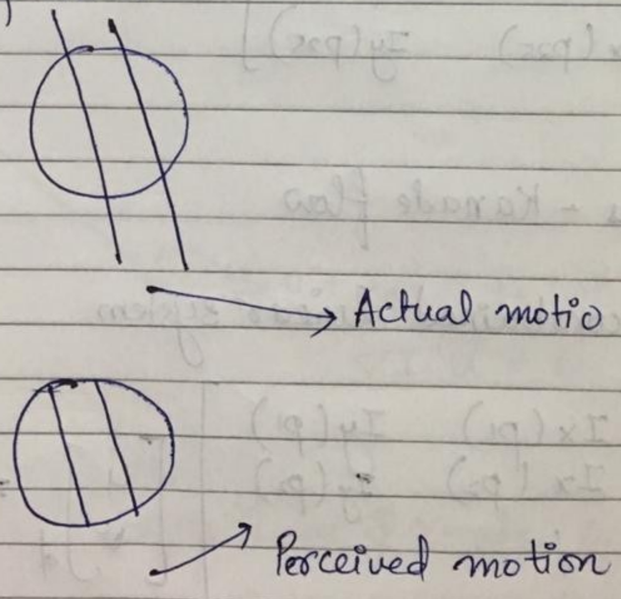
Thus, if (u,v) satisfies the equ^n, so does $(u+u', v+v')$

if $\nabla I \, [u' \cdot v']^T = 0$.



$$\nabla I \begin{bmatrix} u \\ v \end{bmatrix} + \nabla I \cdot \begin{bmatrix} u' \\ v' \end{bmatrix} \Longrightarrow \nabla I \, [u+u', v+v']^T + I_t = 0$$

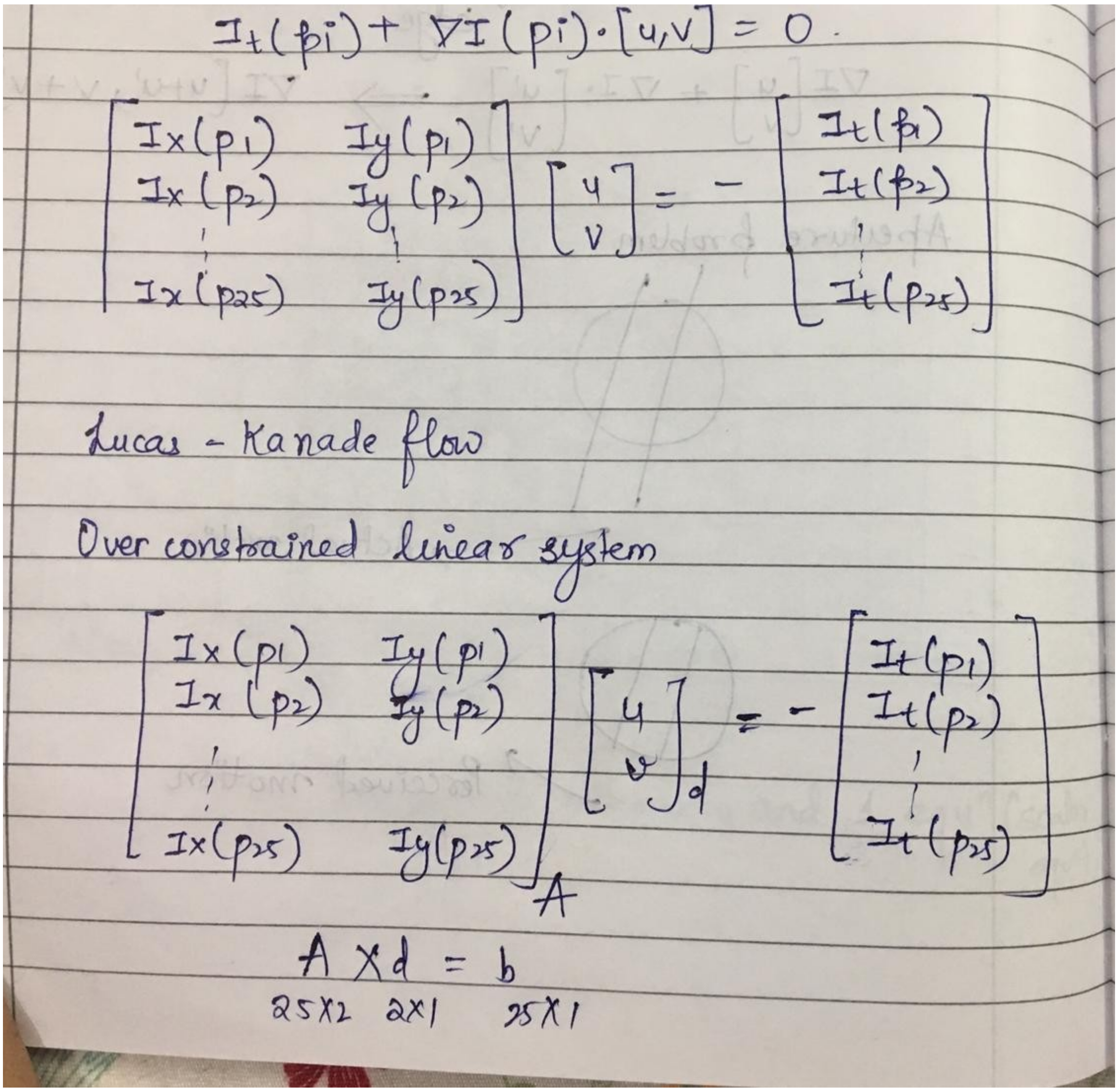
Aperture problem



Thus, we need to consider equations for a pixels.

Sapatial coherence constratint :
Assume a that pixel neighbours has same *(u,v)* . Suppose, we conside 5x5 window, then it will produce 25 equations per pixel.

Page 5:



$$I_t(p_i) + \nabla I(p_i) \cdot [u, v] = 0.$$

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix}$$

Lucas - Kanade flow

Over constrained linear system

$$\underbrace{\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{d} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix}$$

$$\underset{25\times 2}{A} \times \underset{2\times 1}{d} = \underset{25\times 1}{b}$$

Problem : we have more equations than unknowns
Thus, we need to minimize the equation shown below:

minimize $\|Ad - b\|^2$

Solve least square problem:

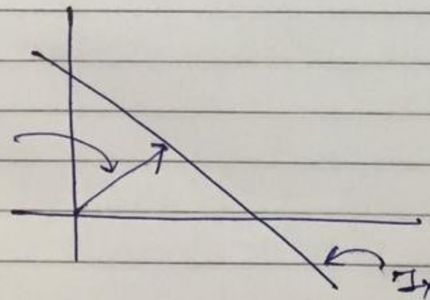
minimum least square solution given by solution (in d) of

$$(A^T A)\, d = A^T b$$
$$2\times2 \quad 2\times1 \quad 2\times1$$

$$\underbrace{\begin{bmatrix} \Sigma I_x I_x & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \Sigma I_x I_t \\ \Sigma I_y I_t \end{bmatrix}}_{A^T b}$$

The summation is over all pixels in $K \times K$ window.
This technique is first imposed by Lukas & Kanade

Normal flow



$$I_x u + I_y v + I_t = 0.$$
$$\nabla I \cdot \vec{u} = 0$$

Conditions for Solvability:
Optimal (u,v) satisfies Lucas & Kanade eqn.

$$\underbrace{\begin{bmatrix} \Sigma I_x I_y & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \Sigma I_x I_y \\ \Sigma I_y I_t \end{bmatrix}}_{A^T b}$$

Conditions for Solvability:

- $\mathbf{A^TA}$ should be invertible.
- $\mathbf{A^TA}$ should not be too small due to noise
  – eigenvalues λ_1 and λ_2 of $\mathbf{A^TA}$ should not be too small.
- $\mathbf{A^TA}$ should be well-conditioned
  – λ_1/λ_2 should not be too large (λ_1 = larger eigenvalue)
- M = $\mathbf{A^TA}$ is the second moment matrix (Harris Corner Detection)

  Eigenvectors and eigenvalues of $\mathbf{A^TA}$ relate to magnitude and edge direction.
- The eigenvectors with larger eigen values points in the direction of fastest intensity change.
- Other eigenvectors are orthogonal to each other.

Errors in Lucas- Kanade

- motion segmentation is to be done since point does not move like it's neighbors
- in this equation there is no brightness constancy
- The motion of point is non linear and larger than pixel

4. **Image co-segmentation:**
   It is defined as jointly segmenting semantically similar objects in multiple images. That means extracting common objects from the multiple images. In other words, it is the problem of automatically discovering the common objects co-occurring in a set of relevant images and segmenting them as foreground simultaneously.
   Different approaches used for implementation purpose are defined below:
   - Suppose we are provided with a set of N images and our goal is to extract the object that is present in all given images. Thus, for every image *n = 1,2,...N,* best 50 proposals segementation are taken in account out of total 200 proposals computed.
   - These proposals can be computed using any off-the-shelf proposal segmentation technique.
   - Therefore, the task of co-segmentation is then converted into a labelling problem in a complete graph, where each proposal segementation corresponds to label and each image corresponds to node in a graph.
   - Finally, goal is find out a labelling which is defined as below:
     $$\textbf{label} = (label_n \mid n=1, 2,...N;\ \ label_n \in \{1,\ldots,50\})$$
   - Then maximize the score function using labelling:

   $$E(label) = \sum_{n,m \le N, n \ne m} P(label_n, label_m)$$

   Co-saliency can also help in co-segmentation discussed below:
   - co-saliency model is used to generate image regions that are similar to each other across images, and also try to retain their distinctness within each image.
   - From the set of N images, the foreground of image I^i is the area containing the common object, where i=1,2,..N.
   - Then over-segmentation technique was applied to partition the I^i into superpixels i.e m_i.
   - After that, for representing foreground and background of image I^i, a binary vector of labels $x^i \in \{0,1\}^{m_i}$ is used.

- At last, co segmentation of these images can be modelled using Markov Random Function by minimizing the following enery function while computig the above binary labels:

$$F(\{\mathbf{x}^i\}) = \sum_i L_i(\mathbf{x}^i) + \lambda \cdot E(\{\mathbf{x}^i\})$$
$$= \sum_i L_i(\mathbf{x}^i) + \lambda \sum_{i,j} G(\mathbf{x}^i, \mathbf{x}^j, I^i, I^j) \ (1)$$

  Where $L_i(x^i)$ represents the MRF energy of the labelling (x^i) on (I^i) in a single image, G function represents the energy measuring the inconsistency between I^i and I^j under the labelings x^i and x^j, and α weighs the importance of the global energy term E({x i}). The co-saliency prior is then used to determine the energy function L here.
- Saliency detection is often defined as to find out distinct areas in an image, as human eyes are easily attracted by the unusual things with respect to the whole view.
- Our co-segmentation model assumes that in most of the images $\{I_i\}^M_{i=1}$, their detected salient areas contain at least parts of the foreground object.
- The technique of saliency involves finding out what 'stands-out' in the image. This usually involves some object in the foreground. This is termed as 'distinctness' property of the image.
- The 'repeatedness' property among images is also crucial for co-segmentation it occur when same saliency object was detected in most of the images.

$$Co-saliency = Saliency * Repeatedness$$

- Based on the above equation, we try to find out co-saliency maps and give weights to inter- map energy. Then energy minimization technique was followed as defined above $F(\{x^i\})$.

To further elaborate this idea and to supress the noise images, a cluster-based co-saliency detection algorithm was defined below:
- Intially method consider two-layer cluster on a set of images i.e first layer consists of images that are grouped together on the basis of the pixels within an each image(single image) and other layer cluster the pixels on multiple images(all images).
- Then we compute the saliency cues for each cluster, and find the cluster – level saliency. The measured features include the uniqueness(on single/multi-image), the distance from the image center(on single/multiple images) and repetitiveness(on multi-image).
- They are know as contrast, spatial, and corresponding cues, respectively.
- Lastly, based on these cues cluster-wise, saliency map is computed by computing the saliency value for each pixel.

This method has the advantage of less memory requirement and computation cost as individual pixel operators are compared cluster-wise.

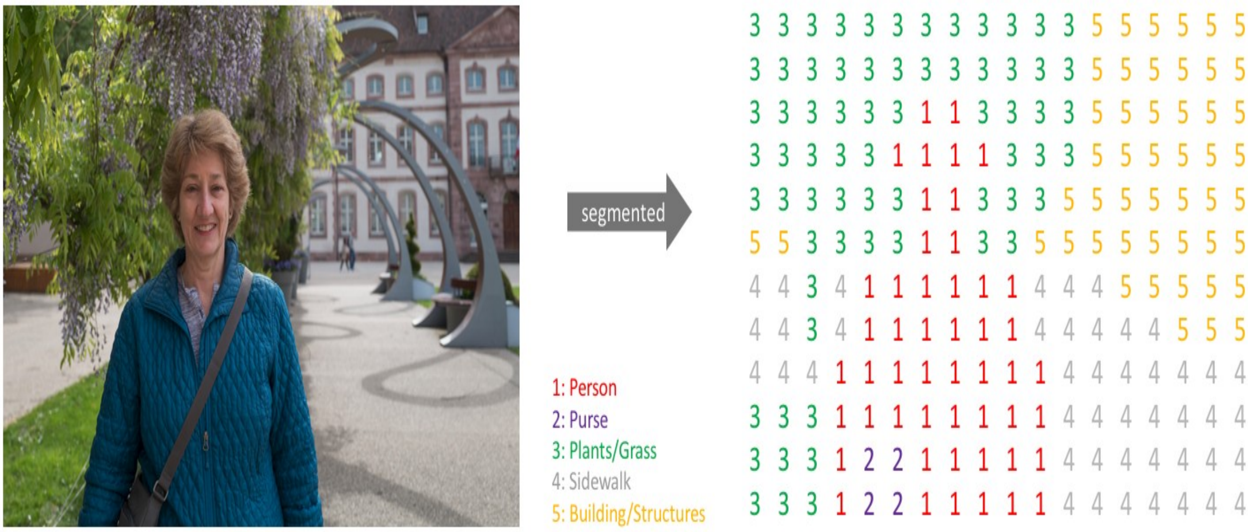
After getting saliency maps, co-segmentation is performed in a way as defined above.

Finally, we discuss the a quality guided fusion based co saliency estimation technique for co-segmentation.

- Fusion of saliency maps of different images is performed using dense correspodence technique, the co-saliency estimation is guided by quality measurement which helps decide whether the saliency fusion really improves the quality of the saliency map or not.
- Moreover, the joint process is guided by quality measurement system, which helps us decide whether to choose the original or fused saliency map as the final one.
- Thus, Quality metric is high quality saliency map that should have well seperated fore and background components.
- Then by using benchmarked saliency maps, co-segmentation is performed using Markov Random Function for energy minimization.

5. Sematic Segmentation:

It is defined as label each pixel of an image, with a corresponding class of what is being presented.

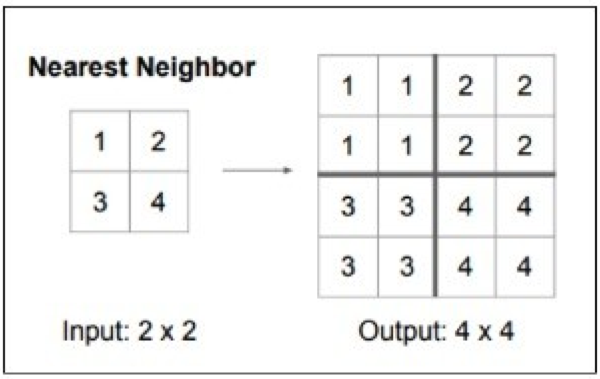


Input Semantic Labels

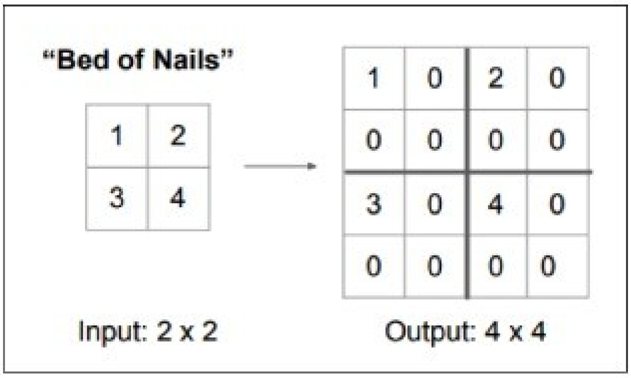
- Here we consider a supervised learning algorithms as ground truth is important.
- Usampling Algorithm by Upooling:

i. *Nearest Neighbour:*

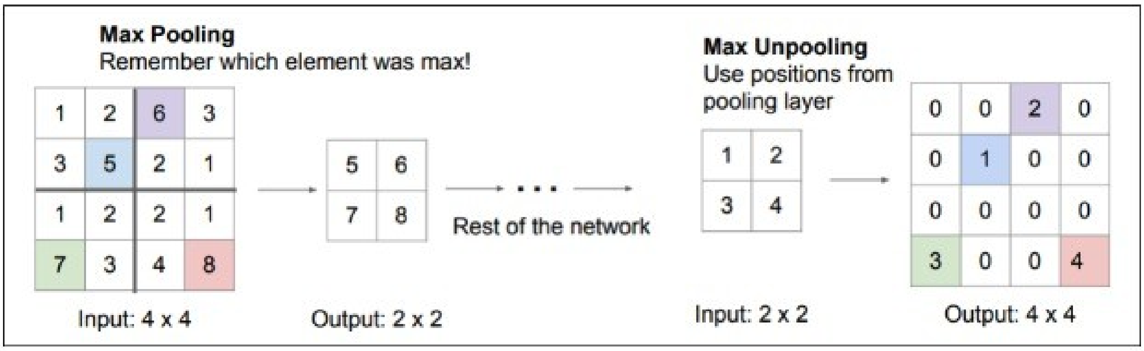
In this technique we simply resize the image or a volume by Nearest Neighbours we repeat values so that volume is resized to size of the original input image.



ii. *Bed of Nails:*
In this technique, we give the first pixel the original value and the rest of the pixels are given 0.



iii. *Max unpooling*
Here, we remember where exactly the maximum value was during Max-pooling and there we will put values of our output.



State of the art: Transpose convolutions, which is also know as Deconvolutions. Deconvolutions allow for us to develop a learned upsampling.

$$f * g = h$$

Where g is the convolution mask and h is the recorded image. Upsampling with a factor of x corresponds to convolution with a fractional stride of 1/x.

In a CNN, between each volume there are several convolutional filters whose weights we want to learn.
Let the squares on output volume be a convolution mask. Using this mask, we want to upsample the given 2x2 input volume.

For example,
Stride 3, pad 1

Output

Input

2	3
-2	1

Conv Mask

2	1	4
-1	-2	-3
-2	1	5

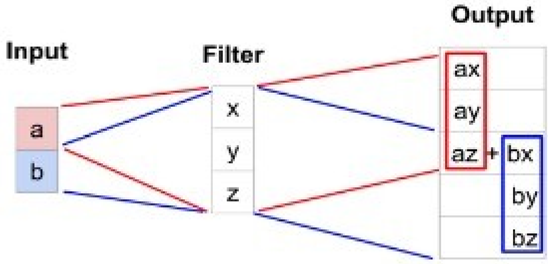
-4	-6	-3	-6
2	10	-6	3
-2	-8	2	1
4	6	-1	-2

Steps:

- Add padding to the 4x4 output volume.
- Place conv mask on the output and multiply pixel wise to fill the values in it.
- Essentially, the input is used as weights of the values in convoution mask to upsample to a larger volume.

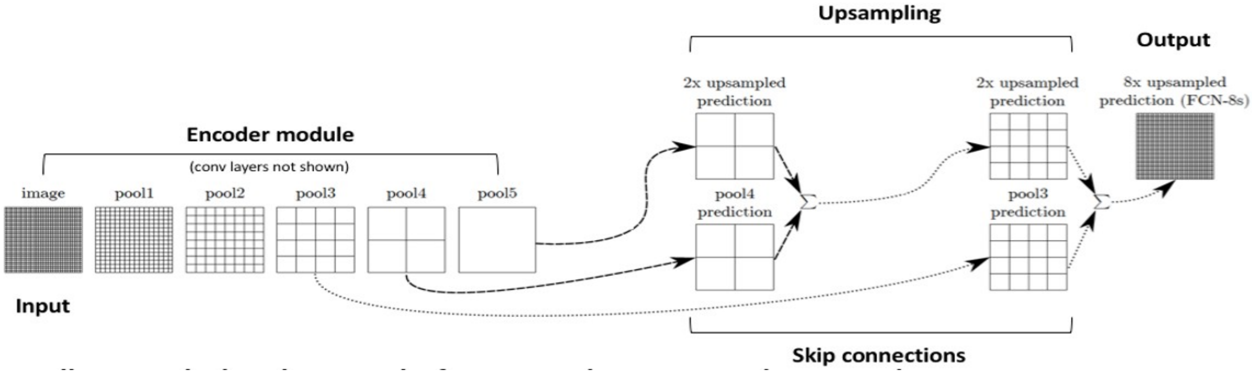
Here, there was no overlap between window as it was existed when stride was 3. If there exists an overlapping then we will keep adding the values we get for a pixel using each window. This is shown in the image below:

Stride 2, pad 0



Thus, output may vary according to stride and padding.

Skip connections: Skip connections can be implemented by using previous convolutional layers to fine tune the output of upsampling.
For example, f
for the image max-pooling might be performed several times. The previous layer must be having more spatial information than the current layer while max pooling. Thus instead of performing upsampling only the current layer, we try to upsample previously pooled layer also.
For the current layer *k*, the value is upsampled to the dimension of previous layer (k-*1)th* and the resultant image is added to the previous layer (k-*1)th* as now they must be having the same dimensions. This can be further explained using the image below:



Refrences:

1. **https://profs.sci.univr.it/~cristanm/teaching/sar_files/lezione4/Piccardi.pdf**
2. **https://www.ijser.org/researchpaper/A-Survey-Background-Subtraction-Techniques.pdf**
3. **http://web.cse.ohio-state.edu/~davis.1719/CVL/Research/MHI/mhi.html**
4. **https://www.researchgate.net/publication/225968468_Motion_history_image_Its_variants_and_applications**
5. **https://ieeexplore.ieee.org/document/938864**
6. **https://sci-hub.tw/https://link.springer.com/article/10.1186/s13640-017-0212-3**
7. **https://dl.acm.org/doi/10.1145/2910674.2935832**
8. **https://www.worldscientific.com/doi/abs/10.1142/S021800141655017X**
9. **https://content.iospress.com/articles/biomedical-spectroscopy-and-imaging/bsi170**
10. **http://vision.stanford.edu/teaching/cs231a_autumn1112/lecture/lecture13_optical_flow_cs231a_marked.pdf**
11. **https://homes.cs.washington.edu/~shapiro/EE596/notes/Optical_Flow.pdf**
12. **https://www.cc.gatech.edu/~afb/classes/CS4495-Fall2014/slides/CS4495-OpticFlow.pdf**
13. **http://www.cs.cmu.edu/~16385/s17/Slides/14.1_Brightness_Constancy.pdf**
14. https://ieeexplore.ieee.org/abstract/document/5539868
15. https://ieeexplore.ieee.org/document/7298965
16. **https://towardsdatascience.com/review-dilated-convolution-semantic-segmentation-9d5a5bd768f5**