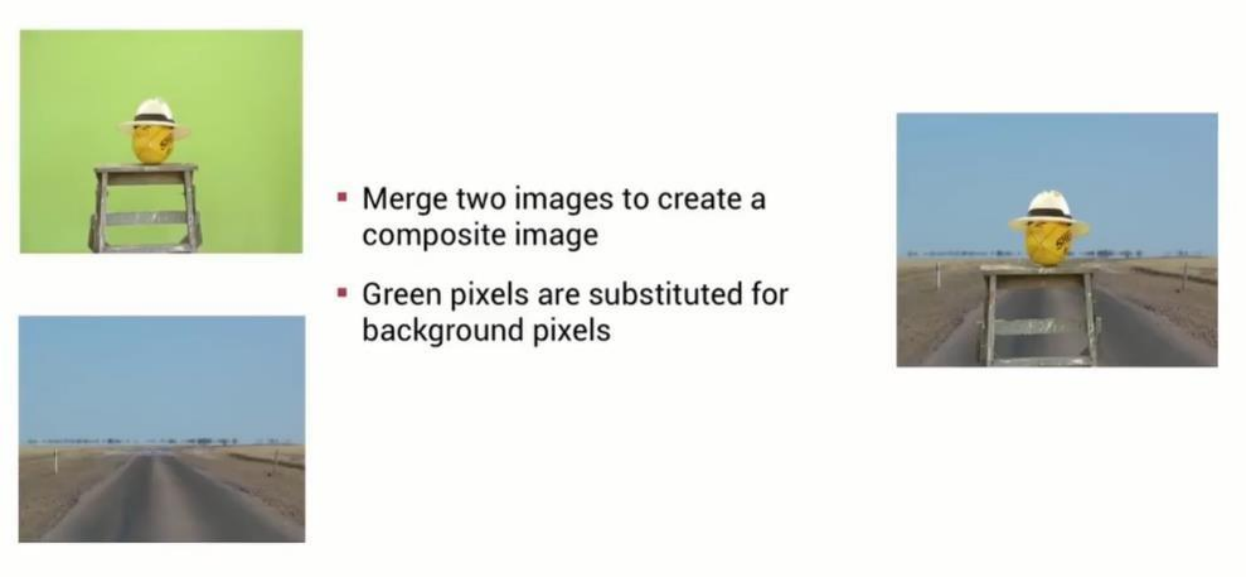
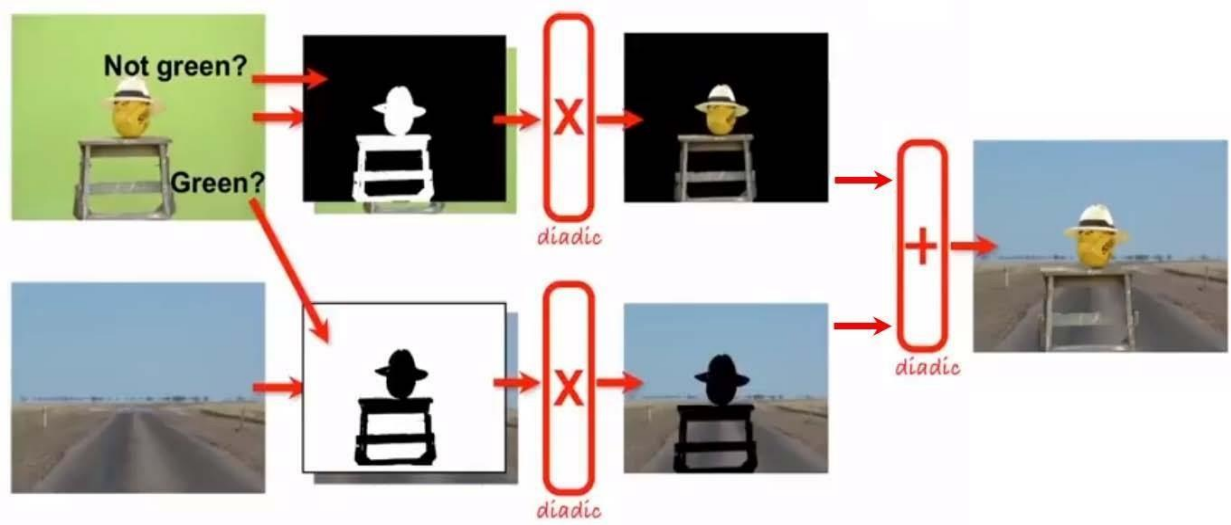
Digital Image Processing
Lab 6

Green Screen Effect, Monadic and Diadic Image Processing

1. Green Screen Effect



Algorithm:



2. Monadic Image Processing

i. Changing Data Type

```
img=imread('E:\mac\image.JPEG');I=im2double(img);
disp(I);
```

It will map image pixel values from type uint8 to double.

ii. Brightness

Offsetting pixel values.

$$F(x) = x + a$$

iii. Contrast

$$F(x) = 2x$$

iv. Negative/ Inverse Transformation

F(x) = 1 – x (In case of pixel values between 0-1)

Inverse transformation is also known as Image Complement.
Mathematically, it can be defined as:

$$s = F(x) = L - x$$

where, x is the current intensity at a pixel point (x, y) and s is the resultant intensity after the transformation function is applied onto r. L is the max. gray level value, usually $L = 2^8 - 1 = 256 - 1 = 255$.
MATLAB Function to implement inverse transformation:

Code:

```
img_neg = imcomplement(img);
```

It will compute the complement of the image img which can be a binary, grayscale, or RGB image. img_neg has the same class and size as img.

v. Gamma Correction / Power Law Transformation:

Power transformation is also known as Gamma Transformation.

Mathematically, it can be described as follows:

$$s = F(x) = x^{\gamma}$$

where, x is the current intensity at a pixel coordinate (x, y) and s is the processed intensity after the transformation is applied. ɣ (gamma) specifies how the intensities are mapped i.e. towards brighter side or darker side.

Gamma Values:

If ɣ < 1 :

then mapping is weighted more towards brighter side.

If ɣ > 1 :

more weight would be given to the darker side and so the image will be darker.

If ɣ = 1 :

then it would act as a linear function.

Code:

```
img2 = im2double(img) . ^ 2;
```

The Matlab function imadjust() can also be used to implement power law transformations:

```
img3 = imadjust(img, [ ], [ ], 2);
```

vi. Logarithmic Transformation:

Logarithmic transformations are implemented as follows:

$$s = F(x) = c \log(1 + x)$$

where, x is the current intensity at a pixel point (x, y) and s is the resultant intensity after the transformation function c * log(1 + x) is applied onto x.

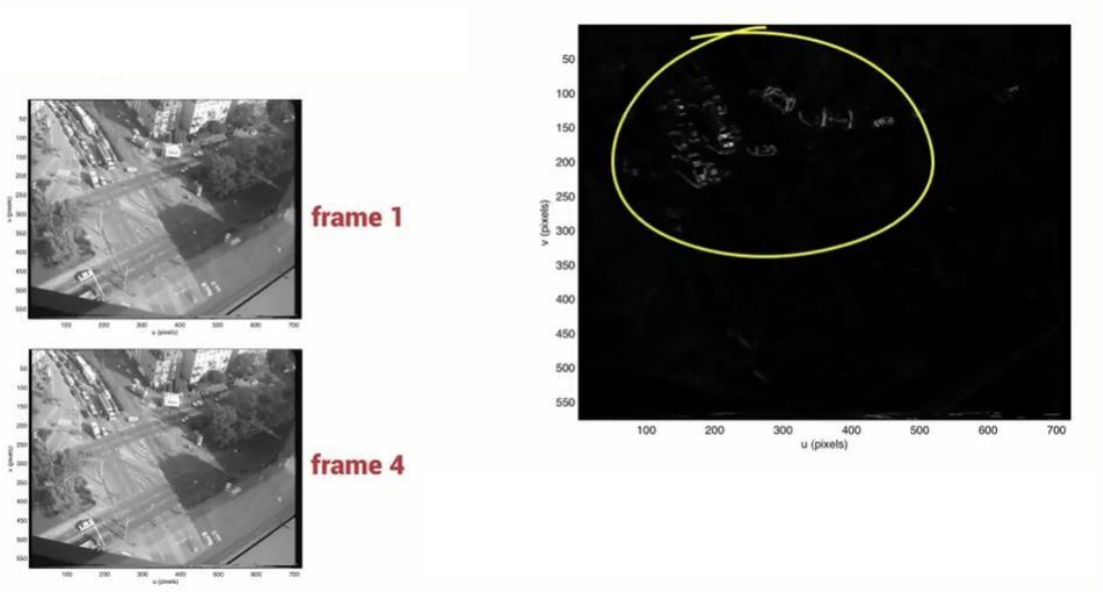
Example:

```
img2 = c * log(1 + im2double(img));
```

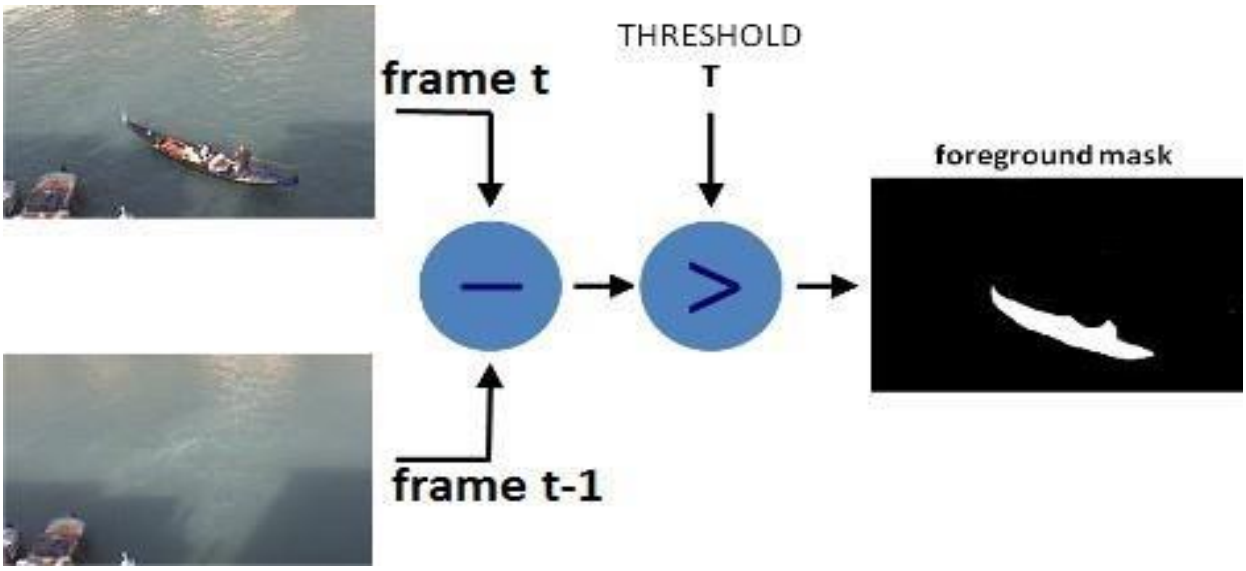
Log transformation of an image means replacing all pixel values, present in the image, with its logarithmic values. Log transformation is used for image enhancement as it expands dark pixels of the image as compared to higher pixel values.

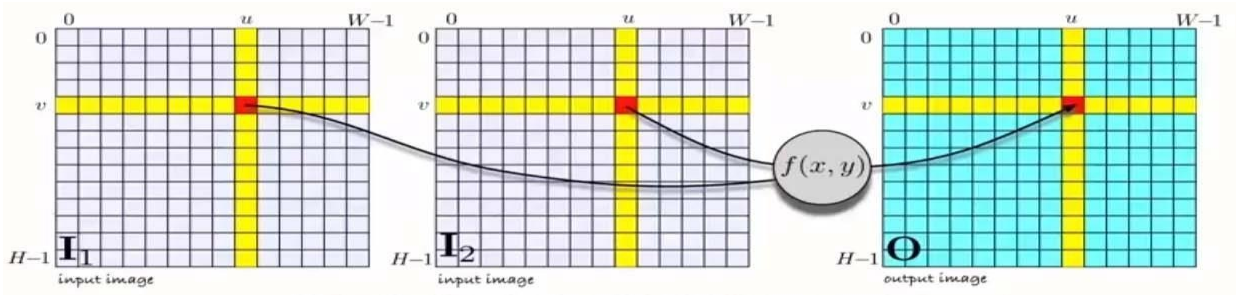
2. Diadic Image Processing

i. Detection of movement in a scene through subsequent frames subtraction:



ii. Detection of object in video frames through Frame Differencing





iii. Addition

iv. Subtraction

v. Multiplication

vi. Division