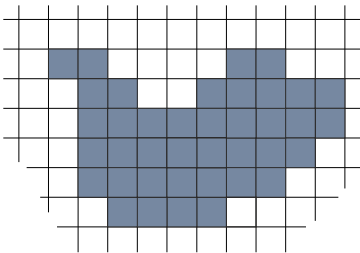


# Video Signals

Date: 7 February 2019

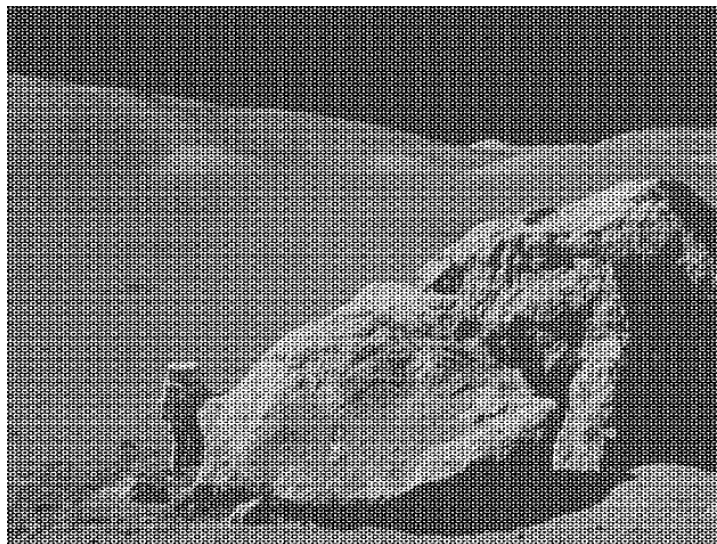
**Ex.1.[11 pt]** An industrial visual inspection system analyzes the conveyor belt of a production line and acquired the following image. The system has to check the screws: all of them are of the same size but with different rotations. Propose an algorithm that is able to localize screws independently from their rotation, disregards bolts and for every screw checks if their shape is consistent with a reference screw.



**Es.2. [11 pt]** Consider the B/W object on the left, describe the morphological procedure to extract its skeleton. Describe the whole procedure detailing the chosen structuring element and the result at each iteration. Furthermore, define a stopping condition for the algorithm.

## Es.3. [11 pt MATLAB Exercise]

You are working for NASA and you are receiving pictures from the moon. Unfortunately, due to errors during acquisition and/or transmission one of the image (stored in a file called 'moon\_landing.png') is heavily corrupted by periodic noise. You decide to perform denoising in the frequency domain. Write a Matlab script able to perform the following operations:



### Matlab List of possible functions

figure  
im2doubl  
e  
imread  
rgb2gray  
imcrop  
ifftshif  
t  
imfilter  
fft2  
ifft2  
fftshift  
max  
imopen  
imshow  
find  
real  
.

- Read the input grayscale image, convert it from a 8-bit unsigned integer representation to a double representation with values between 0 and 1.
- Apply the 2d discrete Fourier transform to the image and shift the result so the DC component would be in the central point of the result matrix.
- Remove the high frequency spikes introduced by the noise.
  - Copy the result obtained in point b) in a matrix called *DFT\_mod*.
  - Consider a 80x80 region centered around the central point of *DFT\_mod* and set the values inside to zero (assume that the original image has even length in both directions).
  - Calculate the power spectrum of *DFT\_mod* and calculate its maximum value *M*.
  - Considering a threshold equal to  $0.00001 * M$ , find the location in which the power spectrum exceed the threshold and set the corresponding values of *DFT\_mod* to 0.
  - In order to restore the low frequency component copy back the 80x80 central region from *DFT* to *DFT\_mod*.
- Obtain the restore image from *DFT\_mod* with the inverse Fourier transformation and show the result.

# Solutions

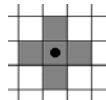
## Ex.1

A possible solution to distinguish between screws and bolts is to transform the image in a black and white image applying a proper threshold based on the background colour; then we can find the area in terms of pixels of all the objects applying multiple times the morphological filling algorithms on pixels not belonging to the background.

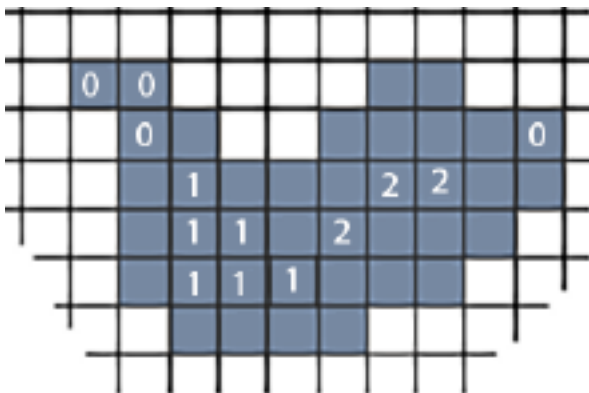
In order to get a robust descriptor invariant to rotations for every screw we can find the barycentre of every screw and then applying the transformation to polar coordinates for every screw with respect to its barycentre. Then, adopting the 2D Fourier Transform and analysing its magnitude we get a good descriptor invariant to translations: this descriptor can then be adopted to compare directly every screw with the reference one. Otherwise, from the polar coordinates we can estimate the rotation with respect to the reference screw simply searching a maximum in vertical circular correlation: a rotation in polar coordinates becomes a vertical circular shift).

## Ex.2

Using a 3x3 cross structuring element,



applying the skeletonization morphological operator, we get the following skeleton hierarchy:



## Ex.3

```
clc
close all
clear all

%a)
img = imread('moon_landing.png');
img = im2double(img);
figure; imshow(img)

%b)
DFT = fft2(img);
DFT = fftshift(DFT);

%c)
%I)
DFT_mod = DFT;
%II)
n = 40;
DFT_mod((end/2+1-n):(end/2+1+n),(end/2+1-n):(end/2+1+n)) = 0;
%III)
DFT_power_spectrum = real(DFT_mod).^2 + imag(DFT_mod).^2;
M = max(DFT_power_spectrum(:));
%IV)
```

```
thr = 0.00001*M;  
DFT_mod(DFT_power_spectrum>thr) = 0;  
%V)  
DFT_mod((end/2+1-n):(end/2+1+n),(end/2+1-n):(end/2+1+n)) = ...  
    DFT((end/2+1-n):(end/2+1+n),(end/2+1-n):(end/2+1+n));  
  
%d)  
DFT_mod = ifftshift(DFT_mod);  
img_filtered = (ifft2(DFT_mod));  
figure; imshow(img_filtered)
```