united nations educational, scientific and cultural organization

international atomic energy agency

the
abdus salam
international centre for theoretical physics

SMR: 1170/5

WORKSHOP ON "MODELLING REAL SYSTEMS: A HANDS-ON FIRST ENCOUNTER WITH INDUSTRIAL MATHEMATICS

(27 September - 22 October 1999)

"On a Computer Vision Project"

presented by:

Gunnar SPARR
Centre for Mathematical Sciences
Lund University/ LTH
P.O. Box 1188
S22100 Lund
Sweden

On a Computer Vision project

Gunnar Sparr

Centre for Mathematical Sciences
Lund university/LTH

Plan

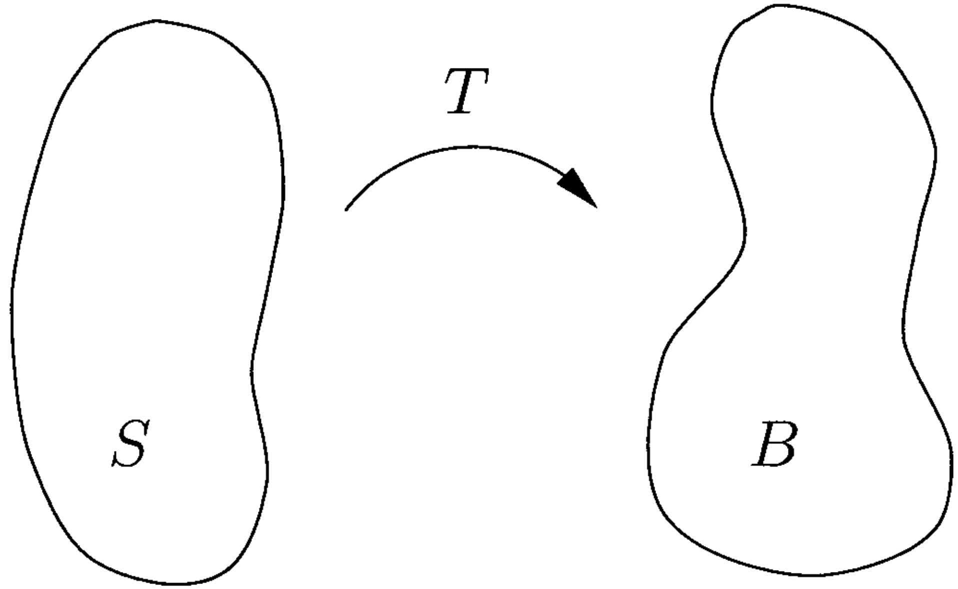
1. What is Computer Vision and what is the Project about?
2. Projective geometry
3. Image processing
4. More about the project

1. What is Computer Vision?

Short answer: A problem driven science, where visual information is used for complex tasks, like recognition, navigation, surveillance, medical diagnosis support, ...

Computer vision often tries to mimic human vision. Can only manage very simple tasks under very special circumstances, presently.

Computer Vision vs Computer Graphics



Computer Vision addresses the problem:

Given an *image*, B, and a model of an *image acquisition process* T:
Make relevant conclusions about the scene S.

Computer Graphics addresses the problem:

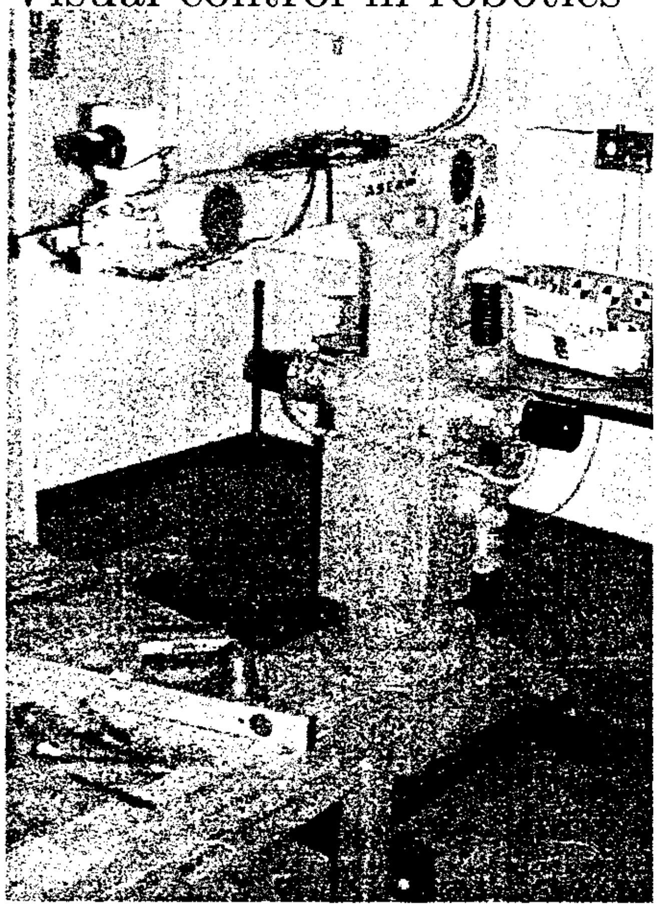
Given a *database*, S, and an *algorithm* T:
Generate a relevant *image* B.

Image acquisition T

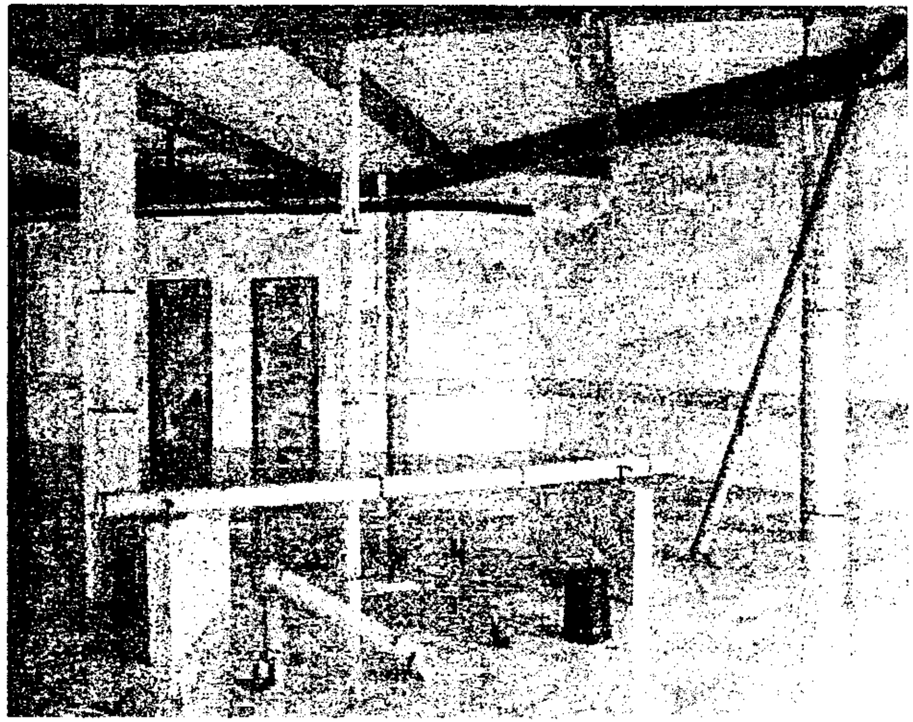
- electromagnetic radiation
  - visible light, 'ordinary' cameras, CCD
  - X-ray, tomography
  - laser scanner
  - nuclear spin resonance, MR, NMR
  - ...
- particle rays
  - electron microscophy
  - nuclear physics
  - ...
- ultrasound
- ...

Examples of Computer Vision problems

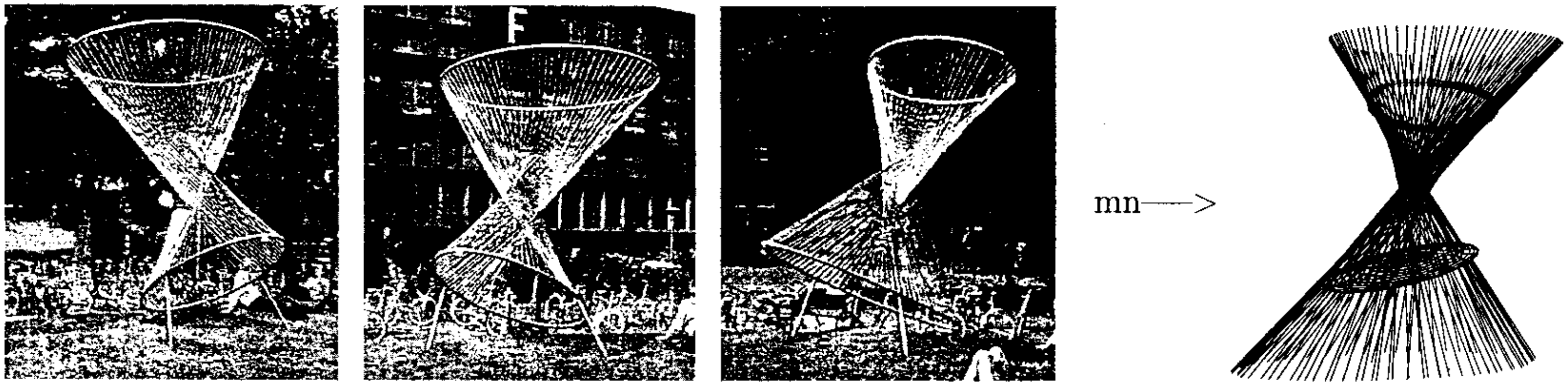
Visual control in robotics



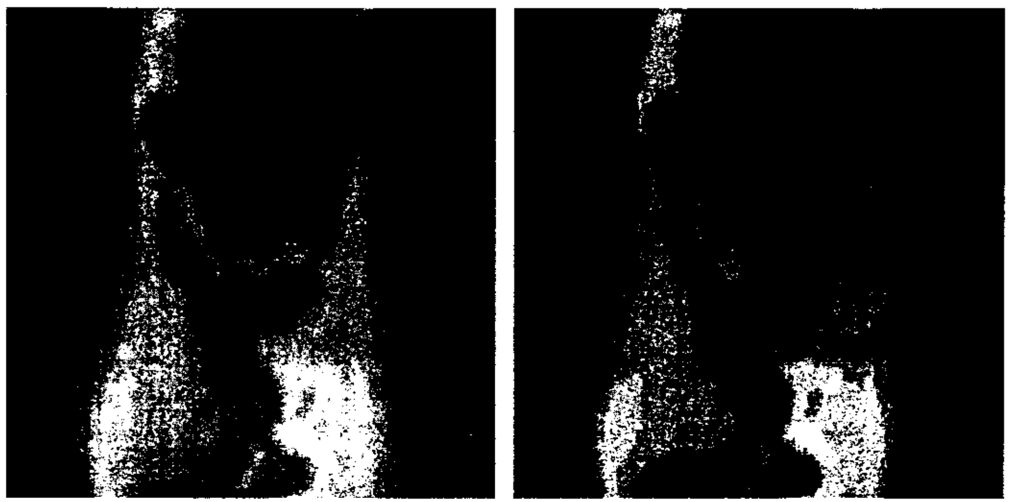
High precision industrial measurements (IMETRIC)



Scene reconstruction from image sequences



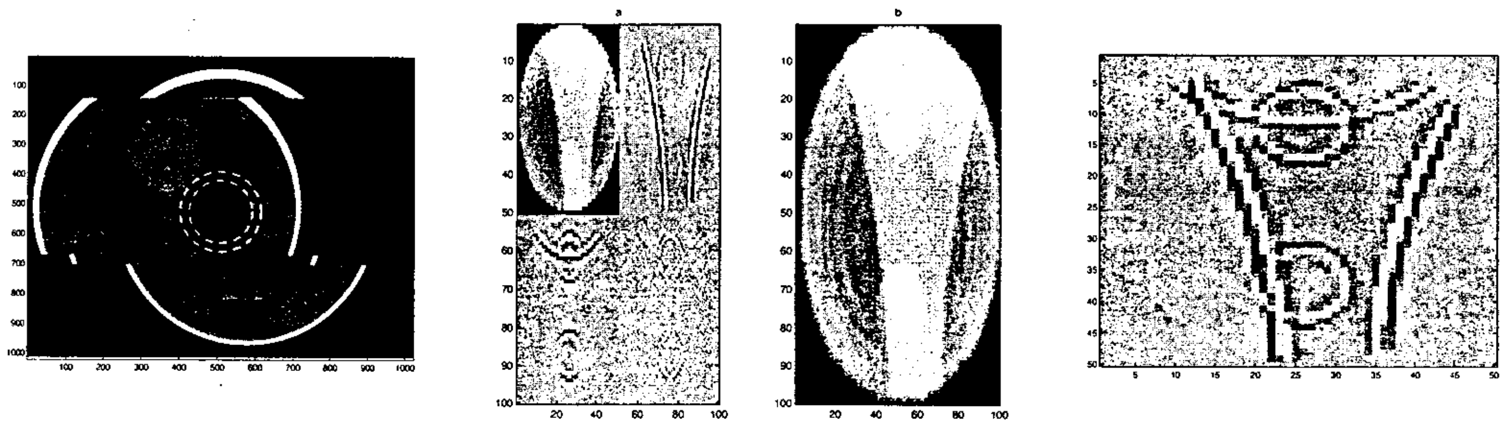
Combustion processes, segmentation by PDE



Analysis of car crash tests

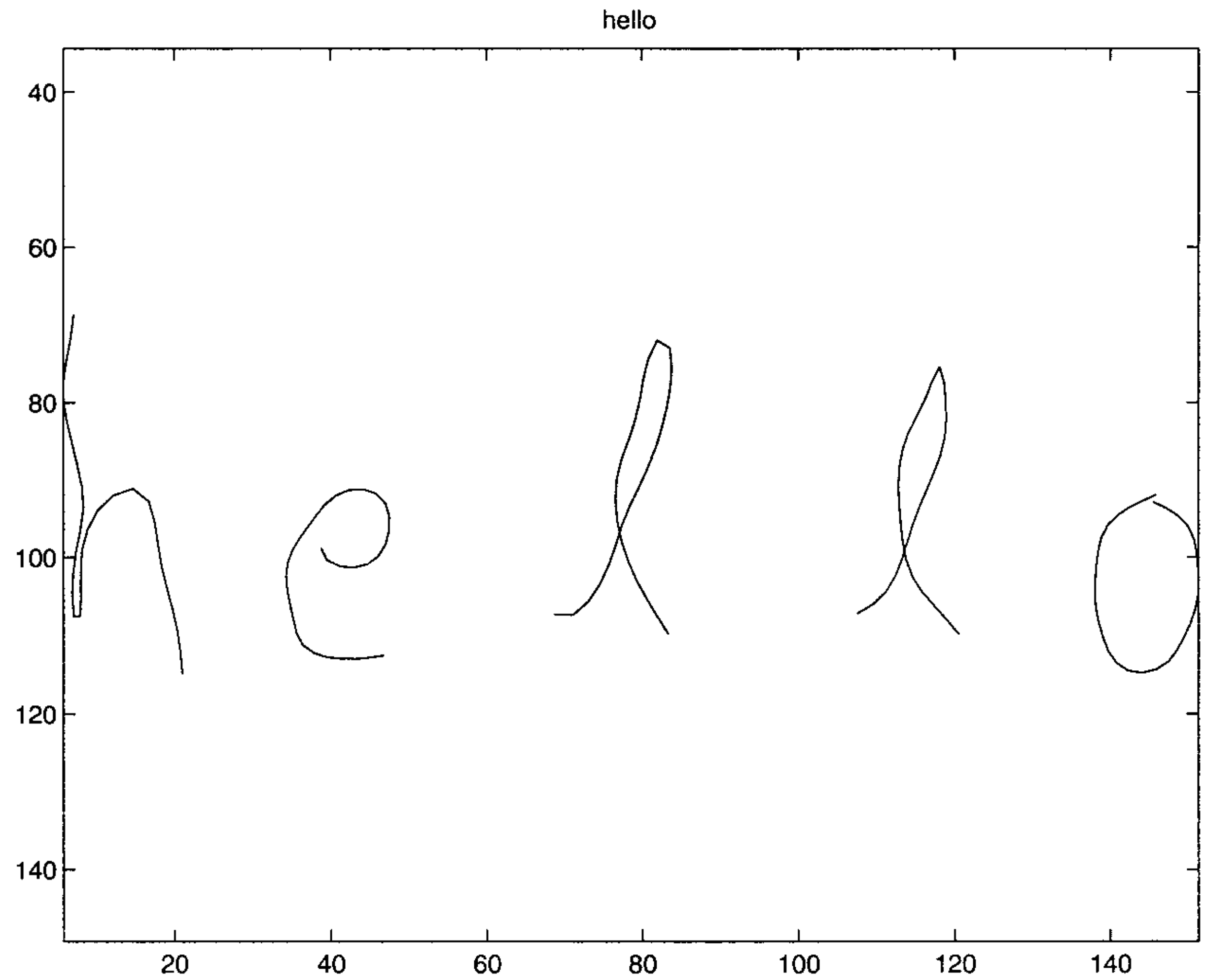
Augmented reality for city planning (Fraunhofer IGD)

Local tomography by wavelets

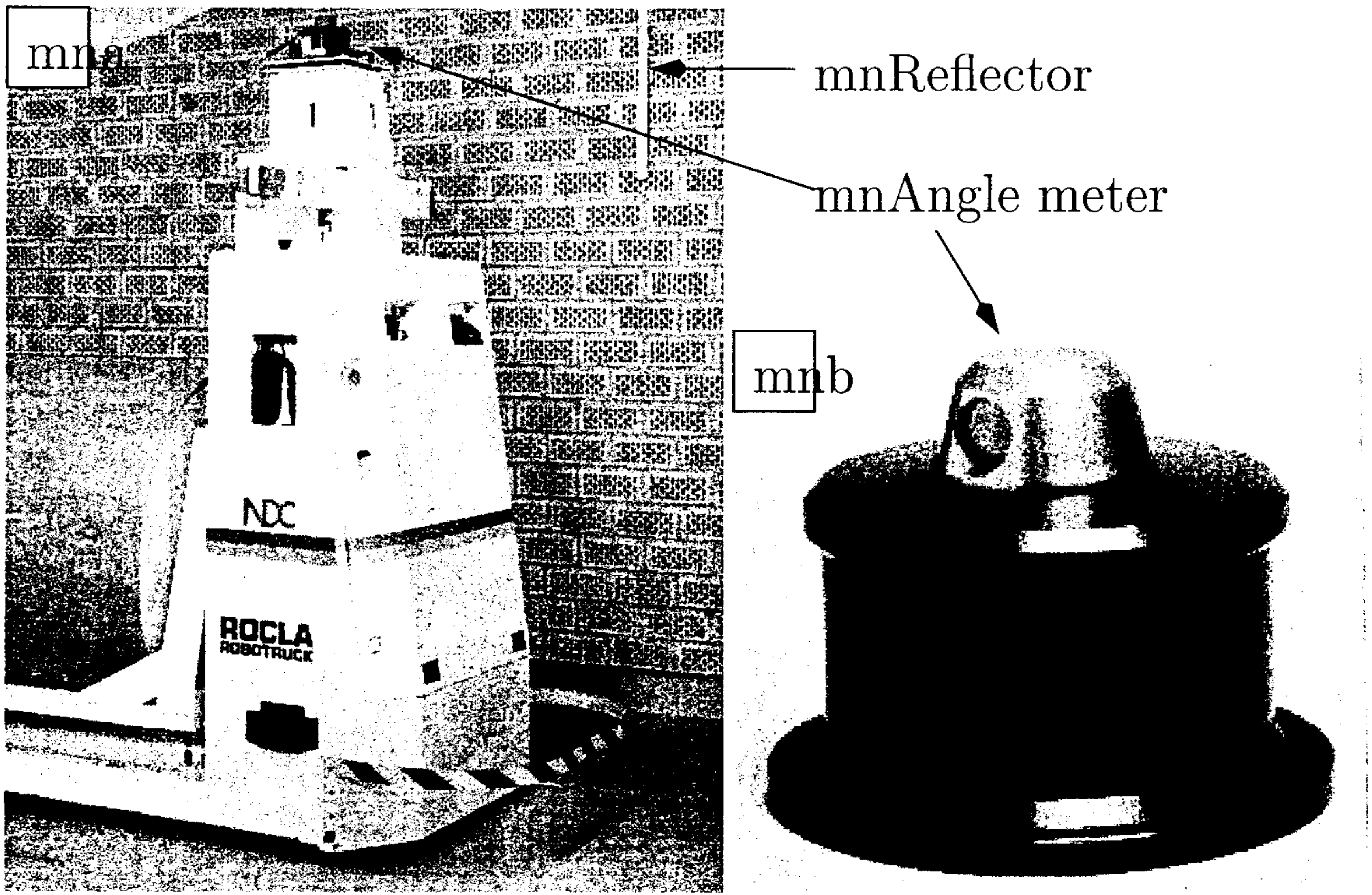


Perfusion and ventilation images of lungs for diagnosis of pulmonary embolism

Handwriting recognition

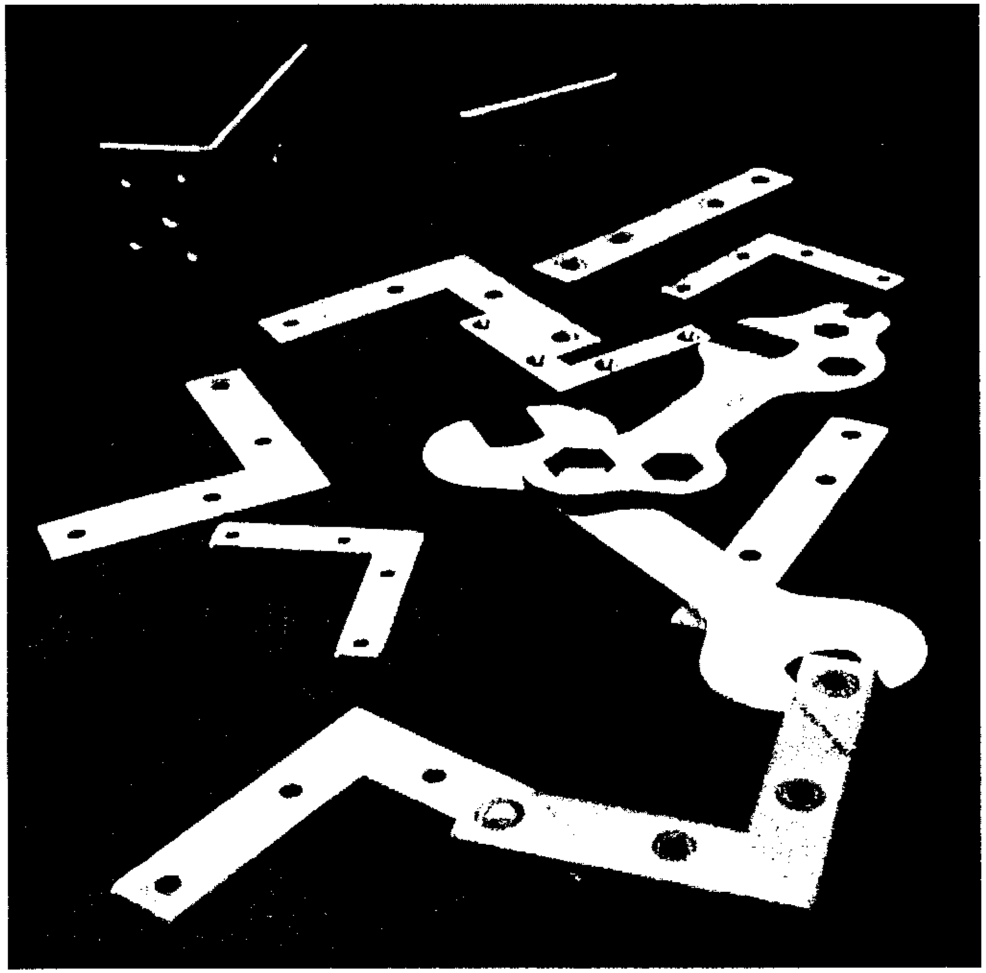


Laser guided navigation



The project

Deals with images like



Project description

A manufacturer produces ironwork and spanners of certain known shapes. The pieces are transported on a conveyor belt, from which they should be grasped and assorted by a robot. The belt is supervised by a video-camera under an oblique angle.

Goal: To present a solution of the visual recognition part of the problem, with as little manual interaction as possible.

System characteristics: Two interacting parts:

- Low level image processing. Feature extraction.
- High level recognition. Computer vision.

Output of first stage is input to second stage. The subsystems must be designed to fit each other.

Images: A number of images are taken in advance, and will be presented. Three types:

- Model images. One image of each piece.
- Simple scenarios. No occlusions. Levels of complexity:
  - few or many objects
  - planar or non-planar objects
- Complex scenarios. Occlusions. Levels of complexity:
  - few or many objects
  - planar or non-planar objects

Subsidiary condition: Change of production, e.g. new kinds of pieces, should be possible without rebuilding of the present recognition system.

Task: A number of images will be presented. Put identification markers on the objects in each picture.

Natural subdivision of the problem:

- *Low level task.* Reduce the information content of the images, extract relevant information, and find an efficient representation of the data for the
- *High level task.* Process the information to fulfil the recognition. Make an appropriate representation/visualization of the outcome.

For this problem:

- the low level task rests on *image processing*,
- the high level task profits on *projective geometry*.

Important *subtask*: to harmonize the output of the low level task with the input of the high level task.

2. Projective geometry

- Synthetic projective geometry
- Analytic projective, affine and Euclidean geometry
- Projective invariants
- Camera modeling

Cultural background

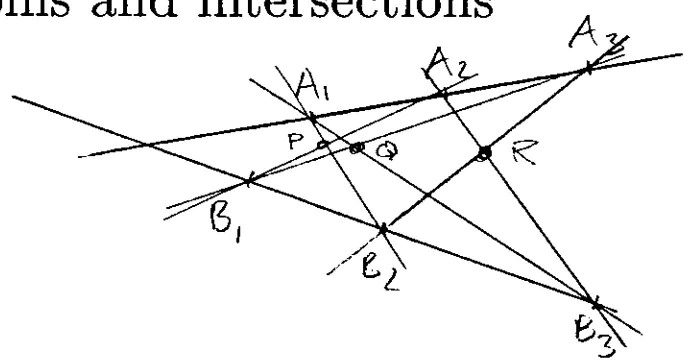
Ancient geometry culminated in the formation of Euclidean geometry

Euclid found it possible to deduce all geometric knowledge of the time from a number of *Axioms*

Euclid didn't explicitly state, but used the following

Parallel axiom: In the plane determined by a point P and a line p, there exists at least one line which passes through P and is parallel to (never meets) p

There are classical theorems not using this axiom, e.g. *Pappus theorem* on joins and intersections



Thm: P, Q, R collinear

Question: Is the parallel axiom a consequence of the other axioms?
No, non-Euclidean geometries were constructed in the 19th century

Pappus' theorem belongs to **projective geometry**, which deals with points and lines, and their joins and intersections, but not with parallelity, and with no concepts of lengths of segments or measures of angles

In *affine geometry*, parallelity exists, but not lengths and angles

Hiearchy: Euclidean $\subset$ Affine $\subset$ Projective geometry,
where $A \subset B$ means that A is more special than B in the sense that it has more axioms

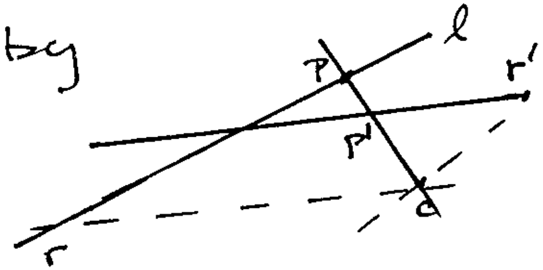
Another principle of division:

- *synthetic geometry* (uses only geometrical constructions, no coordinates)
- *analytic geometry* (coordinate geometry)

~~Synthetic projective geometry~~

Motivation: Given two lines ℓ, ℓ' in the plane, and a point c outside ~~both~~

<u>Def</u>: Perspective transf $P: \ell \to \ell'$ with center c defined by



$p \mapsto p'$

Not bijective $\ell \to \ell'$, exceptional points r, r'

Adjoin a <u>point at infinity</u> to ℓ (and ℓ')

$L = \ell \cup \{\infty\}$, $L' = \ell' \cup \{\infty\}$

P can be extended to a bijective map $L \to L'$ by $P: \infty \mapsto r'$, $P: r \mapsto \infty$

L and L' are models of the <u>projective line</u> $\mathbb{P}^1$

<u>Def</u>: <u>Projective transf</u> = composition of persp. trf

Possible to define proj. trf $L \to L$

<u>Coordinates on $\mathbb{P}^1$</u>

$\mathbb{P}^1$ = set of lines through a point c eplane

Plane $= \mathbb{R}^2$, coordinates (x_1, x_2)

$\mathbb{P}^1 = \mathbb{R}^2 \setminus \{0\}$ where all $t(x_1, x_2)$, $t \neq 0$ are identified (since they describe the same line.)

This is <u>Homogeneous coordinates</u> on $\mathbb{P}^1$

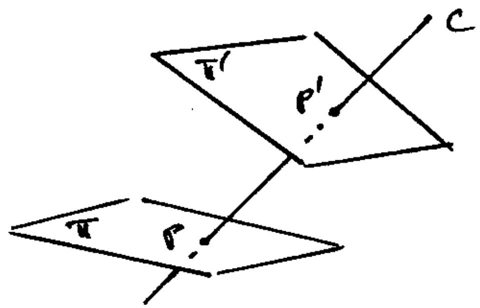
<u>Thm</u>: Projective transf on $\mathbb{P}^1$ can be written

$\lambda y = Ax$

where A is 2×2, λ scalar

The projective plane $\mathbb{P}^2$

Analogous to $\mathbb{P}^1$,
Perspective transf between planes in space



$p \mapsto p'$

Not bijective. Lines of exception, consisting of points at infinity, one in each direction of π (and π'). Form the line at infinity.

Model of projective plane: $\Pi = \pi \cup \underbrace{\{\text{points at infty}\}}_{\text{line at inf.}}$

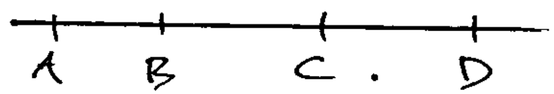
$\mathbb{P}^2 = \mathbb{R}^3 \setminus \{0\}$, with identification of $t(x_1, x_2, x_3)$, $t \neq 0$

Projective transf described by $\lambda y = Ax$,
with A 4×4, λ scalar, in homogeneous coordinates

Projective invariants

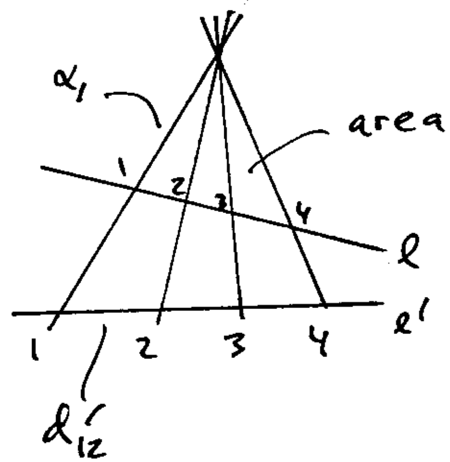
On the line:

Def: Cross-ratio $k = \frac{AC}{BC} \Big/ \frac{AD}{BD}$



Theorem Cross-ratio invariant under proj. trf.

Proof:



$$A_{jk} = c_{jk}\alpha_j\alpha_k = h\, d_{jk}$$

$$A'_{jk} = c_{jk}\alpha'_j\alpha'_k = h'\, d'_{jk}$$

$$\frac{A_{12}A_{34}}{A_{13}A_{24}} = \frac{c_{12}\alpha_1\alpha_2\, c_{34}\alpha_3\alpha_4}{c_{13}\alpha_1\alpha_3\, c_{24}\alpha_2\alpha_4} = \frac{d_{12}\,d_{34}}{d_{13}\,d_{24}} = k_\ell$$

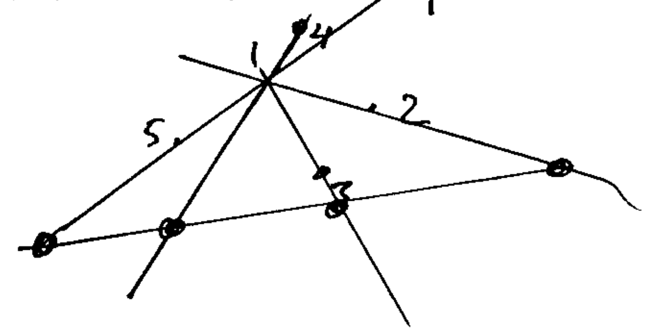
$$\parallel \qquad\qquad \Rightarrow k_\ell = k_{\ell'}$$

$$\frac{A'_{12}A'_{34}}{A'_{13}A'_{24}} = \text{the same} = \frac{d'_{12}\,d'_{34}}{d'_{13}\,d'_{24}} = k_{\ell'}$$

Remark: Permutation of ordering gives cross-ratios $k, \frac{1}{k}, 1-k, \frac{1}{1-k}, \frac{k}{k-1}, \frac{k-1}{k}$

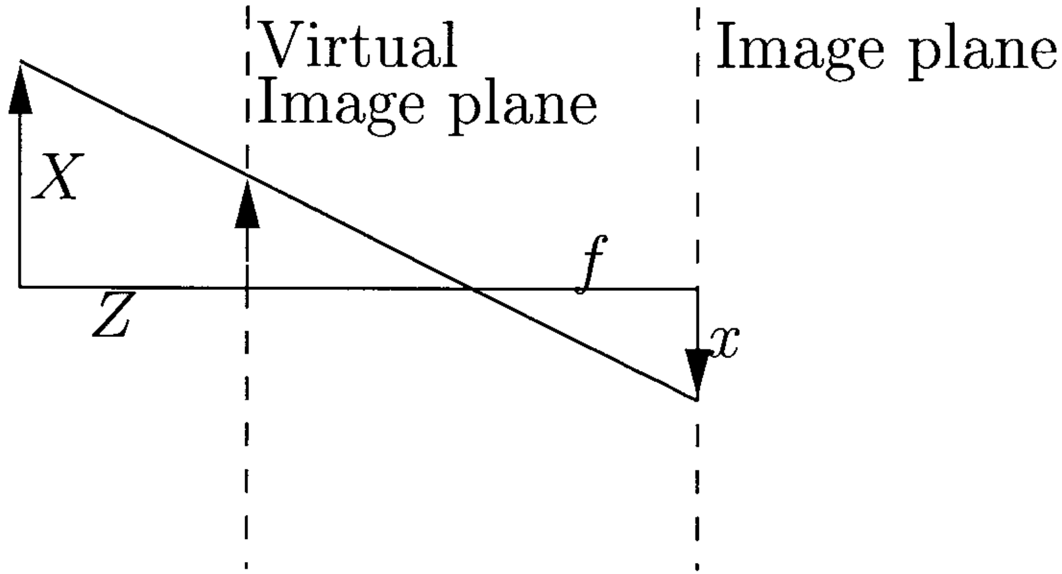
In the plane:

5 points. From one of the points, draw lines through the others, intersect with an arbitrary line, and compute the cross-ratio. Generates all invariants for 5 points



The pinhole camera

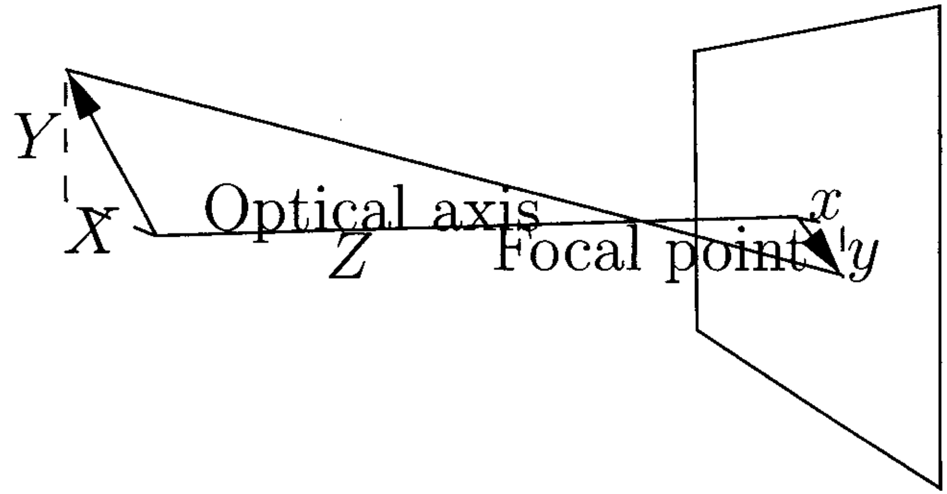
This is a device for singular transformations $\mathbb{P}^3 \longmapsto \mathbb{P}^2$



Convention: Work with a virtual image plane in front of the focal point. (A similar switching is done in the brain.)

Triangular similarity gives $x = fX/Z$, where f is the focal length.

The pinhole camera in 3D



$$(X, Y, Z) \mapsto (x, y) = (fX/Z, fY/Z) \iff Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

In homogeneous coordinates and block matrix notation, with $f = 1$:

$$\lambda v = \begin{bmatrix} I & | & 0 \end{bmatrix} V$$

Change of Object coordinate system

Suppose the ambient space $\mathbb{R}^3$ has a Euclidean structure.
Euclidean change of coordinates ($\mathbb{R}^3 \to \mathbb{R}^3$):

$$y = Rx + t \ ,$$

where R is a 3×3 rotation matrix ($R^T R = I$, $\det(R) = 1$), and t is a translation vector 3×1.

With homogeneous coordinates:

$$V = \begin{bmatrix} y \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = TU$$

The projection from U to v is then

$$v = \begin{bmatrix} I & 0 \end{bmatrix} V = \begin{bmatrix} I \mid 0 \end{bmatrix} TU = \begin{bmatrix} R \mid t \end{bmatrix} U$$

Change of image coordinate system

Avoid assuming Euclidean structure on the image plane (due to defects in LCD array)

Affine change of image coordinates ($\mathbb{R}^2 \to \mathbb{R}^2$):

$$y = Ax + b \Longleftrightarrow u = \begin{bmatrix} y \\ 1 \end{bmatrix} = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = Kv$$

where A is 2×2 matrix, and b is a translation vector 2×1.

In the new *homogeneous* coordinates, U and u, the projection is expressed by

$$u = Kv = K \left[R \mid t \right] U$$

Definition: *Camera matrix* (3×4)

$$P = K \left[R \mid t \right]$$

a

Example. What is the projection of the point $(5,2,2)$ with a camera represented by camera matrix

$$\lambda x = PX\ , \quad P = \begin{pmatrix} 9 & 4 & 4 & 4 \\ 7 & 9 & 9 & 8 \\ 2 & 9 & 1 & 0 \end{pmatrix}\ .$$

Then

$$X = \begin{pmatrix} 5 \\ 2 \\ 2 \\ 1 \end{pmatrix}, \qquad PX = \begin{pmatrix} 65 \\ 79 \\ 30 \end{pmatrix} \Longrightarrow \lambda = 30\ , \quad x = \begin{pmatrix} 13/6 \\ 79/30 \\ 1 \end{pmatrix}$$

The projected point is $(13/6, 79/30)$.

Parameters of the camera matrix:

$$P = K[\,R\,|\,t\,] \ .$$

The extrinsic parameters:

R - orientation of camera.

t - position of camera.

The intrinsic parameters:

$$K = \begin{bmatrix} \gamma f & sf & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{where} \quad \begin{array}{ll} f & \text{focal length} \\ s & \text{skew} \\ \gamma & \text{aspect ratio} \\ (x_0, y_0) & \textit{principalpoint} \end{array}$$

Calibrated camera: K known.

Uncalibrated camera: K unknown.

The focal point

How can the focal point be computed from the camera matrix?

The focal point is the right null space of the camera matrix.

$$f = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \qquad Pf = 0 \ .$$

A common convention is to write

$$f = \begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix}, \qquad P = [R \,|\, (-R\mathbf{t})] \ .$$

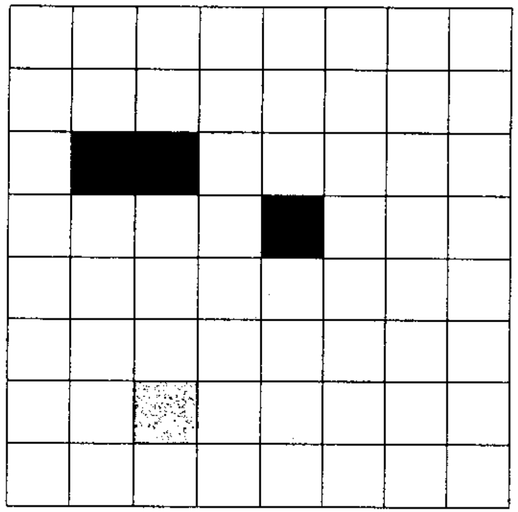
Notice that

$$Pf = R\mathbf{t} - R\mathbf{t} = 0$$

3. Image processing

Representation of *digital images.*

Raster graphics



Matrix representation

$$\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ b_{21} & b_{22} & \cdots & b_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ b_{M1} & b_{M2} & \cdots & b_{MN} \end{pmatrix}$$

The squares are called **pixels**, obtained by *sampling.*

b_{ij} depicts **graylevel**, usually *digitized* between $0 =$ black and $255 =$ white (8 bits = 1 B)

Typical amount of data for a black&white picture: $512 \times 512 \times 1 \approx 250kB$

Colour images are built up by three colour scales: red, green, blue

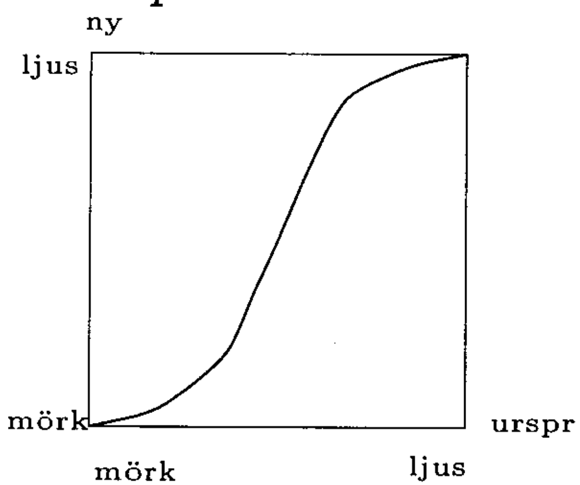
Typical amount of data for a colour picture $512 \times 512 \times 3 \approx 750kB$
$\approx$ half a discette

Video sequences: Enormous data handling problems.

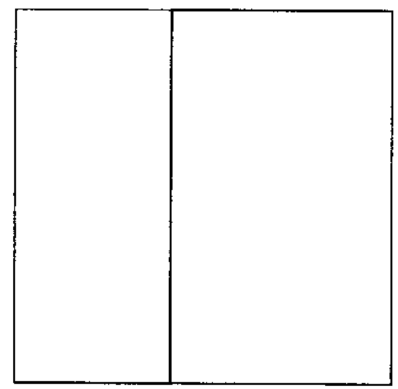
3. Graylevel transformations

Change the graylevel interpretation of pixelvalues b_{ij}, **bitmap**

Example: Increase of contrast and *threshholding*

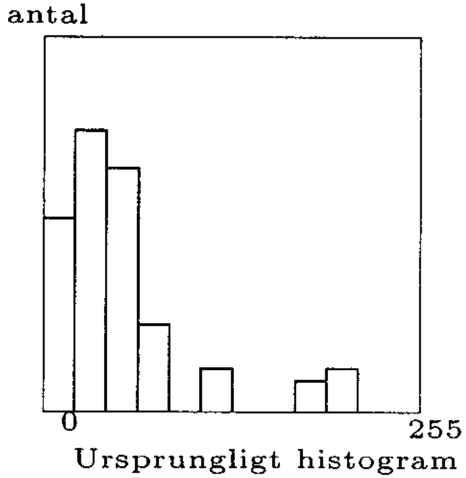


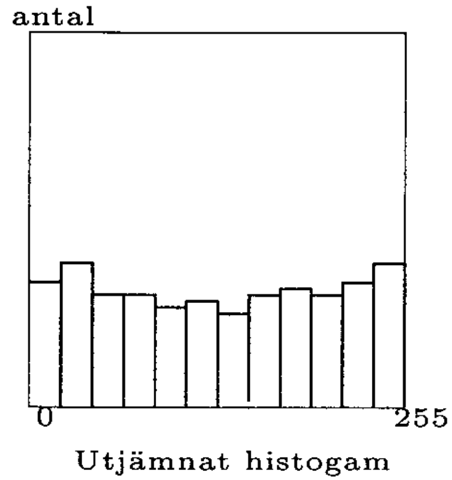
Kontrastökning



Tröskling

Grayish, too dark, or too light pictures, can sometimes be improved by **histogram equalization**





Filtering

A. Filtering in the signal domain

Operate on the image B by a **template** (or **mask**)

$$M = \boxed{\begin{matrix} m_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{nn} \end{matrix}}$$

The template is moved over the image, row by row. In each position:
Each template digit and the corresponding pixel value are multiplied. The products are summed and stored in an *output matrix*, describing the *filtered image*.

Example: Averaging (smoothening, reduces noise and details)

$$\frac{1}{9}\boxed{\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array}}$$

Example 2: x-derivative (detects vertical edges), combined with smoothening in the y-direction.
Approximation of $\frac{\partial}{\partial x}$

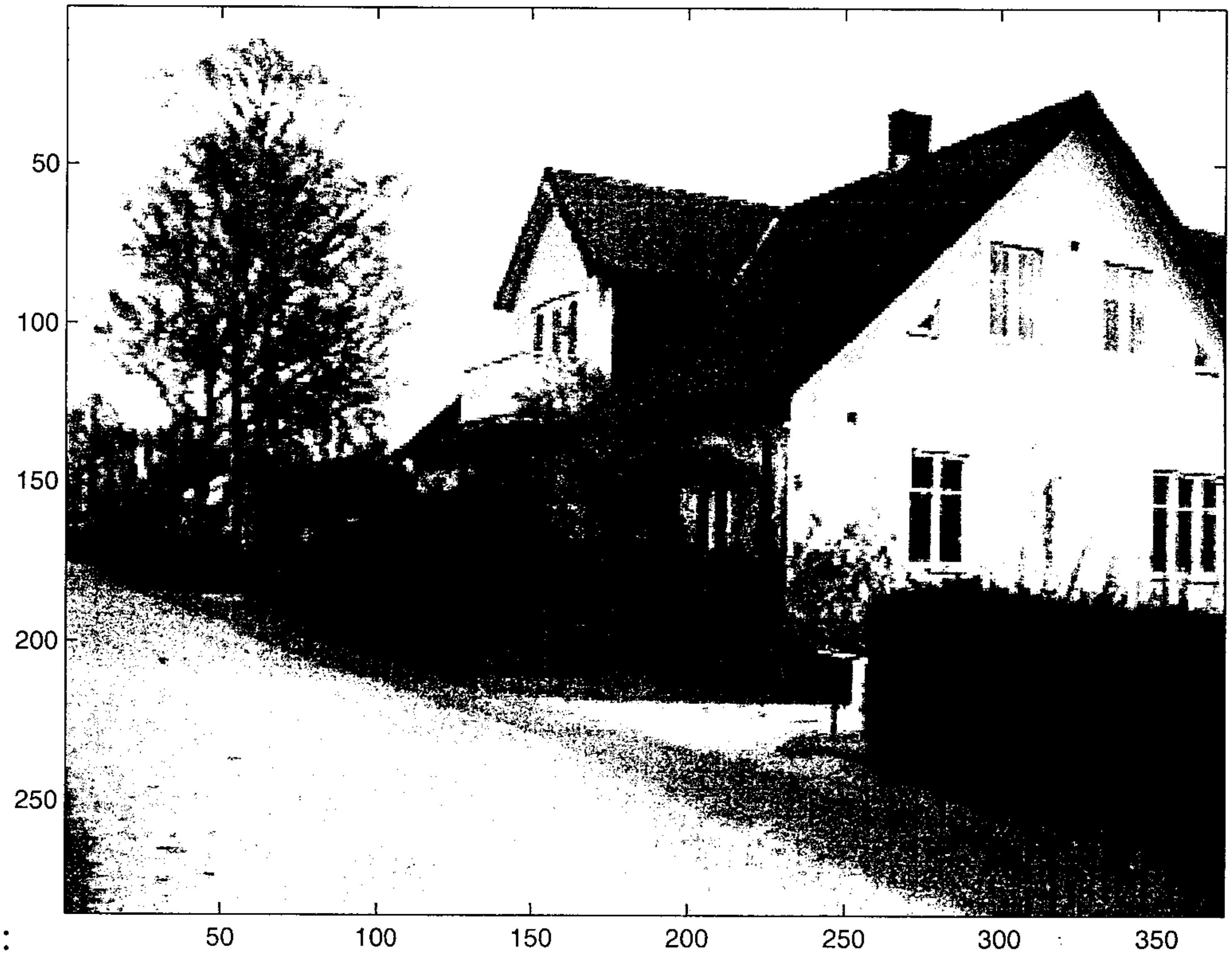
$$\boxed{\begin{array}{ccc} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{array}}$$

5. Filtering in the signal domain (contd)

Exempel 3: Laplacian operator (detects edges).
Approximation of $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$.

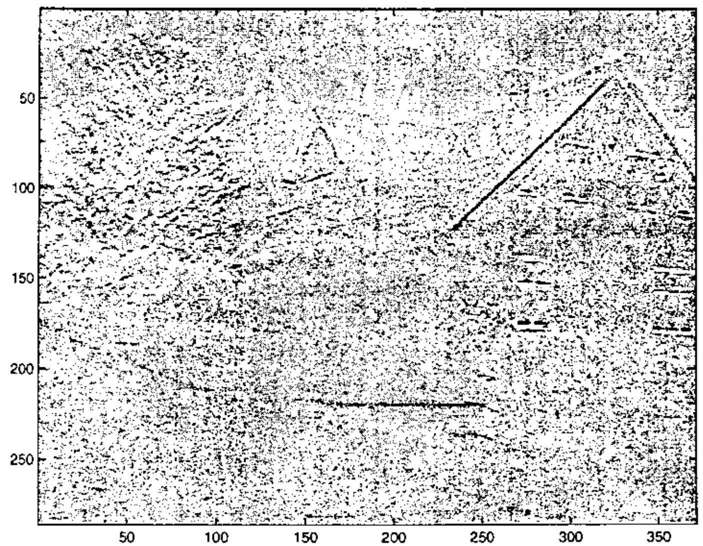
$$\frac{1}{8}\boxed{\begin{array}{ccc} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{array}}$$

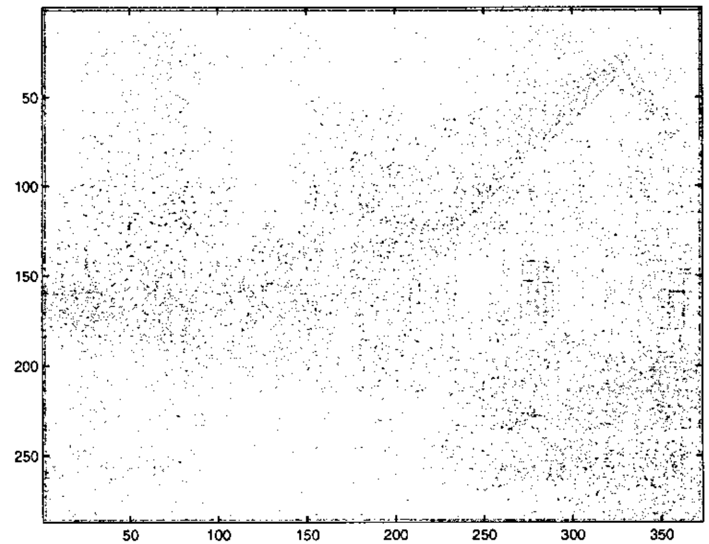
Example: This is my house



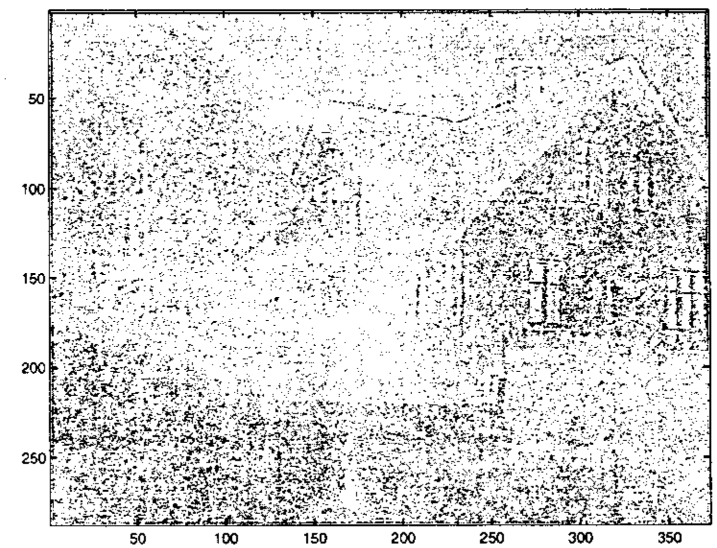
Original:

Output from different filterings:

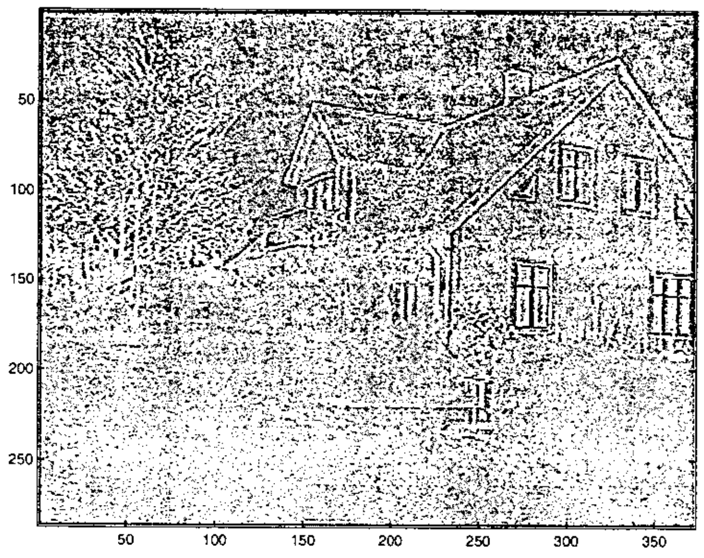
A: 

B: 

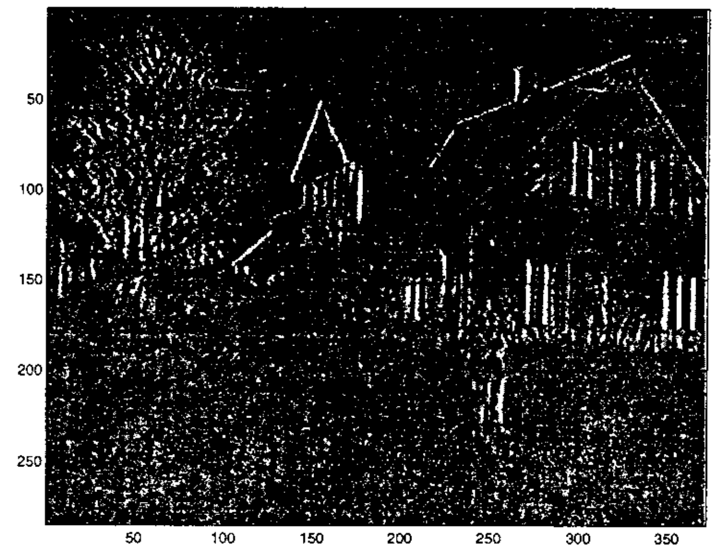
C: 

D: 

E:



F:



Which image corresponds to which of the following templates?

$$a = \boxed{\begin{array}{c} 1 \\ -1 \end{array}} \qquad b = \boxed{1\ \text{-}1}$$

$$c = \frac{1}{8}\boxed{\begin{array}{ccc} 1 & 1 & 1 \\ 1 & -9 & 1 \\ 1 & 1 & 1 \end{array}} \qquad d = \frac{1}{4}\boxed{\begin{array}{ccc} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{array}}$$

$$e = \frac{1}{16}\boxed{\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{array}} \qquad f = \frac{1}{8}\boxed{\begin{array}{ccc} 1 & 1 & 1 \\ 1 & -7 & 1 \\ 1 & 1 & 1 \end{array}}$$

B. Filtering in the frequency domain

Definition. Let f be a function from $\mathbb{R} \to \mathbb{R}$. The Fouriertransform of f is defined by

$$(\mathcal{F}f)(u) = F(u) = \int_{-\infty}^{+\infty} e^{-i2\pi xu} f(x)dx \ .$$

Theorem. Under appropriate assumptions on f, the inversion formula holds true:

$$f(x) = \int_{-\infty}^{+\infty} e^{i2\pi ux} F(u)du \ .$$

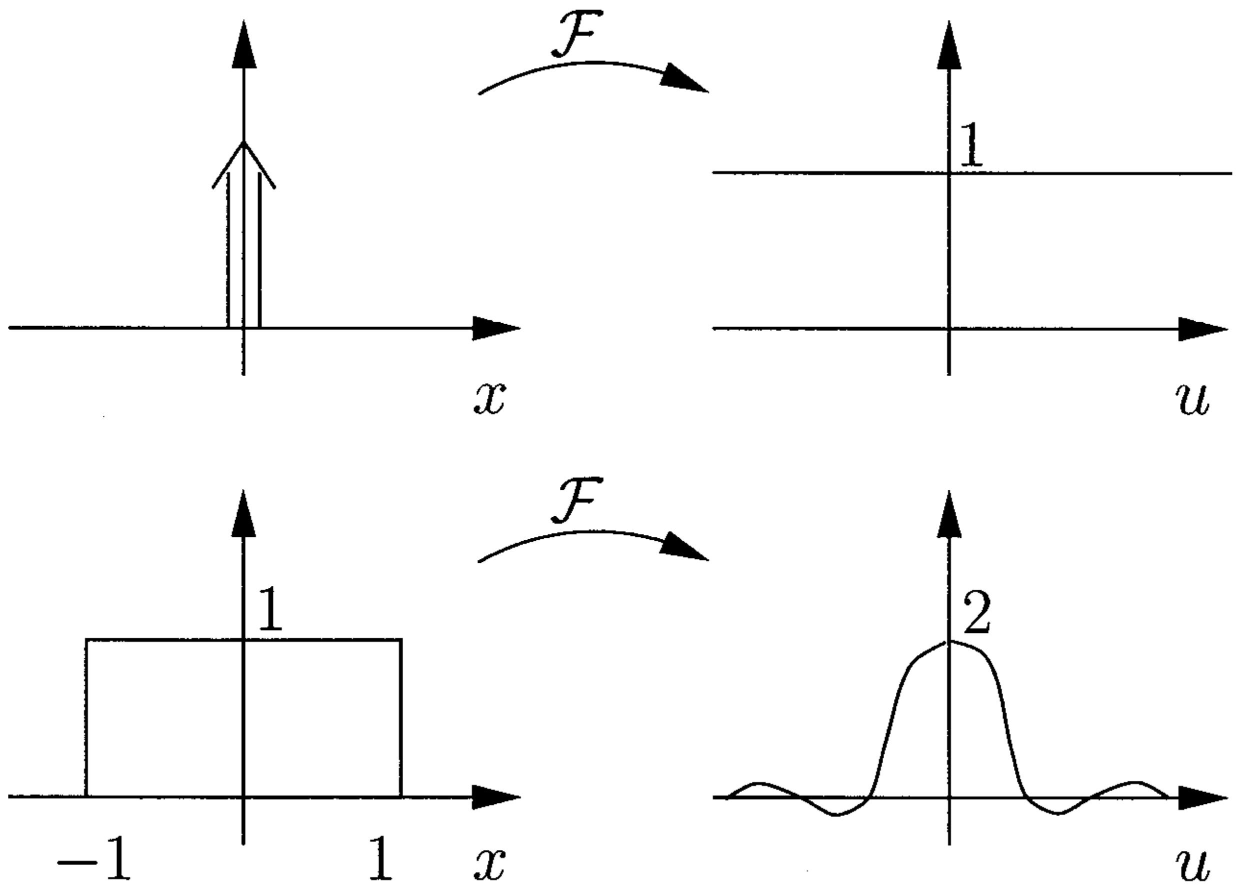
The Fourier inversion formula describes how an image f is built up by different frequences:

For *discrete* signals: Discrete Fourier transform.
Efficient numerical technique: **FFT, Fast Fourier Transform**.

Example.

$$\delta(x) \mapsto 1(u)$$

$$\mathrm{rect}(x) \mapsto 2\frac{\sin(2\pi u)}{2\pi u} = 2\,\mathrm{sinc}(2\pi u)$$



Twodimensional Fouriertransform

Definition. Let $f(x,y)$ be a function $\mathbb{R}^2 \to \mathbb{R}$. The Fouriertransform of f is defined by

$$\mathcal{F}f(u,v) = F(u,v) = \int_{-\infty}^{+\infty} e^{-i2\pi(ux+vy)} f(x,y)dxdy \ .$$

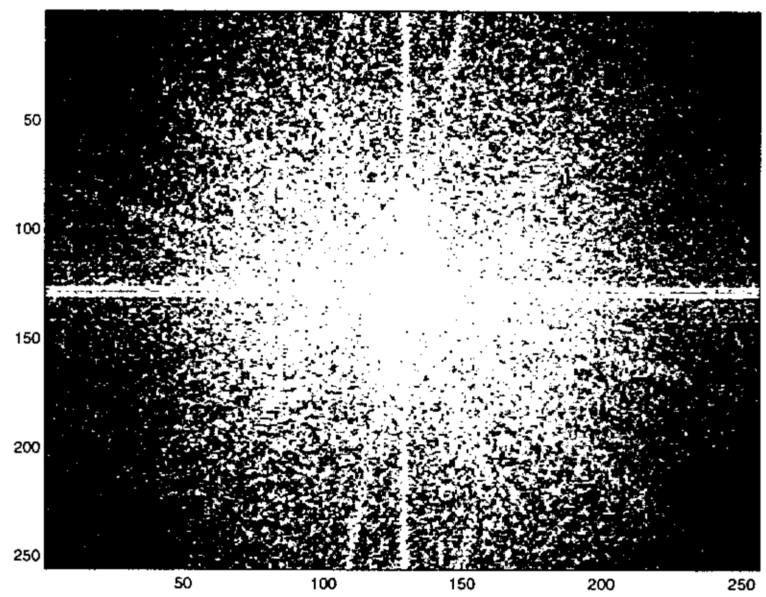
Can also be written, with $\mathbf{u} = (u,v)$, $\mathbf{x} = (x,y)$:

$$F(\mathbf{u}) = \int_{-\infty}^{+\infty} e^{-i2\pi\mathbf{u}\cdot\mathbf{x}} f(\mathbf{x})d\mathbf{x} \ .$$

Theorem. Under appropriate assumptions on f, the inversion formula holds true:

$$f(x,y) = \int_{-\infty}^{+\infty} e^{i2\pi(ux+vy)} F(u,v)dudv \ .$$

Example. An image and its Fouriertransform (absolutevalue)



Fine details correspond to high frequences

Coarse structures correspond to low frequences

Example.

$$\begin{aligned}
\operatorname{rect}(x)\operatorname{rect}(y) &\mapsto 4\operatorname{sinc}(2\pi u)\operatorname{sinc}(2\pi v)\\
\delta(x)1(y) &\mapsto 1(u)\delta(v)\\
\delta(x)\operatorname{rect}(y) &\mapsto 1(u)2\operatorname{sinc}(2\pi v)\\
f(x-1)+f(x+1) &\mapsto (e^{-i2\pi u}+e^{i2\pi u})F(u) =\\
&= 2\cos(2\pi u)F(u)
\end{aligned}$$

Note that F is complex valued in general.

Useful knowledge: Since images f are real:

- even $f \mapsto$ real F
- odd $f \mapsto$ imaginary F
- $F(u) = \overline{F(-u)}$

Relation between filtering in the signal and frequency domains

Filtering in the signal domain is a convolution.

Theorem. If

$$g(x) = h * f = \int h(x-y)f(y)dy$$

and $\mathcal{F}g = G$, $\mathcal{F}h = H$, $\mathcal{F}f = F$, then

$$G = HF \ .$$

Definition. $H = \mathcal{F}$ is called the *frequency function* of the filter.

Filters for image enhancement:

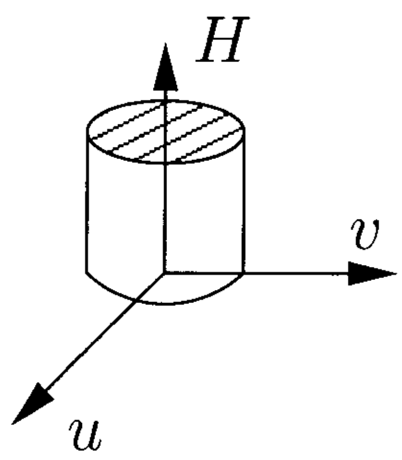
signal domain	frequency domain
smoothening	lowpass
sharpening	highpass

Example. Ideal lowpass filter

The frequency function

$$H_{D_0}(u) = \begin{cases} 1, & |u| \leq D_0, \\ 0, & |u| > D_0, \end{cases}$$

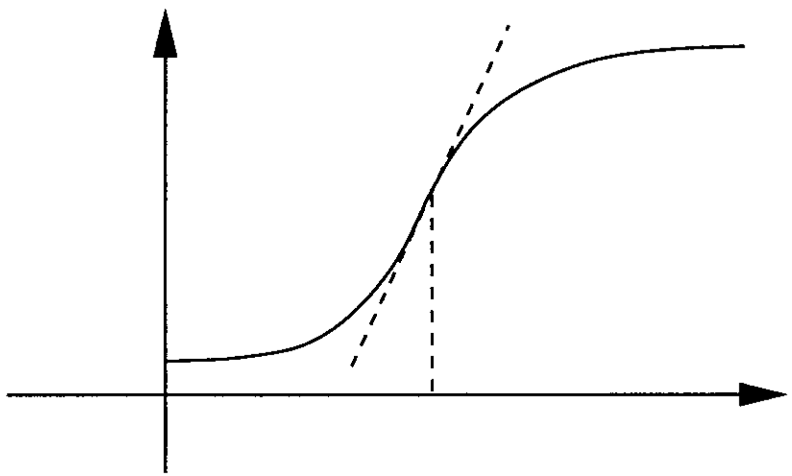
defines an **ideal lowpass filter (ILPF)**, with **"cut-off" frequency** D_0.



Edge detection

What to mean by an edge? Look at the derivative

In one dimension: location of edge = maximum of derivative



In two dimensions: Make a discrete approximation of

$$(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2 = |\nabla f|^2$$

If $|\nabla f|^2 >$ threshold, mark as edgepoint. Also gives the direction of the gradient, i.e. the normal of level curves or edges

There is a family of methods for edge detection, using different discrete approximations of derivatives

Some examples:

Example. A Roberts mask. The operation

$$g(x,y) = f(x,y) - f(x-1,y-1)$$

corresponds to convolution with the mask

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

oblique differentiation

Example. A Prewitt mask. The operation

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

- means differentiation and smoothening in x
- smoothening in y

Example. A Sobel mask. The operation

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

- means differentiation and smoothening in x
- smoothening in y

Example. Laplacean filter

Let

$$h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \text{and} \quad g = \frac{1}{8} h * f \ .$$

Hence

$$g(x,y) = f(x,y) - \text{average}(f) \ .$$

- $g = 0$ on regions where f is constant
- $|g|$ is large on regions where f varies much

This is the discrete analogue of the Laplacian

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \ .$$

Edge detection by means of Sobel and Laplacean

Advanced methods for edge detection

Canny-Deriche: Based on a compromise between

- detection of reliable edges
- no false detections of non-existing edges (noise)
- good localization of edges

Suppose that the edge is described by $G(x)$ (1D). Quantification of detection and localization leads to a variational problem:
Determine f that maximizes

$$\frac{|\int G(-x)f(x)dx||\int G'(-x)f'(x)dx|}{(\int f^2 dx)^{1/2}(\int (f')^2 dx)^{1/2}}$$

The Hough transform

Goal: Find line-structures in a picture

Often used on edge-images

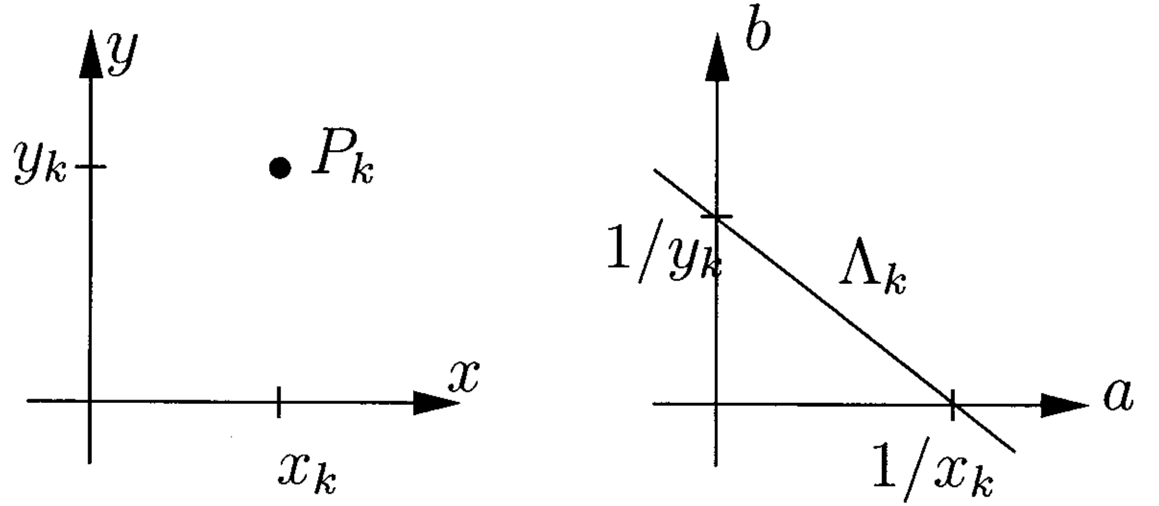
$$l_{a,b} : \quad ax + by = 1 \quad (\text{Suppose } 0 \notin l)$$

$$(x_k, y_k) \in l_{a,b} \quad \Leftrightarrow \quad ax_k + by_k = 1$$

Let

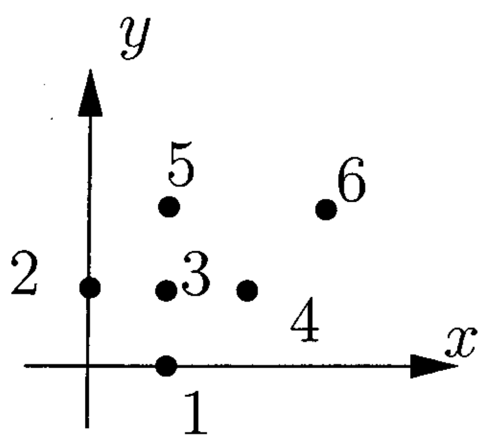
$$\Lambda_k = \{\, (a,b) \,|\, (x_k, y_k) \in l_{a,b} \,\}$$

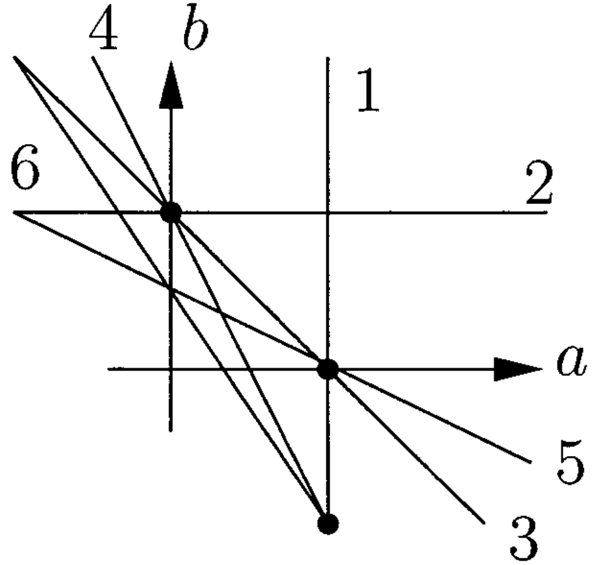
which forms a line in the *ab*-plane.



Idea: Cover the ab-plane by an array of 'accumulator cells' and add 1 to all cells that are intersected by Λ_k.

Exempel 0.1. *Consider the points*





$$\Lambda_1 : a = 1 \qquad \Lambda_2 : b = 1$$
$$\Lambda_3 : a + b = 1 \qquad \Lambda_4 : 2a + b = 1$$
$$\Lambda_5 : a + 2b = 1 \qquad \Lambda_6 : 3a + 2b = 1$$

There are three lines throgh the points $(0,1)$, $(1,0)$ *och* $(1,-1)$, *corresponding to the lines*

$$y = 1 \quad x = 1 \quad x - y = 1$$

Note that in general the Houghtransform has to be thresholded to yield interesting lines.

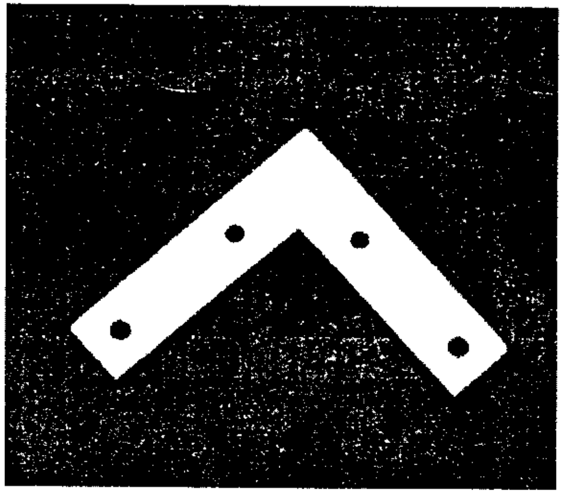
Preparation for the project

To represent images, many different formats are in use, discriminated by suffix, .tif, .jpg, .gif, .pgm, etc. Some of them invoke compression

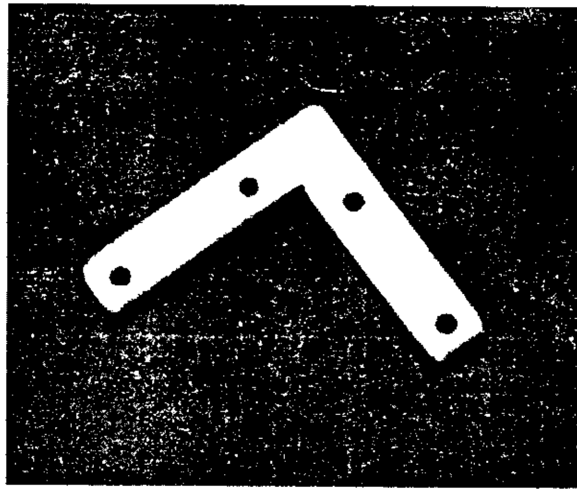
Many different programs for conversion between image formats exist. MATLAB uses `imread`, which can read .bmp, hdf, jpeg, pcx, tiff, xwd.

The images of the project are given in .tiff

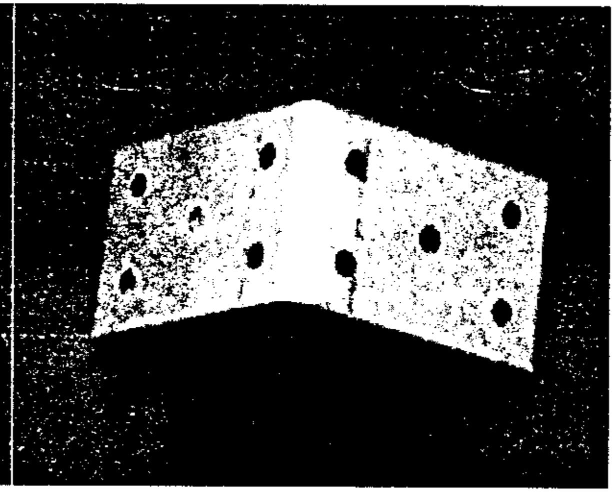
Two kinds of images: Model images, of one ironwork, and task images, containing several pieces



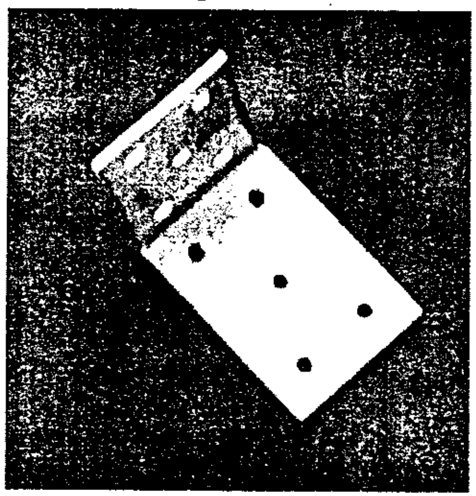
Model 45



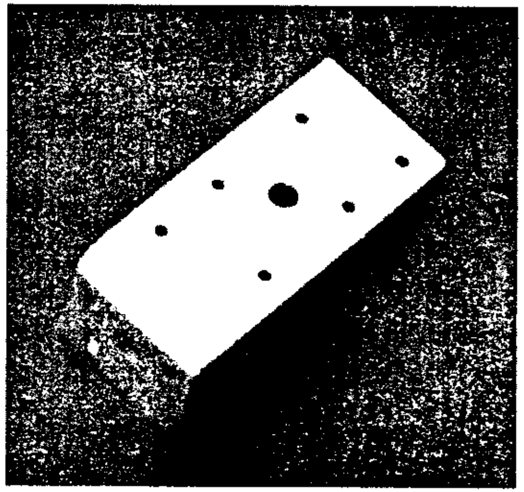
Model 46



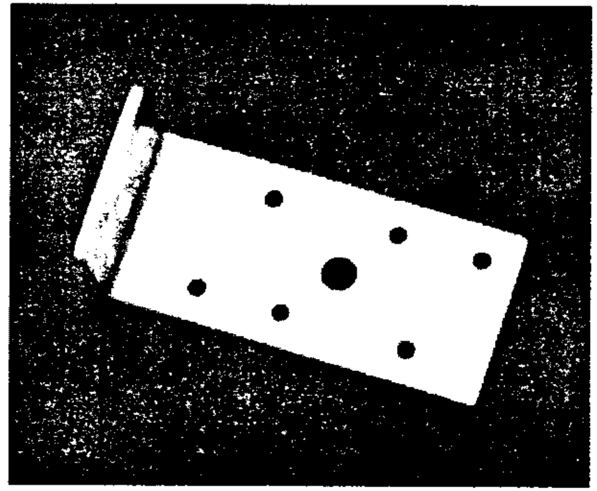
Model 47



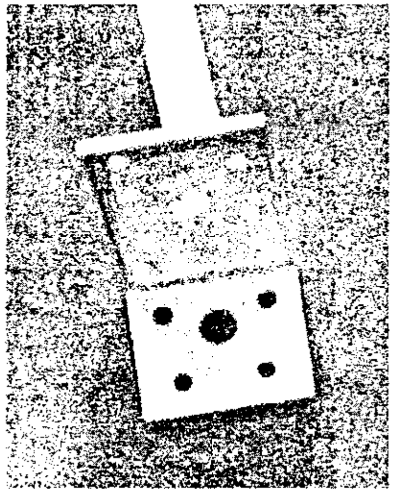
Model 47



Model 49

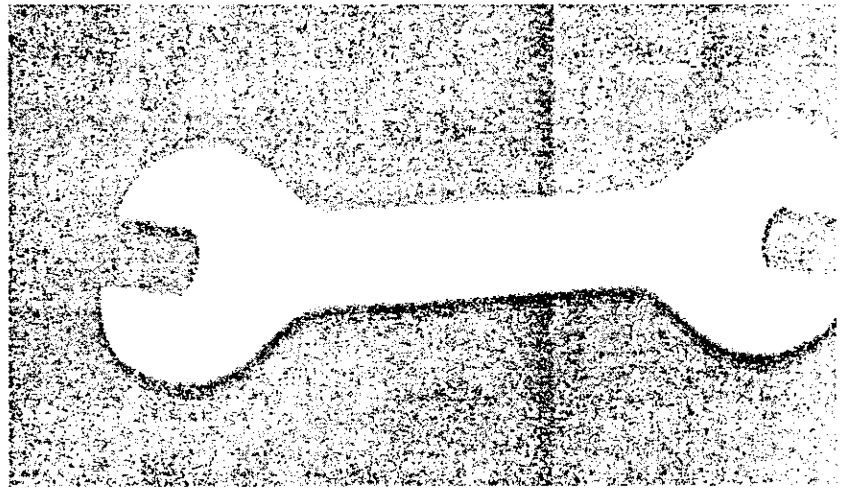


Model 49

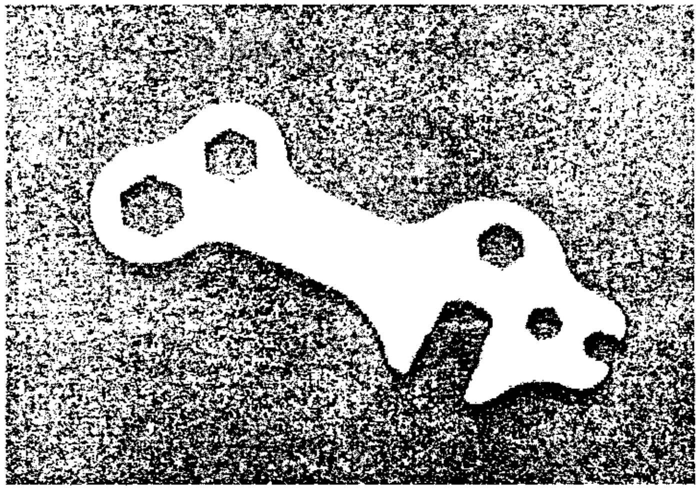


Model 49

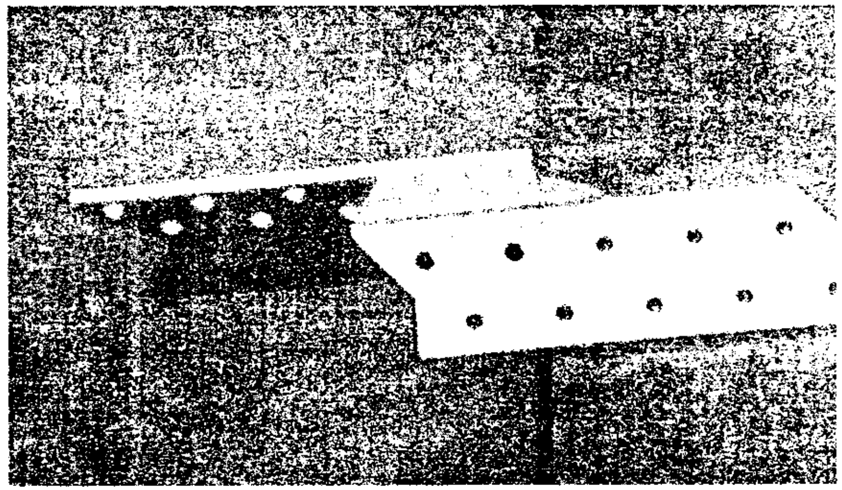
Model 55



Model 52



Model 53



Model 55

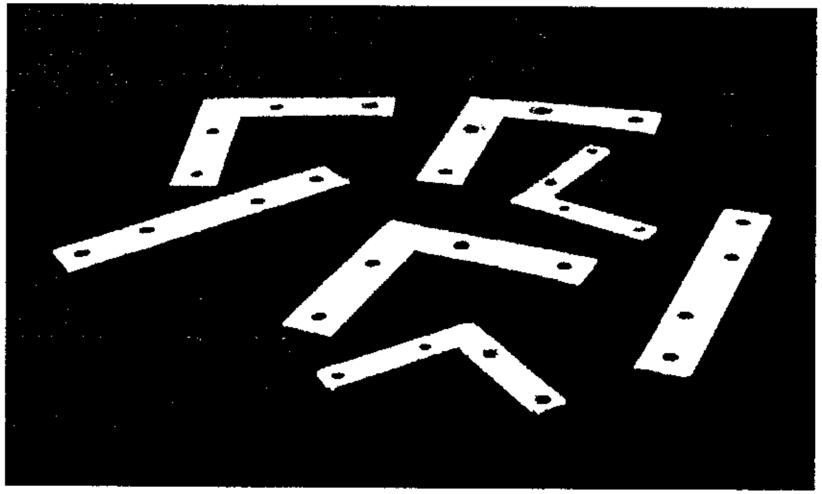


Image 56

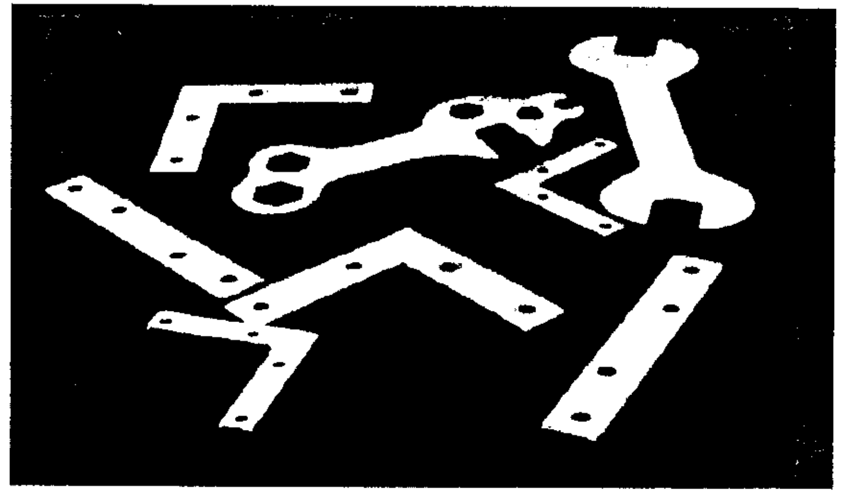


Image 57

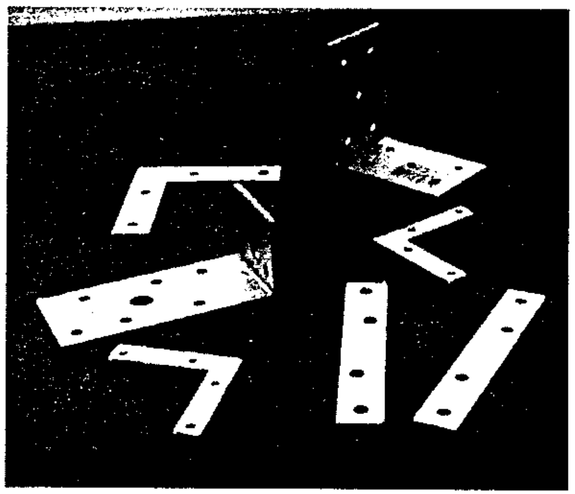


Image 58

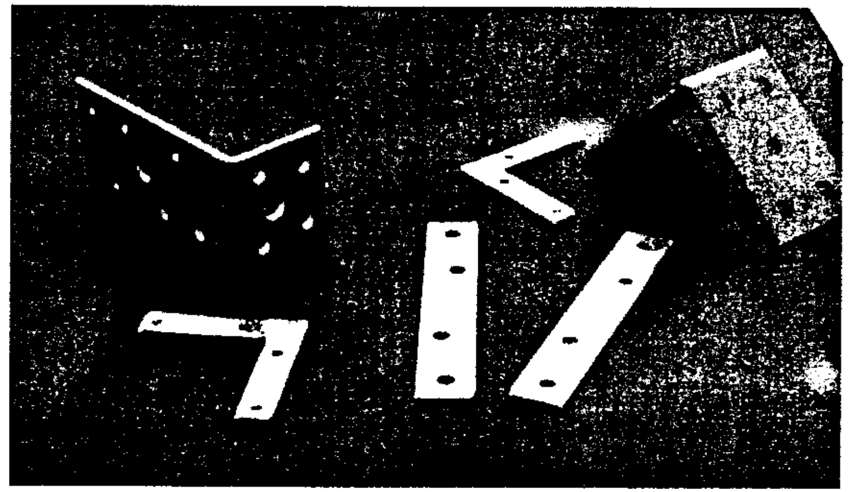


Image 59

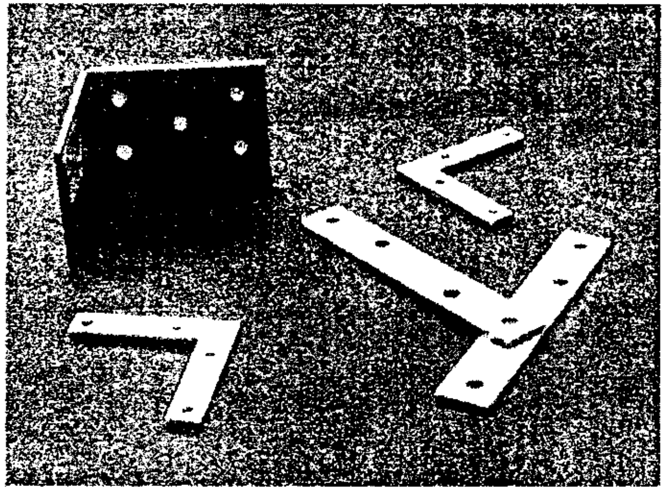


Image 60

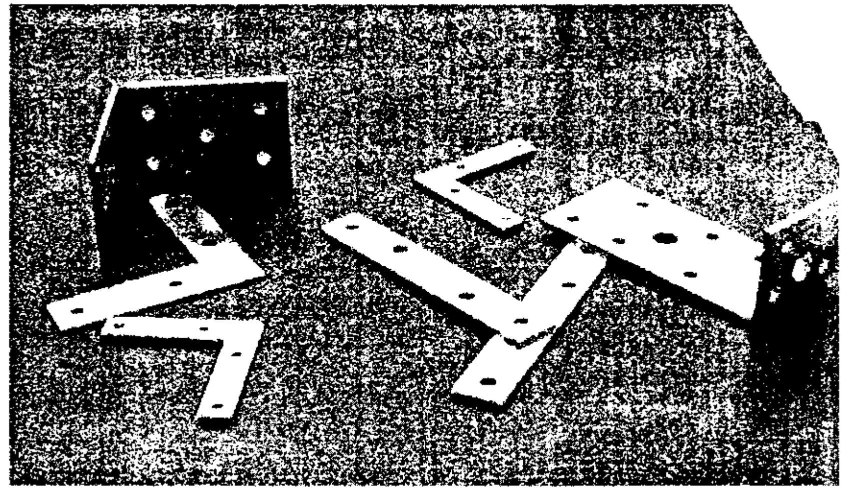


Image 61

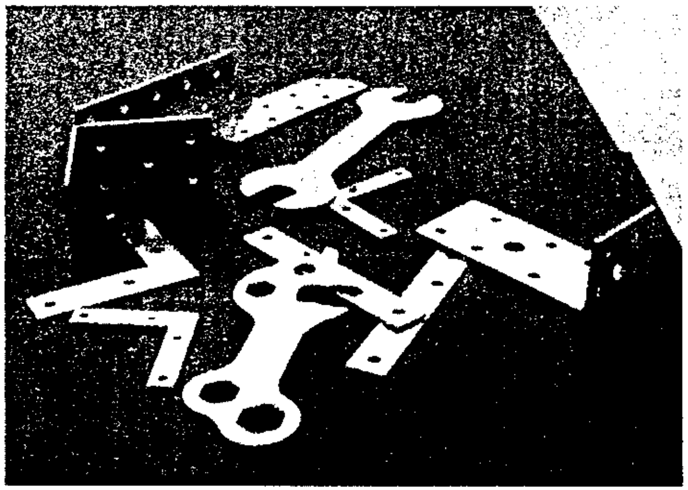


Image 62

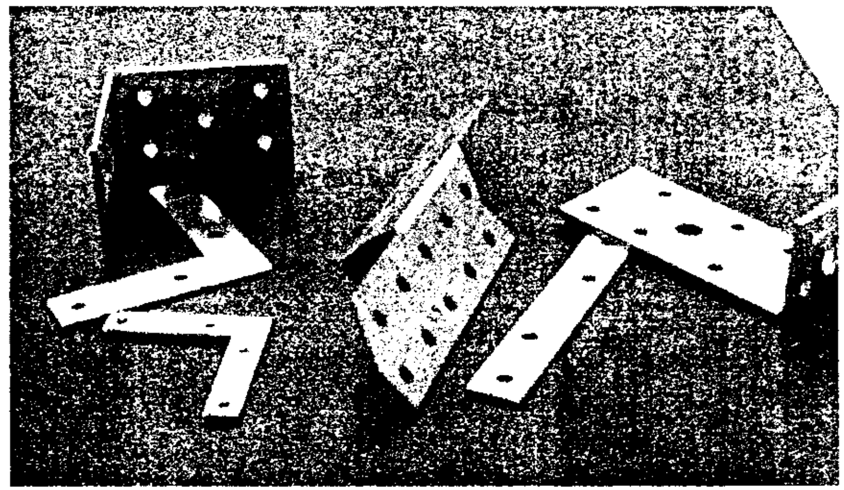


Image 63

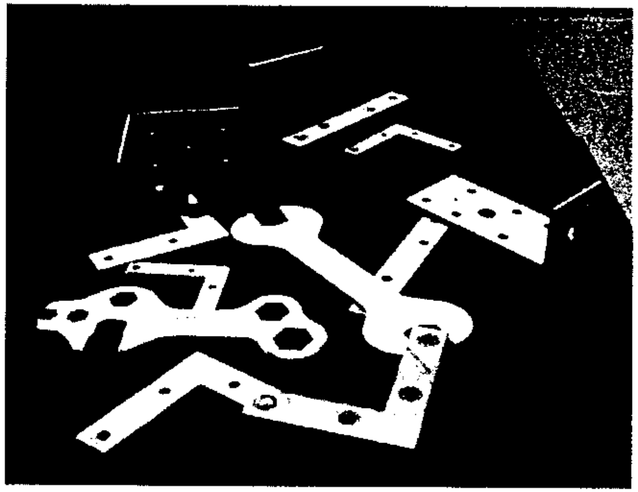


Image 64

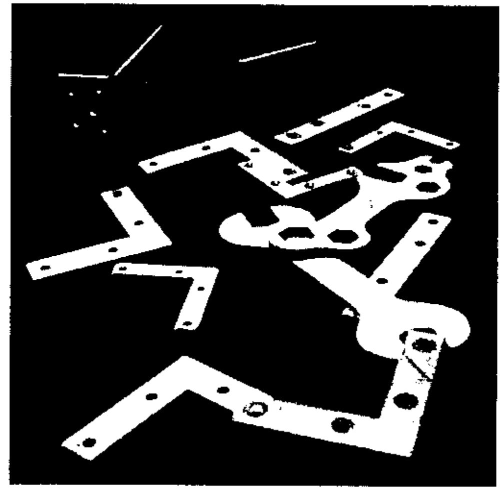


Image 65