School of Computer Science and Communication, KTH
Lecturer: Mårten Björkman

EXAM

Image Analysis and Computer Vision, DD2423

Friday, 16th of December 2011, 14.00–19.00

Allowed helping material: Calculator, the mathematics handbook Beta (or similar).

Language: The answers can be given either in English or Swedish.

General: The examination consists of Part A and Part B. For the passing grade E, you have to answer correctly at least 70% of Part A. If your score is less than 70%, the rest of the exam will not be corrected. Part B of the exam consists of **six** exercises that can give at most 50 points.

The bonus credits from the labs will be added to Part A if you do not reach 70% - otherwise they will be added to Part B.

The results will be announced within three weeks.

Part A

Provide short answers to the questions! Each answer is worth maximum one point.

1. What is the characteristic of a 'pinhole camera' and why is it important in computer vision?

   Answer: A pinhole camera is a camera that has an infinitesimally small hole for light to enter into a dark room, such that an image of the outer world is projected on a plane inside the room. The light coming into the hole is assumed to have the same direction as the light coming out of the hole. It serves as a model for cameras in computer vision, even if real cameras have lenses and the opening is not infinitesimally small.

2. If you are expressing perspective projections using homogeneous coordinates, why doesn't a scaling of the homogeneous coordinates matter?

   Answer: In the last stage of the projection, you divide all the coordinates (X,Y,W) *by the last coordinate* W*. Thus a rescaling by* k *of all coordinates cancels out, that is*

   $$\frac{(X,Y,W)}{W} = \frac{(kX,kY,kW)}{kW}.$$

3. If you look at an image of a city scene, how could you determine if an affine projection matrix would be reasonable to describe the image projection of the scene, instead of using a perspective projection matrix?

   Answer: The difference between the two projection matrices is that the last row of the affine projection matrix is $(0,0,0,1)$*. What this means in practise is that parallel lines in the real world will become paralell also in the image. So if most lines that are parallel in the world look parallel in the image, an affine projection matrix could probably be used. You can come to the same conclusion, if it is appers as if everthing in the image is located at approximately the same distance.*

4. What is a 'neighborhood system' and a 'connected component', and how are the two concepts related?

   Answer: A neighborhood defines when two pixels in an image are considered as neighbors depending on their relative positions. A connected component consists of all those points that are connected through paths, where all pairs along a path are neighboring pixels, given the particular neighborhood system.

5. Give an example of a non-linear filter that you have seen in the course. Why are linear filters often preferable from non-linear filters?

   Answer: Morphological operations such as opening, closing, dilation and erosion, as well as filter such as median, min and max filters, are examples of non-linear filters. Linear filters are preferable because they can be expressed as convolution and can be analysed in Fourier space. Linear filters can also be combined in any order.

6. What does it mean that a 2D kernel is separable? Is a Fourier Transform separable?

   Answer: It means that you can divide the application of the filter into two steps, first x-wise and then y-wise, speeding up the filtering. Fortunately, Fourier transforms are separable.

7. What happens to the Fourier domain representation of an image, if the image is translated?

   Answer: The magnitudes of the complex Fourier coefficients remain the same, but the phases (the relations between the real and imaginary parts) changes according to the translation.

8. Mention a segmentation method that doesn't take spatial coherence into consideration. Why is it often a bad idea to ignore spatial coherence for segmentation?

   Answer: Segmentation by histogramming and thresholding or by using K-means clustering are examples of methods that ignore spatial coherence. If you don't use spatial coherence the different segments might be fragmented into many different parts. You can force the segments to be connected, if you take advantage of spatial coherences using for example graph methods.

9. What kind of image features are preferable for matching in stereo or motion? Why are they preferable?

   Answer: Corner features are preferable, because the appearence of a corners changes as you move in any direction, unlike line features that look the same in the direction of the line. That makes corners more suitable for stereo and motion matching, where the exact positioning of the match is important. They are also relatively easy to compute and are (somewhat) invariant to for example view-point and illumination changes.

10. Describe shortly how one can create a multi-scale (scale-space) representation of an image. Why are such representations useful in computer vision?

    Answer: One example is a Gaussian pyramid. You take the original image and blur it multiple times and store each blurred version in the representation. If you blur enough you might subsample the images, without losing any information. Such a representation is convenient for finding image features at different scales. The first scale (from coarse to fine) where a feature appears, provides you information on the size (scale) of that features.

11. Given a set of edge points, a Hough Transform can be used to find straight lines. How is this done in practice? Shortly explain the steps involved.

    Answer: You first create a discrete array of values (accumulator space), each value corresponding to a particular set of parameter values. Then you go through all your edge points and for each edge point you place votes, in the accumulator space, to all those combinations of parameters that lead to a line passing through that edge point. Finally, you look at those accumulator values that have the highest number of votes. These give you the most likely lines in the image.

12. There are many kinds of shape descriptions, and they differ in different ways. Give two examples of aspects for which shape descriptions may differ.

    Answer Descriptors can vary in different ways. They can be either local or global, feature or pixel based, complete or incomplete, boundary or region-based, rigid or deformable, geometric or statistical.

13. Give two reasons why robust object recognition is so hard in practice.

    Answer: Object recognition is hard because the same object might look very different in the image, depending on the camera orientation, noise, occulusions, illumination, etc.

14. What is a 'vergence angle' and what is a 'gaze direction'?

    Answer: The vergence angle is the angle between the optical axes of two cameras placed in stereo and tells you how far away the fixation point is located. The gaze direction is the mean direction of the two axes with respect to the direction of the baseline.

15. What are the effects of morphological 'opening' and 'closing' operations?

    Answer: An opening operation first performs erosion on an image and then dilation. The net effect is that small foreground regions might disappears. The closing operation first befores dilation and the erosion, and will close small holes in foreground regions.

Part B

Exercise 1 (1+2+3+2=8 points)

Grey-level transformations is a convenient tool to enhance details in images, without doing any spatial filtering, by simply changing the grey-level values.

1. Assume that you apply a grey-level transformation $s = T(r)$ to an image. For which parts of the image do the contrasts increase, and for which do they decrease?

   Answer: The contrasts increase for those areas where the grey-level values satisfy $T'(r) > 1$. They decrease for areas where $T'(s) < 1$.

2. What is the goal of 'histogram equalization' and why would you be interested in applying it to an image?

   Answer: The goal is to create a new image where the histogram is as uniform as possible. In most cases it makes the image more balanced and more pleasing to a human. It might also help seeing details that were otherwise hidden, and improve the overall contrasts.

3. Assume you have an image with a histogram of grey-level values given by the distribution $p_R(r) = \frac{3}{5}(4r - 4r^2 + 1)$, $r \in [0,1]$. Determine a transformation $s = T(r)$, such that the histogram after the transformation becomes $p_S(s) = 1$, $s \in [0,1]$. For which values of r do you get a stretching of grey-level values, and for which do you get a compression?

   Answer: The transformation can be determined by computing the integral of $p_R(r)$, that is

   $$s = T(r) = \int_0^r p_R(x)dx = \frac{3}{5}[\ \frac{4}{2}x^2 - \frac{4}{3}x^3 + x\ ]_0^r = \frac{6}{5}r^2 - \frac{4}{5}r^3 + \frac{3}{5}r.$$

   Since $T(1) = 1$, which is a condition for $p_R(r)$ to be a distribution, we don't need to normalize the transformation. Its derivative, which is the same as $p_R(r)$, determines whether the new histogram is stretched or compressed.

   $$T'(r) = 1 \Rightarrow 4r - 4r^2 + 1 = \frac{5}{3} \Rightarrow 4r - 4r^2 - \frac{2}{3} = 0 \Rightarrow\ r^2 - r + \frac{1}{6} \Rightarrow$$

   $$\Rightarrow r_{high,low} = \frac{1}{2} \pm \sqrt{\frac{1}{2^2} - \frac{1}{6}} = \frac{1}{2} \pm \sqrt{\frac{3}{12} - \frac{2}{12}} = \frac{1}{2} \pm \sqrt{\frac{1}{12}}$$

   The grey-level values are stretched when $T'(r) > 1$, which is true for $0.211 \approx r_{low} < r < r_{high} \approx 0.789$ and compressed when $r < r_{low}$ or $r > r_{high}$.

4. How do you derive the transformation in the general case, when $p_S(s)$ is not necessarily a uniform transformation, such as in question 3? Give a short sketch of a solution, without doing any calculations.

   Answer: One way of doing this is to first determine two different transformation from $p_R(r)$ to a uniform distribution (leading to $T_r(r)$) and from $P_S(s)$ to another uniform distribution (leading to $T_s(s)$) and then combine them by taking the inverse of $T_s(s)$, that is $s = T(r) = T_s^{-1}(T_r(r))$. Inverting $T_S(s)$, however, is not always possible. Another way is to conclude that $P_S(s)ds = P_R(r)dr$ leads to $P_R(r)/P_S(s) = T'(r)$ and $T'(r)P_s(s) = P_R(r)$. If s is replaced by $T(r)$, you get a differential equation to solve.

Exercise 2 (2+3+2+2=9 points)

Most operations in image processing involve filtering in one way or the other. Depending on the image quality and your objective, it is essential to know what kind of filter to choose.

1. What are the properties of linear shift invariant filters? Mention at least two properties and explain what they mean.

   Answer: A linear filter L applied to a signals $f(x)$ and $g(x)$ obeys the rules of homogenity

   $$L(\alpha f(x)) = \alpha L(f(x))$$

   and additivity

   $$L(f(x)+g(x)) = L(f(x))+L(g(x)).$$

   If L is shift-invariant then it is also true that

   $$f(x) \to L(f(x)) \Rightarrow f(x-x_0) \to L(f(x-x_0)).$$

2. Assume you have a 1D image given by

   $$F(x) = \{1,6,8,11,7,3,1\}.$$

   Using a convolution, apply to $F(x)$ each of the following three filters:

   $$\begin{aligned} G_1(x) &= \{1,0,-1\}, \\ G_2(x) &= \{1,2,1\} \text{ and} \\ G_3(x) &= \{1,-2,1\}. \end{aligned}$$

   Answer with only five values per filter, disregarding the remaining undefined values.

   Answer: By changing the order of $G_k(x)$, letting it shift over $F(x)$ and summing up the pair-wise products of the three filter coefficients and the values of $F(x)$, you get

   $$F(x) * G_1(x) = \{7,5,-1,-8,-6\},$$

   $$F(x) * G_2(x) = \{21,33,37,28,14\}$$

   and

   $$F(x) * G_3(x) = \{-3,1,-7,0,2\}.$$

3. Which one of the three filters above respectively corresponds to a smoothing operation, an approximate 1st order derivative and a 2nd order derivative?

   Answer: $G_2(x)$ is a smoothing operation, $G_1(x)$ is an approximate 1st order derivative and $G_3(x)$ a 2nd order derivative.

4. For the filter you believe is a 2nd order derivative, compute its Fourier Transform. How well does this correspond to the frequency response of a real 2nd order derivative, that is

$$G(\omega) = -\omega^2?$$

Illustrate with a simple drawing, if it helps you.

Answer: If we take the Fourier transform of $G_3(x)$ (in continuous domain like $G(w)$), you get $G_3(\omega) = e^{-i\omega} - 2 + e^{i\omega} = 2cos(\omega) - 2$. This can be compared to $G(\omega)$ either by drawing or by using a Taylor series expansion. If we attempt the latter we get

$$G(0) = 2cos(0) - 2 = 0, \;\; G'(0) = -2sin(0) = 0,$$

$$G''(0) = -2cos(0) = -2, \;\; G^{(3)}(0) = 2sin(0) = 0 \;\; and \;\; G^{(4)} = 2cos(0) = 2.$$

Thus

$$G_3(\omega) = -\frac{2}{2!}\omega^2 + O(\omega^4) = -\omega^2 + O(\omega^4),$$

which is equal to $G(\omega)$ up to $O(\omega^4)$. Thus it is only at high frequencies that the differ considerably.

Exercise 3 (1+1+1+3+2=8 points)

In order to limit the amount of data to process and concentrate on the most relevant parts of an image, most computer vision methods rely on the extractions of features, such as edges, corners and regions.

1. Why is edge detection important, if you want to tell something about the 3D world?

   Answer: Since the 3D world consists of connected groups of points with discontinuities in-between, and these discontinuities appear as edges in images, it's a good idea to find such edges in order to study the 3D world. Furthermore, by studying edges in images, you concentrate your computational resources on what is most important, making the analysis as fast as possible.

2. What is an image gradient magnitude and how can you compute it?

   Answer: The image gradient is the rate of change in an image. For a 2D image L, the gradient at each pixel is a vector of two orthogonal derivatives, (L_x, L_y). The image gradient magnitude is the length of this vector $|(L_x, L_y)| = \sqrt{(L_x^2 + L_y^2)}$. Derivatives can be computed through convolusions of for example [1/2, 0, -1/2].

3. Why does edge detection normally include image smoothing as a first stage?

   Answer: Since edge detection measures the rate of change in images, a high-pass operation, it also enhances noise that typically dominates at high frequencies. If image smoothing is done prior to edge detection, the noise is suppressed and edge detection becomes less noise sensitive.

4. In Lab3 we found edges in images by asserting that two different conditions (on the image derivatives) have to be true for a pixel to be part of an edge. What are these two conditions and why do we need each of them?

   Answer: We were looking for edge points where the image derivative L_v in the gradient direction v reaches its maximum. In order to find such extremal points, we asserted that the derivative of the image gradient had to be zero, that is $L_{vv} = 0$, and in order to separate maxima from minima we further required that $L_{vvv} < 0$ had to be true. In the end we removed results that only appeared due to noise by requiring that L_v had to be less than some threshold.

5. Why would we often like to avoid using a 2nd order image derivative for edge detection? How does the Canny Edge detector solve the same problem, without using a 2nd order derivative?

   Answer: The 2nd order derivative is even more high-pass than the 1st order derivative, which makes it even more sensitive to noise. Thus one often tries to avoid it. The Canny Edge detector localizes the exact point where L_v reaches its maximum, not by computing L_{vv} and testing whether it is zero, but by performing Non-Maximum Suppression of L_v directly. By looking in the gradient direction v, points are found where L_v is locally maximum.

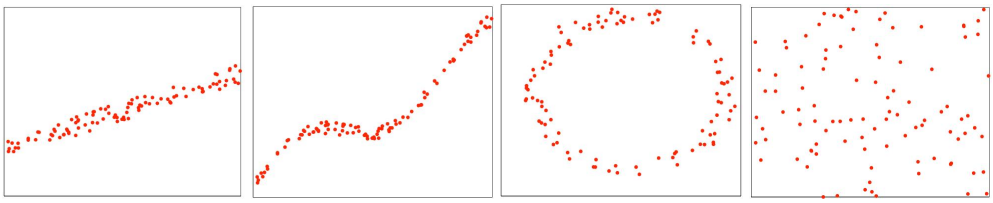
Exercise 4 (2+2+2+4=10 points)

Grouping of data (pixels or feature data) is essential for both image segmentation and classification. Even if the purpose varies, many of the methods used share similarities.

1. What are the similarities and differences between segmentation methods based on either 'K-means clustering' or 'Mean-shift'? Respond with at least one similarity and one difference.

   Answer: Both methods are iterative and aim at dividing a set of samples into a number of clusters. The way of doing this is different though. K-means assigns each point to a cluster center, if this center is the closest one among K centers, where K is a number given in advance. Mean-Shift is a density maximization method that can find arbitrarily many cluster centers. Points are assigned to a center based on which center the method will converge to starting from each point. The data of each point could be the same, but typically you would use colours and image positions for Mean-Shift, whereas K-means usually ignore positions (but doesn't have to).

2. Which of the four point clouds below apparently come from one-dimensional distributions? For which of these would Principal Component Analysis (PCA) work for dimensionality reduction? Why doesn't it work for the remaining case(s)?



a) b) c) d)

   Answer: The three first distributions must come from one-dimensional distributions, since the all could be approximated by some 1D curve in the 2D image. For PCA such an approximation will always be a straight line, which is feasible for the first two examples. For the remaining two cases a line is not a good approximation. Given two nearby points on a PCA projected line, one cannot tell whether the two original points are close, for the last two examples.

3. You are given a set of seven points placed on a 2D plane

$$\mathbf{p_1} = (8,6),\ \mathbf{p_2} = (1,5),\ \mathbf{p_3} = (5,5),\ \mathbf{p_4} = (0,4),\ \mathbf{p_5} = (7,3),\ \mathbf{p_6} = (5,3) \text{ and } \mathbf{p_7} = (2,2).$$

   Apply K-means clustering with $\mathbf{c_1} = (0,0)$ and $\mathbf{c_2} = (7,7)$ as the two initial cluster centers, and show how the division of points will be.

Answer: The squared distances to the two mean positions at the first iteration are respectively

	$\mathbf{p_1}$	$\mathbf{p_2}$	$\mathbf{p_3}$	$\mathbf{p_4}$	$\mathbf{p_5}$	$\mathbf{p_6}$	$\mathbf{p_7}$
$\mathbf{c_1}$	100	26	50	16	58	34	8
$\mathbf{c_2}$	2	40	8	56	16	20	50

Thus $\mathbf{p_2}$, $\mathbf{p_4}$ *and* $\mathbf{p_7}$ *will be assigned to* $\mathbf{c_1}$ *and* $\mathbf{p_1}$, $\mathbf{p_3}$, $\mathbf{p_5}$ *and* $\mathbf{p_6}$ *to* $\mathbf{c_2}$. *The new mean centers will be* $\mathbf{c_1} = (1+0+2,5+4+2)/3 = (3,11)/3 \approx (1.0,3.7)$ *and* $\mathbf{c_2} = (8+5+7+5,6+5+3+3)/4 = (25,17)/4 \approx (6.2,4.2)$. *It is then easily verified that there will be no change in the assignment in the second iteration. Thus it converges in only one iteration.*

4. Assume that the seven points $\mathbf{p_i}$ mentioned in question 3 actually come from a one-dimensional distribution, i.e. without noise they would be placed along a line. Find the line that best fits the seven points, if you apply Principal Component Analysis (PCA).

   Answer: In order to solve this problem you need to compute the (centered) covariance matrix. The center is computed as $\bar{\mathbf{p}}$ *=(8+1+5+0+7+5+2,6+5+5+4+3+3+2)/7 = (4,4), which leads to the centered coordinates (4,2), (-3,1), (1,1), (-4,0), (3,-1), (1,-1) and (-2,-2), and a covariance matrix*

$$C = \begin{pmatrix} C_{xx} & C_{xy} \\ C_{xy} & C_{yy} \end{pmatrix} = \frac{1}{7}\begin{pmatrix} 16+9+1+16+9+1+4 & 8-3+1+0-3-1+4 \\ 8-3+1+0-3-1+4 & 4+1+1+0+1+1+4 \end{pmatrix} = \frac{1}{7}\begin{pmatrix} 56 & 6 \\ 6 & 12 \end{pmatrix}$$

   Note that we can safely drop the factor $1/7$ *for the remaining computations. The (largest) eigenvalue is given by the solution of*

$$\begin{vmatrix} \lambda - 56 & -6 \\ -6 & \lambda - 12 \end{vmatrix} = \lambda^2 - 68\lambda + 636 = 0. \Rightarrow \lambda = 34 + \sqrt{34^2 - 636} = 34 + \sqrt{520}$$

   The direction (e_x, e_y) *of the line, which is the eigenvector corresponding to the eigenvalue* λ, *is thus given by* $(34+\sqrt{520}-56)e_x - 6e_y = 0 \Rightarrow (e_x, e_y) = (6, \sqrt{520}-22)$. *The point* $\bar{\mathbf{p}} = (4,4)$ *must be a point on this line. Thus a line equation can be written as* $(\sqrt{520}-22)x - 6y = (\sqrt{520}-22)\cdot 4 - 6\cdot 4$ *or* $0.804x - 6y + 20.786 = 0$.

Exercise 5 (2+2+4=8 points)

Instead of using pixel data directly, object recognition is usually performed using some low-dimensional representation of images. Part of the problem is solved by finding the best possible such representation.

1. If you have a set of images of some object classes, how can you tell whether a particular representation would be good for recognition of these classes?

   Answer: First of all, points within the same class has to be close to eachothers in the representation, compared to the distance between points of different classes. Secondly, it has to be possible to measure distances, that is a metric has to exist, for which this is true. There could be other reasons why a representation is preferable, such as whether it allows necessary invariances.

2. Explain shortly how 'nearest neighbour' classification works. What makes nearest neighbour different from statistical classification methods?

   Answer: Unlike statistical classification, nearest neighbour classification doesn't have any statistical model of the data. It is represented by the training points themselves. For a new test point, the nearest neighbour is searched among all training samples. The result of the classification is based on the class identity of this closest neighbour.

3. We want to find a classifier to separate two classes, class A and class B. The classes are both assumed to have normal distributions, i.e. the probability density functions are of the form

$$p(z \mid k) = \frac{1}{\sqrt{2\pi\sigma_k^2}}\, e^{-(z-m_k)^2/(2\sigma_k^2)},$$

where z is a one-dimensional measurement and k a class. The classes are further assumed to have means $m_A = -4$ and $m_B = 2$, and variances $\sigma_A^2 = 18$ and $\sigma_B^2 = 2$. Estimate the optimal classification boundaries and decision rules, if the prior probabilities are $p(k = A) = 2/3$ and $p(k = B) = 1/3$.

Answer: The classification is based on the most probably class k given the data z by maximizing $P(k|z) = K^{-1}P(z|k)P(k)$*, where* $P(z|k)$ *is the distribution of class k and* $P(k)$ *the prior distribution. Here* $K = P(z)$ *is a constant that can be ignored. The decision boundary (in the case of classes A and B) is given by* $P(z|A)P(A) = P(z|B)P(B)$*. If* $P(z|A)P(A) > P(z|B)P(B)$*, then z is classified as belonging to class A.*

$$P(z|A)P(A) > P(z|B)P(B) \Rightarrow \frac{2}{3\sqrt{2\pi 18}}\, e^{-(z+4)^2/(2\cdot 18)} > \frac{1}{3\sqrt{2\pi 2}}\, e^{-(z-2)^2/(2\cdot 2)} \Rightarrow$$

$$[\text{ multiply by } 6\sqrt{\pi}\,] \Rightarrow (2/3)\, e^{-(z+4)^2/36} > e^{-(z-2)^2/4} \Rightarrow$$

$$[\text{ take the logarithm and multiply by } 36\,] \Rightarrow 36\log(2/3) - (z+4)^2 > -9(z-2)^2 \Rightarrow$$

$$[\text{ simplify }] \Rightarrow 36\log(2/3) - (z+4)^2 + 9(z-2)^2 = 8z^2 - 44z + 36\log(2/3) + 20 > 0$$

$$z^2 - 5.5z + 4.5\log(2/3) + 2.5 = 0 \Rightarrow z_{h,l} = 0.25(11 \pm \sqrt{11^2 + 72\log(3/2) - 40}) \Rightarrow$$

$$z_h = 5.37,\ z_l = 0.126$$

Thus if $z > 5.37$ *or* $z < 0.126$*, z is classified as belonging to class A, otherwise class B.*

Exercise 6 (3+2+2=7 points)

Stereo is useful in order to determine the distance to objects seen in an image. However, even in cases with only one camera, distances can often be computed, if you add some assumptions.

1. Assume you are moving towards a person seen in an image. At time t_0 the height of the person in the image is $h_0 = 200$ pixels and at time t_1 it is $h_1 = 240$ pixels. If you have moved 1.00 meter between t_0 and t_1, what was the distance Z_0 to the person at time t_0? If we assume the person is 1.80 meters in height, what does the focal length f of the camera have to be (measured in pixels)?

   Answer: The projective size of a person of height H at t_0 *is given by* $h_0 = fH/Z_0$*, where f is the focal length (measured in pixels) and* Z_0 *is the distance to the person. If the you have moved* $1m$ *closer to the person at* t_1*, the size is* $h_1 = fH/(Z_0 - 1)$*. Thus* $fH = h_0Z_0 = h_1(Z_0 - 1)$*, which in our case means* $200Z_0 = 240(Z_0 - 1) \Rightarrow 40Z_0 = 240m \Rightarrow Z_0 = 6m$*. If the person is* $H = 1.80m$ *in height it means that the focal length* $f = h_0Z_0/H = 200 \cdot 6/1.8\ pix = 1200/1.8\ pix \approx 667\ pix$*.*

2. Assume that you have another camera placed in parallel with and next to the first camera in the question 1 and the distance between the two cameras is $b = 0.10$ meters. If both these cameras have the same focal length f, what is the stereo disparity at the region corresponding to the person in the image at time t_0?

   Answer: The stereo disparity is given by $d = bf/Z_0$*, where b is the baseline between the cameras and f is the focal length. In our case it means* $d = 0.1(1200/1.8)/6\ pix = 20/1.8\ pix \approx 11.1\ pix$*.*

3. What is an epipolar plane and an epipolar line? How do these concepts facilitate stereo matching? Make a drawing, if it helps you explain.

   Answer: Three points are enough to define a 2D plane in a 3D space. An epipolar plane is a plane given by a projective point p_1 in one image and the optical centers of the left and right cameras. This plane will intersect the image plane of the opposing camera in a line, which is called an epipolar line. This facilitates stereo matching, since the point p_2 that corresponds to p_1 is known to be located along the epipolar line defined by p_1 and the two camera centers.