Issues in Robot Vision

Goesta H. Granlund
Computer Vision Laboratory,
Linköping University,
581 83 Linköping, Sweden

Abstract
In this paper we will discuss certain issues regarding robot vision. We will deal with aspects of pre-attentive versus attentive vision, control mechanisms for low level focus of attention, representation of motion as the orientation of hyperplanes in multidimensional time-space. Issues of scale will be touched upon.

1 Introduction
In this paper we will discuss certain issues regarding robot vision. Due to the limited format, it will by no means be possible to give any comprehensive overview of the field. Most aspects will in fact have to be omitted, and it is unfortunately not possible to give due references to all significant contributions within the area.

Machine vision has developed over the years, and different methodologies have been emphasized as crucial at different times. Classically, the methodology of image analysis contains many procedures to perform various tasks [16, 3, 23]. A common problem is that these procedures are often not suitable as components of a larger system, where procedures interact. One reason is that information is represented in different ways for different types of features. It is difficult to have such descriptors cooperate and to control each other in a structured, parametrical way.

An important feature of current state of the art is the view that sufficiently efficient interpretation of complex scenes can only be implemented using an adaptive model structure. In the infancy of computer vision, it was believed that objects of interest could unequivocally be separated from the background using a few standard operations applied over the entire image. It turns out, however, that this simple methodology only works on simple images having a good separation between object and background. In the case of more difficult problems with noise, ambiguities and disturbances of different types, more sophisticated algorithms are required with provisions to adapt themselves to the image content. A further extension of this adaptivity is the current development of Active Vision [46].

It consequently turns out to be necessary to use what we may call different sub-algorithms on different parts of an image. The selection of a particular sub-algorithm is often based upon a tentative analysis of the image content. The reason for using different sub-algorithms is the simple fact that all possible events can not be expected in a particular context. In order for this handling of sub-algorithms to be manageable, it has to be implemented as a parameterization of more general algorithms.

In some of the work cited, as well as in our own work, there has been taken a great deal of impression from what is known about biological visual systems [24, 25, 35]. This is not to say that we assume that the structures presented are indeed models used in biological visual systems. Too little is so far known to
form any firm opinions on the structures used. The ultimate criterion is simply performance from a technical point of view.

2 Pre-attentive Versus Attentive Vision

Classically, most vision procedures have been applied uniformly over an image or a scene. Such an indiscriminate application of computation power is very costly, and as complexity in the desired processing is increasing, it becomes necessary to find methods to restrict the attention to regions of maximal importance.

Humans can shift the attention either by moving the fixation point or by concentrating on a part of the field of view. The two types are called overt and covert attention respectively. The covert attention shifts are about four times as fast as the overt shifts. This speed difference can be used to check a potential fixation point to see if it is worthwhile moving the gaze to that position.

A number of paradigms describing human focus of attention have been developed over the years [36]. We will here mainly discuss the search light metaphor [26]. A basic assumption of this metaphor is the division between preattentive and attentive perception. The idea is that the preattentive part of the system makes a crude analysis of the field of view. The attentive part then analyzes areas of particular interest more closely. The two systems should not be seen as taking turns in a time multiplex manner, but rather as a pipeline where the attentive part uses the continuous stream of results from the preattentive part as clues. The reason for having to focus the attention in this metaphor is that certain tasks are of inherently sequential nature, rather than amenable to a parallel processing.

What features or properties are important for positioning the fixation point? Yarbus pioneered the work on studying how humans move the fixation point in images depending on the wanted information [51]. For preattentional shifts, gradients in space and time, i.e. high contrast areas or motion, are considered to be the important features. Abbott and Ahuja present a list of criteria for the choice of the next fixation point [1]. Many of the items in the list relate to computational considerations. A few clues from human visual behavior were also included, of which the following is a sample:

Absolute distance and direction If multiple candidates for fixation points are present, the ones closer to the center of the viewing field are more likely to be chosen. Upward movement is generally preferred over downward movement.

2D image characteristics If polygonal objects are presented, points close to corners are likely to be chosen as fixation points. When symmetries are present, the fixation point tends to be chosen along symmetry lines.

Temporal changes When a peripheral stimulus suddenly appears, a strong temporal cue often leads to a movement of the fixation point towards the stimulus.

Since fixation point control is a highly task dependent action, it is probably easy to construct situations that contradict the list above. The reader is urged to go back to the appropriate references in order to get a full description of how the results where obtained.
2.1 Focus of attention in machine vision

A number of research groups are currently working on incorporating focus of attention mechanisms in computer vision algorithms. This section is by no means a comprehensive overview, but rather a few interesting examples.

Ballard and Brown have produced a series of experiments with ocular reflexes and visual skills [4, 9, 11, 10, 5]. The basic idea is to use simple and fast image processing algorithms in combination with a flexible, active perception system.

A focus of attention system based on salient features has been developed by Milanese [37]. A number of features are extracted from the input image and are represented in a set of feature maps. Features differing from their surroundings are moved to a corresponding set of conspicuity maps. These maps consist of interesting regions of each feature. The conspicuity maps are then merged into a central saliency map where the attention system generates a sequence of attention shifts based on the activity in the map.

Brunnström, Eklund and Lindeberg have presented an active vision approach to classifying corner points in order to examine the structure of the scene. Interesting areas are detected and potential corner points scrutinized by zooming in on them [12]. The possibility of actively choosing the imaging parameters, e.g. point of view and focal length, allows the classification algorithm to be much simpler than for static images or pre-recorded sequences.

A variation of the search light metaphor, called the attentional beam has been developed by Tsotsos and Culhane [14, 43, 44]. It is based on a hierarchical information representation where a search light on the top is passed downwards in the hierarchy to all processing units that contribute to the attended unit. Neighboring units are inhibited. The information in the ‘beamed’ part of the hierarchy is reprocessed, without the interference from the neighbors, the beam is then used to inhibit the processing elements and a new beam is chosen.

The ESPRIT Basic Research Action project 3038, Vision as Process [46], is designed to study the scientific hypothesis that vision should be handled as a continuous process. The project is aimed at bringing together knowhow from a wide variety of research fields ranging from low level feature extraction and ocular reflexes through object recognition and task planning.

Westelius, Knutsson and Granlund have developed a hierarchical gaze control structure for use with multi-resolution image sensors [47, 48].

2.2 Variable resolution sensors

The human eye has its highest resolution at the center of the optical axis, and it decays towards the periphery. There are a number of advantages in such an arrangement. To mention a few:

- Data reduction compared to having the whole field of view in full resolution.
- High resolution is combined with a broad field of view.
- The fovea marks the area of interest, and disturbing details in the surround are blurred.

These advantages can be utilized in a robot vision system as well. There are a number of research projects developing both hardware and algorithms for heterogeneous sampled image arrays, implementing the fovea concept in one form or another, e.g. [42].
2.3 Control mechanism components

Having full resolution only in the center part of the visual field makes it obvious
that a good algorithm for positioning the fixation point is necessary. A number
of focus-of-attention control mechanisms must be active simultaneously to be able
to both handle unexpected events and perform an effective search. The different
components can roughly be divided into the following groups:

1. Preattentive, data driven control. Non-predicted structured image informa-
tion and events attract the focus-of-attention in order to get the information
analyzed.

2. Attentive, model driven control. The focus-of-attention is directed toward
an interesting region according to predictions using already acquired image
information and knowledge from models.

3. Habituation. As image structures are analyzed and modeled their impact on
preattentive gaze control is reduced.

The distinction between the preattentive and attentive parts is floating. It
is more of a spectrum from pure reflexes to pure attentional movements of the
fixation point.

2.4 Gaze control

We will discuss an example of a simple control system with three levels:

Camera Vergence. Cameras are verged towards the same fixation point using
the disparity estimates from a stereo algorithm.

Edge tracker. Magnitude and phase from quadrature filters form a vector field
drawing the attention towards and along lines and edges in the image [31, 47].

Object finder. Symmetry properties in the orientation estimates are used to
indicate potential objects [8, 21].

The refinement of the positioning of the fixation point is handled with potential
fields in the robots parameter space. It can be visualized as an 'energy landscape'
where the trajectory is the path a little ball freely rolling around would take.
The fixation point can be moved to a certain position by forming a potential well
around the position in the parameter space corresponding to the robot looking
in that direction. The potential fields from the different controlling modules are
weighted together to get the total behavior.

2.4.1 Model acquisition and memory

The system marks the states in its parameter space that corresponds to the direc-
tion in which it has been tracking edges. This is the first step towards a memory
of where it has looked before, and components of a model of its environment. In
a general system, where many points in the parameter space might correspond
to looking at the same thing, a more sophisticated handling of model properties
is required. It is then important to remember and build up a model of not only
WHERE but also WHAT the system has seen. For non-static scenes, WHEN
becomes important. This leads to a procedure for model acquisition which is an
ultimate goal for this process.
3 Image Measurements and Representation

In order for a system modeling a high structural complexity to be manageable and extendable, it is necessary that it exhibits modularity in various respects. This implies for example standardized information representations for interaction between operator modules. Otherwise, the complexity will be overwhelming and functional mechanisms will be completely obscure. One way to satisfy these requirements is to implement the model structure in a hierarchical, fragmented fashion. In order for such a structure to work efficiently, however, certain requirements have to be fulfilled for information representation and for operations.

It is apparent that there are two issues related to hierarchies and pyramid structures. One has to do with level of abstraction, and the other with size or scale. Although they are conceptually different, there are certain relations. With increased level of abstraction generally follows an increase of the scale over which we relate phenomena [39].

Hierarchical structures is nothing new in information processing in general, or in computer vision in particular. A regular organization of algorithms has always been a desired goal for computer scientists.

Among the first structured approaches were those motivated by knowledge about biological visual systems. The perceptron approach by Rosenblatt [40], has attracted new attention as neural network theory has become a hot research topic [22]. The work on layered networks continued, where such networks would accept image data at their bottom level [45, 41, 19].

The Fourier transform has found considerable use in signal analysis. In image analysis, however, the global Fourier transform representation gives rise to problems due to the loss of spatial localization in the transform domain. The Short Time Fourier Transform, or windowed Fourier transform, is one way to modify the Fourier transform for better performance on non-stationary signals. The widely chosen windowing function is the Gabor function due to its simultaneous concentration in both domains [17]. Gabor and wavelet transforms have proved to be very useful.

Most of the work so far has dealt with hierarchies relating to size or scale, although they have indirectly given structural properties. Granlund introduced an explicit abstraction hierarchy [18], employing symmetry properties implemented by Gaussian wavelets in what today is commonly referred to as Gabor functions [17].

Burt introduced an approach to hierarchical image decomposition using the Laplacian or DOLP (Difference Of Low Pass) pyramid [13]. In this way an image is transformed into a set of descriptor elements. The image can then be reconstructed from its set of primitives.

The concept of scale or size as a dimension, was further extended in the so called scale space representation of images [50, 33, 34].

3.1 Representation of Motion as Orientation in 3-D

Motion of a point in 2-D can be viewed as a line in 3-D time-space. Correspondingly, the motion of a line in 2-D can be viewed as a plane in 3-D time-space. There are however some complications, not only due to the increased volume of data, but also from a more fundamental point of view. In two dimensions, the orientation of a line or an edge can unambiguously be represented by a vector in a “double angle” representation [18]. The mapping requirements of operations in multiple
dimensions are more severe than for two dimensions [32]. With a hemisphere as
the original space, an equivalent of the complication encountered in 2-D occurs:
Surfaces that differ by a small angle can end up being represented by vectors that
are very different, i.e. close to opposite sides of the rims of the hemispheres. This
is of course unacceptable if the metric properties of the space are of any conse-
quence, which will always be the case if there is a next stage where the information
is to be further processed. Consider e.g. the case of differentiation when the vector
passes in a step-like fashion from one side of the hemisphere to the other. It is
necessary therefore, that a mapping is established that “closes” the space in the
same manner as earlier discussed for the two-dimensional case.

It turns out that information can for this purpose advantageously be repre-
sented by tensors [27]. The tensor representation can be used for filtering in
volumes and in time sequences, implementing spatio-temporal filters [49]. It can
also be used for computation of higher level features such as curvature [6] or ac-
celeration. The tensor mapping can be controlled by transforms to implement
adaptive filtering of volume data or time sequences [29].

The tensor representation of the local orientation of a neighbourhood with one
single orientation in 3-D, is given by

\[
\mathbf{T} = \frac{1}{x}
\begin{pmatrix}
\mathbf{x}_1^2 & \mathbf{x}_1 \mathbf{x}_2 & \mathbf{x}_1 \mathbf{x}_3 \\
\mathbf{x}_1 \mathbf{x}_2 & \mathbf{x}_2^2 & \mathbf{x}_2 \mathbf{x}_3 \\
\mathbf{x}_1 \mathbf{x}_3 & \mathbf{x}_2 \mathbf{x}_3 & \mathbf{x}_3^2
\end{pmatrix}
\] (1)

where \( \mathbf{x} = (x_1, x_2, x_3) \) is a normal vector to the plane of the neighbourhood and
\( x = \sqrt{x_1^2 + x_2^2 + x_3^2} \). The magnitude of \( \mathbf{x} \) is determined by the local energy distribution
estimated by filters.

The orientation estimation requires a number of precomputed quadrature fil-
tersevenly spread in one half of the Fourier space [27, 28]. The minimum number
of quadrature filters required for orientation estimation in 3-D is 6, where the
filters are directed as the vertices of a semiicosahedron, see Figure 1:

\[
\begin{align*}
\hat{\mathbf{n}}_1 &= c (a, 0, b)^t \\
\hat{\mathbf{n}}_2 &= c (-a, 0, b)^t \\
\hat{\mathbf{n}}_3 &= c (b, a, 0)^t \\
\hat{\mathbf{n}}_4 &= c (b, -a, 0)^t \\
\hat{\mathbf{n}}_5 &= c (0, b, a)^t \\
\hat{\mathbf{n}}_6 &= c (0, b, -a)^t
\end{align*}
\] (2)

with

\[
\begin{align*}
a &= 2 \\
b &= 1 + \sqrt{5} \\
c &= (10 + 2\sqrt{5})^{-1/2}
\end{align*}
\] (3)

A quadrature filter designed with a lognormal function, is given in the frequency
domain by:

\[
\begin{cases}
F_k(\omega) = F_0(\omega) (\omega \cdot \hat{\mathbf{n}}_k)^2 & \text{if } \omega \cdot \hat{\mathbf{n}}_k > 0 \\
F_k(\omega) = 0 & \text{otherwise}
\end{cases}
\] (4)

The spatial filter coefficients are found by a straightforward 3-D-DFT or by use
of an optimization technique. The resulting spatial filter is complex-valued. This
procedure is used to obtain the six quadrature filters.
Figure 1: Orientation in 3-D space of filter symmetry axes, in the form of an icosahedron.

It is easy to implement the orientation algorithm with these precomputed filters [27]. The tensor describing the neighbourhood is given by:

\[ T^e = \sum_k q_k (N_k - \frac{1}{5} I) \]  

(5)

where \( q_k \) again denotes the magnitude of the output from filter \( k \). \( N_k = \hat{n}_k \hat{n}_k^T \) denotes the direction of the filter expressed in the tensor representation and \( I \) is the unity tensor. A less compact description of Eq. 5 is:

1. Convolve the input data with the six complex-valued filters, i.e. perform twelve scalar convolutions.

2. Compute the magnitude of each complex-valued filter by

\[
q_k = \sqrt{q_{k\epsilon}^2 + q_{ko}^2}
\]

where \( q_{k\epsilon} \) denotes the filter output of the real part of filter \( k \) and \( q_{ko} \) denotes the filter output of the imaginary part of filter \( k \).

3. Compute the tensor \( T^e \) by Eq. 5, i.e.

\[
T^e = \begin{pmatrix}
T_{11} & T_{12} & T_{13} \\
T_{12} & T_{22} & T_{23} \\
T_{13} & T_{23} & T_{33}
\end{pmatrix}
\]

where

\[
T_{11} = A(q_1 + q_2) + B(q_3 + q_4) - S \\
T_{22} = A(q_3 + q_4) + B(q_5 + q_6) - S \\
T_{33} = A(q_5 + q_6) + B(q_1 + q_2) - S \\
T_{12} = C(q_3 - q_4) \\
T_{13} = C(q_1 - q_2) \\
T_{23} = C(q_5 - q_6)
\]
3.1.1 Evaluation of the Representation Tensor

It is shown in [27] that the eigenvector corresponding to the largest eigenvalue of $T^e$ is the normal vector of the plane best describing the neighborhood. This implies that an eigenvalue analysis is appropriate for evaluating the tensor. Below the eigenvalue distribution and the corresponding tensor representation are given for three particular cases of $T^e$, where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ are the eigenvalues in decreasing order, and $\hat{e}_i$ is the eigenvector corresponding to $\lambda_i$.

1. $\lambda_1 > 0$; $\lambda_2 = \lambda_3 = 0$;
   $$T^e = \lambda_1 \hat{e}_1 \hat{e}_1^t$$
   This case corresponds to a neighborhood that is perfectly planar, i.e. is constant on planes in a given orientation. The orientation of the normal vectors to the planes is given by $\hat{e}_1$.

2. $\lambda_1 = \lambda_2 > 0$; $\lambda_3 = 0$;
   $$T^e = \lambda_1 \left( I - \hat{e}_3 \hat{e}_3^t \right)$$
   This case corresponds to a neighborhood that is constant on lines. The orientation of the lines is given by the eigenvector corresponding to the least eigenvalue, $\hat{e}_3$.

3. $\lambda_1 = \lambda_2 = \lambda_3 > 0$;
   $$T^e = \lambda_1 I$$
   This case corresponds to an isotropic neighborhood, meaning that there exists energy in the neighborhood but no orientation, e.g. in the case of noise.

The eigenvalues and eigenvectors are easily computed with standard methods such as the Jacobi method, e.g. [38]. Note that the spectral decomposition theorem states that all neighborhoods can be expressed as a linear combination of these three cases.

3.1.2 Velocity Estimation

If the signal to analyze is a time sequence, a plane implies a moving line and a line implies a moving point. The optical flow will be obtained by an eigenvalue analysis of the estimated representation tensor. The projection of the eigenvector corresponding to the largest eigenvalue onto the image plane will give the flow field. However, the so-called aperture problem will give rise to an unspecified
velocity component, the component moving along the line. The aperture problem is a problem in all optical flow algorithms which rely on local operators. On the other hand, the aperture problem does not exist for moving points in the sequence. In this case of velocity estimation the correspondence between the energy in the spatial dimensions and the time dimension is established to get correct velocity estimation.

By examining the relations between the eigenvalues in the orientation tensor it is possible to divide the optical flow estimation into different categories, [7, 20]. Depending on the category, different strategies can be chosen, see Section 3.1.1. Case number two in Section 3.1.1, i.e. the line case, gives a correct estimation of the velocity in the image plane and is thus very important in the understanding of the motion.

To do this division of different shapes of the tensor the following functions are chosen:

\[
P_{\text{plane}} = \frac{\lambda_1 - \lambda_2}{\lambda_1} \quad (6)
\]

\[
P_{\text{line}} = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad (7)
\]

\[
P_{\text{iso}} = \frac{\lambda_3}{\lambda_1} \quad (8)
\]

These expressions can be seen as the probability for each case. The discrimination is made by selecting the case having the highest probability.

The calculation of the optical flow is done using Eq. 9 for the plane case and Eq. 10 for the line case. In neighborhoods classified as 'isotropic' no optical flow is computed. The 'true' optical flow in neighborhoods of the 'plane' type, such as moving lines, cannot be computed by optical flow algorithms using only local neighbourhood operations as mentioned earlier. The optical flow is computed by

\[
x = \hat{e}_1
\]

\[
v_{\text{line}} = (-x_1 x_3 \hat{x}_1 - x_2 x_3 \hat{x}_2)/(x_1^2 + x_2^2) \quad (9)
\]

where \(\hat{x}_1\) and \(\hat{x}_2\) are the orthogonal unit vectors defining the image plane.

The aperture problem does not exist for neighborhoods of the 'line' type, such as moving points. This makes them, as mentioned, very important for motion analysis. The optical flow is computed by

\[
x = \hat{e}_3
\]

\[
v_{\text{point}} = (x_1 \hat{x}_1 + x_2 \hat{x}_2)/x_3 \quad (10)
\]

The use of certainty measures is one of the central mechanisms in the hierarchical framework and the optical flow is not used directly. Separate estimates of the direction of movement and velocity are accompanied with certainty measures computed by combining the tensor norm and the appropriate discriminant function. It is possible to use the confidence statement to process incomplete or uncertain data, as well as data emerging from spatial or temporal transients [30].

4 Spatio-Temporal Channels

The human visual system has difficulties handling high spatial frequencies simultaneously with high temporal frequencies [2, 15]. This means that objects with high
velocity cannot be seen sharply without tracking. One aspect of this is that the visual system performs an effective data reduction. The data reduction is made in such a way that high spatial frequencies can be handled if the temporal frequency is low, and vice versa. This strategy is possible to use in a computer vision model for time sequences.

An input image sequence is subsampled both spatially and temporally into different channels. In Table 1 the data content in the different channels relatively to a reference sequence, ch00, is shown. For a typical example. The reference sequence has maximum resolution in all dimensions; typically this means a video signal of 50 Hz, height 576 and width 720 pixels. The frequency difference between adjacent channels is one octave, i.e. a subsampling factor of 2 is used.

The numbers in Table 1 indicate that a large data reduction can be made by not using the channels with high resolution in both spatial and temporal domains. For instance, the channels on the diagonal together contain approximately 1/4 of the data in the reference sequence (ch00). There is a signal theoretical reason to use a pyramid representation of the image. A single filter has a particular limited pass band, both temporally and spatially, which may or may not be tuned to the different features to describe. In Figure 2a the upper cut-off frequency for a spatio-temporal quadrature filter set is indicated. The lower cut-off frequency is not plotted for the sake of clarity. Only the first quadrant in the $\omega_s, \omega_t$ plane is plotted. The use of this filter set on different subsampled channels corresponds to using filters with different center frequencies and constant relative bandwidth. Figure 2b indicates the upper cut-off frequency when convolving the channels on the diagonal in Table 1 with this filter set.

To avoid aliasing in the subsampling, the sequence must be prefiltered with a lowpass filter. As the resulting channel shall be further processed, the design of the lowpass filter is critical. The estimation of optical flow from Eq. 9 and Eq. 10 utilizes the relationship of energies originating from spatial variations and from temporal variations. The lowpass filter used for anti-aliasing should then not influence this relationship.

Table 1: Data content and name convention for the different spatio-temporal channels.

<table>
<thead>
<tr>
<th>Spatial subsampling</th>
<th>1/8</th>
<th>1/4</th>
<th>1/2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/64</td>
<td>1/16</td>
<td>1/4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Relative data content</td>
<td>1/128</td>
<td>1/32</td>
<td>1/8</td>
<td>1/2</td>
</tr>
<tr>
<td>1/256</td>
<td>1/64</td>
<td>1/16</td>
<td>1/4</td>
<td>1/4</td>
</tr>
<tr>
<td>1/512</td>
<td>1/128</td>
<td>1/32</td>
<td>1/8</td>
<td>1/8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Notation for sequence</th>
<th>ch30</th>
<th>ch20</th>
<th>ch10</th>
<th>ch00</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch31</td>
<td>ch21</td>
<td>ch11</td>
<td>ch01</td>
<td></td>
</tr>
<tr>
<td>ch32</td>
<td>ch22</td>
<td>ch12</td>
<td>ch02</td>
<td></td>
</tr>
<tr>
<td>ch33</td>
<td>ch23</td>
<td>ch13</td>
<td>ch03</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Spatial subsampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/8</td>
</tr>
<tr>
<td>Notation for sequence</td>
</tr>
<tr>
<td>ch31</td>
</tr>
<tr>
<td>ch32</td>
</tr>
<tr>
<td>ch33</td>
</tr>
</tbody>
</table>

1/2 Temporal subsampling

Figure 2: Cut-off frequency for a spatio-temporal filter.

5 Acknowledgements

The author wants to acknowledge the financial support of the Swedish National Board of Technical Development, which supported a great deal of the local research and documentation work mentioned in this overview. Considerable credit should be given to the staff of the Computer Vision Laboratory of Linköping University, for discussion of the contents as well as for text and figure contributions to different parts of the manuscript.

References


