Chapter 2

Automated carbon masking and particle picking in data preparation for single particles


Abstract

Two new algorithms, automated carbon masking and quaternion based rotation space sampling for automated particle picking, are presented here. They are implemented as plug-ins in the Cyclops software suite and are intended for data preparation for 3D single particle reconstruction. Cyclops is a new computer program designed as a graphical front-end that allows easy control and interaction with tasks and programs for 3D reconstruction.

Automating a particle search needs an algorithm that finds out where in the image the search has to be done. Normally only the particles in the holes (circular or irregular) of the carbon layer are of use. Currently no other automatic carbon masking algorithm for EM image processing exists. Traditional edge detection and segmentation algorithms do not work due to the extremely high noise in cryo-EM images. The new masking algorithm is based on the relatively high variance within carbon regions and gives good results.

A quaternion is a 4D number that can be used to represent and manipulate rotations in 3D space. The uniform sampling of rotations in 2D space is straightforward, but for rotations in 3D space, uniform sampling is more problematic. With the help of quaternion theory, we implemented an algorithm for uniform sampling in 3D rotation space that is based on subdivision of the regular polytopes in 4 dimensions. The algorithm can be used in single particle picking and alignment using a set of projection
classes from a known or inferred low resolution 3D model.

2.1 Introduction

In recent years the resolution obtained in three-dimensional reconstruction of biological complexes using cryo-EM has been considerably improved both through better instrumentation and new software tools. Simultaneously, more effort has been put into automation of the data collection and processing steps. As a result of these developments a large amount of software for cryo-EM is now available. At the same time there is still considerable potential for improvement in terms of resolution, automation and ease of use.

In cryo-EM single particle reconstruction, the vast majority of particle projections are picked when the low resolution 3D structure of the complex is (or could be) known. This additional information should be used, as it allows cross-correlation searches, which are more objective than hand-picking projections, and have a better yield than automatic procedures based on local density or variance. However, such cross-correlation searches are expensive in terms of computer resources, as every distinctive view and orientation of the low resolution 3D structure requires a separate search.

There are several ways of speeding up such model-inspired particle picking. Most importantly use is made of the correlation theorem, which states that the product of the Fourier transform of one function with the complex conjugate of the Fourier transform of another, is the Fourier transform of their correlation. Proper local and resolution dependent scaling are essential to avoid false positives, but in general this is fairly straightforward. As discrete Fourier transforms are calculated using FFT routines, numerically efficient algorithms result. However, additional optimizations are still required, including two optimizations we designed and implemented in Cyclops.

First, when dealing with samples deposited on holey carbon, it often is important to select only those particles that are suspended in the film of vitreous ice and exclude particles that have attached themselves to the carbon. In order to automate recognition
of the carbon region, so that it can be excluded from computerized particle searches, we developed a new algorithm that is discussed below.

Second, efficiency can be increased if the list of 3D projections of the low resolution model used for automated correlation searches is sampled as sparsely as possible, implying uniform sampling. As uniform Eulerian or polar angle sampling produces a non-uniform set of orientations, in which certain orientations occur far more often than others, we developed an algorithm that generates such a uniform set of orientations using unit quaternions. We also discuss this new algorithm below.

### 2.2 Methods

The new methods of automated carbon masking and uniform sampling of rotational space for a model based particle selection have now been implemented as plug-ins in Cyclops software.

#### 2.2.1 Automated carbon masker

Fully automated particle picking requires a masking procedure that identifies the areas of the micrograph that contain the useful data. Usually the microscopist is only interested in particles within the holes of the carbon layer. One way of finding the proper regions is to use a carbon layer with a regular grid of circular holes. These layers, however, are not (yet) being used routinely and most of the times the holes are irregular in both size and spacing.

Currently no other automatic carbon masking algorithm for EM image processing exists. Traditional edge detection and segmentation algorithms do not work due to the extreme high noise in this type of cryo-EM image. Here we present a new masking algorithm which is based on the relatively high variance within carbon regions. Since this is also a property of regions containing aggregates, these are also masked by the method. The method consists of a series of image processing steps, which try to keep the edge information of EM images as much as possible while dealing with the high
noise levels.

Figure 1. Intermediate results of the carbon masking algorithm on a micrographs of 50S ribosomal subunits showing: (a) original image, (b) result of edge detection, (c) removal of sparse points and growth of masked regions, (d) initial mask, (e–h) iterative closing of the (scaled down) initial mask.

The algorithm for automated masking of the carbon comprises the following steps: First, the image is scaled down to a smaller size by binning $n \times n$ pixels, where $n$ is an integer number, thus speeding up processing and suppressing the noise level by averaging (Fig. 1a). Second, edge detection with a large size Prewitt operator (Prewitt, 1970) is applied, and the result is converted to a binary map using a self-adaptive threshold based on the statistics of the gray scale distribution of the image (Chang et al., 1995) (Fig. 1b). Next, sparse points, usually located outside the carbon layer, are removed from the binary map. The amount of pixels with value 1 within a given distance of the pixel examined must exceed a threshold, otherwise the pixel is set to
zero. This leaves most of the points in a carbon region, whereas the sparse points in regions with just vitreous ice are erased.

Subsequently, the regions near every none-zero pixel are searched in the map resulting from the edge detection result of the second step using a lower threshold in order to construct a new binary map. This allows the regions already masked to grow and holes in the mask to be filled leading to better segmentation (Fig. 1c). Next, an initial mask image is created by binning the binary map by a large factor ($10 \times 10$ pixels) (Fig. 1d).

In the last step, a closing process for the mask image is performed. In the primary mask image some holes are present in carbon regions, and some false positive points in non-carbon regions. A new algorithm is used to close and smooth the image (Fig. 1e-h). The basic idea is that the edge of the carbon region is continuous and smooth and doesn’t have sharp turns. A masked point on the edge of a carbon region should have at least four masked neighbors or the mask at this pixel will be removed. A similar rule for unmasked points is applied. After several, normally 5–6, of these iterative closing operations, the mask map will converge to a nice map with smooth edges. By default five iterations are performed, but this value may be changed by the user.

The plug-in produces mask images for carbon regions of the EM micrographs, but large ice aggregates and over-crowded blocks are masked out as well.

In our experience, the module works well for most EM images, producing adequate masks in ~95% of cases.

### 2.2.2 Even sampling of 3D rotation space

We define the *angular distance* to be the angle about a common rotation axis that maps one object onto another. The centres of mass of both objects are superimposed, and the rotation axis goes through this joint centre of mass. Orientation space is sampled by a discrete set of 3D orientations with a precision of $\Delta$ if the angular distance between any orientation from the continuum of possibilities and at least one orientation from the sampled set, is smaller than $\Delta$. 
There are many ways to sample orientations with a given angular distance. One example is Eulerian sampling, where each of the Euler angles is sampled by Δ and all possible combinations of (α, β, γ) are generated. There are many definitions of the Eulerian rotation angles, and here we use the convention of a rotation by α about the Z-axis, then a rotation of β about the new Y axis and finally a rotation of γ about the new Z-axis. Clearly, when β=0, only the sum of α and γ is defined, a property also known as a gimbal lock. At even sampling of Euler angles, rotations with a final rotation axis close to the Z-axis are therefore overrepresented, resulting in a non-uniform distribution of orientations in 3D rotation space.

Polar angle sampling suffers from similar problems. Here the orientation is defined by the angles (φ, θ, κ), where κ is the right handed rotation about an axis with polar coordinates φ and θ. Uniform sampling of the polar angles is also inefficient, as at (κ=0), φ and θ are undefined, and at (θ=π/2), φ is undefined. Therefore, in uniform polar angle sampling, orientations around (κ=0) and (θ=π/2) are overrepresented, again resulting in a non-uniform distribution of orientations in 3D rotation space. Sampling of φ and θ does not have to be linear, but is also possible to use platonic solids like the dodecahedron and the icosahedron. Here, the vertices of the polyhedron can be used as sampling points covering the sphere uniformly. The sampling density of φ and θ may be increased by subsampling the triangular or pentagonal faces of the polyhedron (Yershova and LaValle, 2004). This sampling, however, only describes a rotation with 2 degrees of freedom (2D). The in-plane rotation still needs to be sampled in a separate step and the same objections remain: orientations crowd around (κ = 0).

Orientations can also be defined by quaternions, which do allow uniform sampling of rotational space. Quaternions are 4D complex numbers of the form:

\[ q = a + xi + yj + zk \]

where: \[ i^2 = j^2 = k^2 = -1 \]
\[ jk = -kj = i \]
\[ ki = -ik = j \]
ij = -ji = k

Rather than a real axis and a single imaginary axis as in ordinary, 2D complex numbers, quaternions have a real axis and three orthogonal imaginary axes. The orthogonal directions of these axes are defined by the unit quaternions i, j and k. Arithmetic with quaternions is straightforward, but multiplication does not commute, e.g. jk = -kj. In analogy to complex numbers, the following properties of a quaternion are defined:

- **Conjugation:**  
  \( q^* = a - xi - yj - zk \)

- **Sum:**  
  \( (q_1 + q_2)^* = q_1^* + q_2^* \)

- **Product:**  
  \( (q_1q_2)^* = q_2^*q_1^* \)

- **Magnitude:**  
  \( |q| = \sqrt{qq^*} \)

- **Real part:**  
  \( q + q^* = 2a \)

Quaternions are attractive for describing orientations. If:

\[
qq^* = 1 \quad (q \text{ is a unit length quaternion})
\]

\[
p + p^* = 0 \quad (\text{the real part of } p \text{ is zero})
\]

\[
p' = qpq^*
\]

then \( p' \) is related to \( p \) by a 3D rotation in imaginary quaternion space. The axis about which \( p \) is rotated to generate \( p' \) is \((xi + yj + zk)\) and the angle of rotation is \((2acos(a))\). Another useful notation of a unit length quaternion therefore is:

\[
q = \cos(\kappa/2) + xi + yj + zk,
\]

where \( \kappa \) is the angle of rotation and \((x,y,z)\) is the positive direction of the rotation axis.

Suppose \( q_1 \) and \( q_2 \) are unit quaternions, then both define a 3D rotation of a volume \( V \), generating two copies \( V_1 \) and \( V_2 \), respectively. This being the case, the operation that rotates \( V_1 \) onto \( V_2 \) is defined by the quaternion product \( q_2(q_1^*) \). The angular distance \((\Delta_{1,2})\) between the two objects is given by the real part of the quaternion \( q_2(q_1^*) \) according to:

\[
\cos(\Delta_{1,2} / 2) = (q_2q_1^* + (q_2q_1^*)) / 2
\]
The orthogonal distance between $q_1$ and $q_2$ is given by:

$$|q_1 - q_2|^2 = (q_1 - q_2)(q_1 - q_2)^* = (q_1 - q_2)(q_1^* - q_2^*) = q_1 q_1^* - q_1 q_2^* - q_2 q_1^* + q_2 q_2^* = 1 - q_1 q_2^* - q_2 q_1^* + 1$$

Substitution of Eq. (1) in Eq. (2) shows that the orthogonal distance between two unit quaternions $q_1$ and $q_2$ is strictly related to the angular distance ($\Delta_{1,2}$) between the two new objects that are generated by rotating an object using either $q_1$ or with $q_2$, respectively:

$$\cos(\Delta_{1,2} / 2) = 1 - |q_1 - q_2|^2 / 2$$

Hence the problem of uniformly sampling 3D rotations is reduced to the more straightforward task of uniformly sampling the 4D hypersphere of unit quaternions. In other words, we need to uniformly distribute the quaternions over the hypersurface. When done uniformly, the nearest neighbor distance can substitute $|q_1 - q_2|$ in Eq. (3), establishing its association with $\Delta$, the precision of sampling.

Platonic solids also exist in 4D space, where beasts like the hexacosichoron live, which has 1200 triangular faces and 120 legs (vertices). Similar to sub-sampling 3D platonic solids (which can generate better spherical approximations like the soccer ball), 4D platonic solids can also be sub-sampled if a higher precision is required (Yershova and LaValle, 2004). Fig. 2 shows a polar representation of 5880 rotations generated by subsampling the hexacosichoron. The angular distance between the rotations is about 7.59°. For comparison, naive Euler sampling of rotational space with a similar angular distance yields 53,088 rotations.
Automated carbon masking and particle picking

Figure 2. Polar representation of 5880 sampled rotation quaternions using subsampling of the 4D hexacosichoron. Green points represent the viewing directions, whereas the red bars indicate the in plane rotations. The angular distance is about 7.59°.

As a plug-in Cyclops, we implemented rotational sampling using all 4D platonic solids and their sub-sampled approximations. The user has the choice of generating sets of between 5 and 5880 orientations, corresponding to an angular precision of sampling that ranges between $2 \pi /5$ and $2 \pi /30$.

In the current application, the generated projections are used by a plug-in for particle picking using template matching. Fig. 6 shows a typical example of the result of template matching when picking 50S particles using 16 projections. Clearly, the plug-in detects the particles, but at these low-sampling densities the advantages versus Euler sampling for projection generation are fairly small. More significant improvement is achieved in, for instance, projection matching for orientation assignment.
2.3 Implementation

In the methods section, the principles and general methods have been described. Here we will focus on some technique details for implementation.

2.3.1 Implementation of automated carbon masking

To mask the carbon region, block ice and over-crowded particles, special image processing methods are needed for the micrographs with extremely high noise. Not all the known image processing operators are suitable to deal with high noise, though they may work very well in most other cases. We have to select and customize the operators to make them be really functional with lower signal-to-noise ratio (SNR) images.

Firstly, edge detection with a large Prewitt operator.

There are lots of known edge detectors, such as, the Roberts operator, Sobel operator, Laplacian or Gaussian, etc. Here 9*9 size Prewitt operator (see below, Prewitt H1 & H2) was selected, because it can also suppress noise by averaging. For instance, if every pixel is 12.7Å, a 9*9 size Prewitt operator covers 114.3Å (=9*12.7) width/height in real space, which is comparable to single particle size of 200-250 Å.

Edge detector Prewitt H1

```
-1 -1 -1 0 1 1 1 1 1
-1 -1 -1 0 1 1 1 1 1
-1 -1 -1 0 1 1 1 1 1
-1 -1 -1 0 1 1 1 1 1
-1 -1 -1 0 1 1 1 1 1
-1 -1 -1 0 1 1 1 1 1
-1 -1 -1 0 1 1 1 1 1
-1 -1 -1 0 1 1 1 1 1
-1 -1 -1 0 1 1 1 1 1
```
Automated carbon masking and particle picking

Edge detector Prewitt H2

\[
\begin{bmatrix}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

The filtering result (Fig. 1b) clearly shows the edge of carbon region, block ice and 50S particles.

**Secondly, binarize image with self-adaptive threshold.**

When transferring grayscale images to binary images (only white and black colors), we can hardly find a fixed threshold which works well for all of the images. To solve this problem, an adaptive threshold needed to be implemented, which was realized here according to the statistic attributes of each of the images.

We chose \( \text{Threshold} = \beta \times \text{Average} \) (\( \text{Average} \) is the averaged grayscale value of all pixels). Clearly, the threshold is adaptive, it changes for every different image that has different total mean grayscale value. The parameter \( \beta \) can be changed by the user. The default value by experience is 2.3, which is stable for most of EM images. For EM photos taken in different facilities, it may need to be slightly adjusted.

The other techniques, generating a primary mask image and iterative closing of the final mask, have already been presented in the methods section.

In conclusion, the abundant variance information is well used in the algorithm of automated carbon masking. It relies on the fact that the carbon region and white ice region always have higher variance than the vitreous ice region. In a few cases, when the variance of the carbon region is very low for example, insufficient exposure of the
carbon region may cause wrongly classifying carbon regions as vitreous ice. In these cases, the failed data normally have lower quality and should be excluded in later processing.

2.3.2 Implementation of even sampling of 3D rotation space

Regular polytopes of 4D quaternion

Regular polytopes (also called platonic solids) are convex solids where all the building blocks (vertices, edges, faces, hyperfaces) have the same characteristics. That is, vertices have the same number of neighbours, edges are all the same length, polygons are all the same shape and area, and hyperfaces have the same volume (Bourke, 1993).

In 2 dimensions, the type of regular polytopes is infinite. E.g. regular triangle (3 edges), square (4 edges), right pentagon (5 edges), right hexagon (6 edges) etc.

In 3 dimensions, there are 5 regular polytopes (regular 3-polytopes) (Fig. 3): Tetrahedron (4 faces), Cube (6 faces), Octahedron (8 faces), Dodecahedron (12 faces), Icosahedron (20 faces).

In 4 dimensions, there are just 6 regular polytopes (regular 4-polytopes) (Fig. 4): Simplex (5 tetrahedral cells), Hypercube (8 cubic cells), Cross-polytope (16 tetrahedral cells), 24 cell (24 octahedral cells), 120 cell (120 dodecahedral cells), 600 cell (600 tetrahedral cells).

In geometry, a four-dimensional polytope is sometimes called a polychoron (plural: polychora).

<table>
<thead>
<tr>
<th>Tetrahedron</th>
<th>Cube</th>
<th>Octahedron</th>
<th>Dodecahedron</th>
<th>Icosahedron</th>
</tr>
</thead>
</table>

Figure 3. Five plantonic solids. (From wikipedia)
Automated carbon masking and particle picking

Figure 4. *Wireframe perspective projections of six convex regular 4-polytopes. (From wikipedia)*

The coordinates/quaternions of the vertices of these regular 4-polytopes are known and can be found in numerous tables. E.g. the coordinates/quaternions of the vertices of unit 4-simplex are:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>-0.559017</td>
<td>0.559017</td>
<td>0.559017</td>
<td>-0.25</td>
</tr>
<tr>
<td>0.559017</td>
<td>-0.559017</td>
<td>0.559017</td>
<td>-0.25</td>
</tr>
<tr>
<td>0.559017</td>
<td>0.559017</td>
<td>-0.559017</td>
<td>-0.25</td>
</tr>
<tr>
<td>-0.559017</td>
<td>-0.559017</td>
<td>-0.559017</td>
<td>-0.25</td>
</tr>
</tbody>
</table>

**Evenly sampling rotation space by subdivision of regular polytopes of 4D quaternion and its application in 3DEM**

As discussed above, the problem of uniformly sampling 3D rotations can be reduced to the more straightforward task of uniformly sampling the 4D hypersphere of unit quaternions. To uniformly sample the 4D quaternions, a subdivision procedure is performed:

(i) Select one of the regular 4-polytopes as the base of sampling, e.g. the simplex.

(ii) Then, construct a stack in the program and push the known quaternions of the 4-polytope onto the stack, e.g. the 5 vertices of Simplex are pushed onto the stack.

(iii) Calculate the geometric mean of each two quaternions/vertices in the stack, until all the combinations are used; then push the medians to the stack

1 Not all combinations of quaternions are allowed: only those combinations which result in a new quaternion with a length that is close to 1 are included. In the algorithm this is optimized by a specific selection process, but it goes too far to describe it here in great detail.
(iv) This subdivision step can be done iteratively, until the precision of sampling (the angular distance $\Delta_{1,2}$ between two neighbor vertices) reaches the user requirement.

Subdivision into edges, faces, and cells of certain regular polytopes may result in a series of discrete number of quaternions in the stack: 5, 8, 15, 16, 24, 32, ..., 5880, 6120, 26520, 30360, ... as showing in Table 1.

<table>
<thead>
<tr>
<th>Basic platonic</th>
<th>Simplex</th>
<th>Cross-polytope</th>
<th>Hypercube</th>
<th>24 Cell</th>
<th>600 Cell</th>
<th>120 Cell</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic vertices no.</td>
<td>5</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>120</td>
<td>600</td>
</tr>
<tr>
<td>Subdivision 1 iter.</td>
<td>15</td>
<td>32</td>
<td>40</td>
<td>120</td>
<td>840</td>
<td>1320</td>
</tr>
<tr>
<td>Subdivision 2 iter.</td>
<td>65</td>
<td>176</td>
<td>168</td>
<td>600</td>
<td>5880</td>
<td>6120</td>
</tr>
<tr>
<td>Subdivision 3 iter.</td>
<td>285</td>
<td>848</td>
<td>712</td>
<td>2712</td>
<td>30360</td>
<td>26520</td>
</tr>
</tbody>
</table>

Table 1. Numbers of uniform quaternions generated by iterative subdivision of regular 4-polytopes.

It means that we can not randomly select any number of quaternions for uniformly sampling 3D rotation space, but we can certainly select a sampling with the precision that is higher than what we need. To use the quaternions, we need to convert the 4D quaternions to Euler angles triples, which are accepted by most other programs to represent rotations.

An equivalent problem in 3D space can be solved by sampling the unit sphere using platonic solids like the dodecahedron and the icosahedron. The subdivision procedure was degraded to sampling 3-dimensional regular polytopes (regular 3-polytopes), generating uniform distributions on the 3D unit sphere. It is worth mentioning that although this distribution evenly samples rotation axes (e.g. Fig. 5), it does not evenly sample rotation space, as no in-plane rotation is included. Nevertheless, also this result is still very useful in current popular 3DEM reconstruction software packages.
Automated carbon masking and particle picking

Figure 5. Uniform sampling on the surface of 3D unit sphere based on subdivision of icosahedron, 1002 sampling points with ~6° angular distance. There is no in-plane rotation included (no red bar indicates the in-plane rotation compared with Figure 2).

The new algorithm can be applied in model-based particle searching, in which the projections generated from a starting model and a set of Euler angles are used as references. For this searching, the in-plane rotation is not necessary, because the in-plane rotation is already included in the procedure.

2.3.3 Implementation as plug-ins in Cyclops

As mentioned above, new methods for automated carbon masking and uniform sampling of rotational space have been implemented as plug-ins in the Cyclops software (Fig. 6). Other methods currently implemented as Cyclops plug-ins cover a wide range of common image processing techniques, such as compression, low-, high- and band-pass filtering and edge detection. Methods previously implemented in the Tyson program (Plaisier, 2004) for automated selection of particles have now been
re-written as Cyclops plug-ins. The sorting of particles, a prominent feature of Tyson, is an intrinsic part of the Cyclops program.

Figure 6. Cyclops has a friendly graphic user interface (GUI) and plug-in architecture. The new algorithms for carbon masking and uniform sampling of rotation space (applied in model based particle searching) are marked by green ellipses. In the sub-window of micrograph, the black area is the result of automated carbon masking, blue boxes indicate selected particles, which are segmented and shown in the sub-window of particles gallery below.

The new algorithms were written in C++/Python. They communicate with Cyclops through XML files. The XML file describes the input and the type of output it produces (e.g. a new particle set).
An XML file example of automated carbon masker:

```
<CyclopsPlugin>
  <module>Carbon masker</module>
```

36
Automated carbon masking and particle picking

The information of the XML file is used to construct an input dialog window (shown in Fig. 7). A simple wrapper of Cyclops will pass the input parameters to the program.

Figure 7. Dialog window for entering the input parameters for the module of automated carbon masker.

These applications are now routinely used through the Cyclops interface. Due to the
modular structure of Cyclops software, the plug-in applications can be easily extended and updated.

2.4 Conclusions

Two new algorithms dealing with the automation of particle selection are presented. The automated carbon masking routine allows automated removal of carbon region and only searching particles in the vitreous ice region of micrographs. The algorithm of even sampling of 3D rotation space can be used to generate uniform projections from a starting model and a set of rotational-equal-distance vectors. These projections are further used in a template matching procedure for particle picking. Both algorithms boost the automation and efficiency of particle selection in the step of data preparation. These algorithms greatly assisted in the structure determination of the stalled 50S ribosomal complexes described in chapter 4.

Acknowledgements

Thanks to Jasper R. Plaisier for his great help in embedding the new algorithms in Cyclops software suite. Cyclops is available under a GPL license and can be downloaded from http://www.bfsc.leidenuniv.nl/software/Cyclops.
References
