

Measurements in COMPASS tokamak plasmas

Bencze Attila

Tartalomjegyzék

1. EDICAM Measurements	2
1.1. Introduction: the EDICAM camera	2
1.1.1. Hardware specifications and capabilities	2
1.1.2. Actual setup	4
1.1.3. Operating instructions	4
1.2. Video diagnostic applications	4
1.2.1. Reconstruction of the light profile	4
1.2.2. Brief introduction to 3D graphics	5
1.2.3. The solution of the visibility problem by introducing the Z-buffer	8
1.3. Blender manual	9
1.3.1. Simple functions	9
1.3.2. Manipulating objects	11
1.3.3. Working with cameras	12
1.3.4. Node system	13
1.4. Exercise tasks	15
1.4.1. Camera operation	15
1.4.2. Image filtering	15
1.4.3. Calibration of the camera coordinates and generation of the Z buffer	16
1.4.4. Light profile reconstruction	17
2. Principles of Li-beam Measurement	20
2.1. Li-beam	20
2.2. Hardware modifications	21
2.2.1. High voltage system	22
2.2.2. Magnetic shielding	22
2.2.3. The ion source	24
2.2.4. Recirculating neutralizer	26
2.3. Slow Detector	27
2.4. APD detector	27
2.5. Measurement tasks	29

1. fejezet

EDICAM Measurements

Suggested skills:

- Elementary optics,
- Basic linear algebra,
- 3D modeling, computer graphics,
- Knowledge of image manipulation software (GIMP, Photoshop).

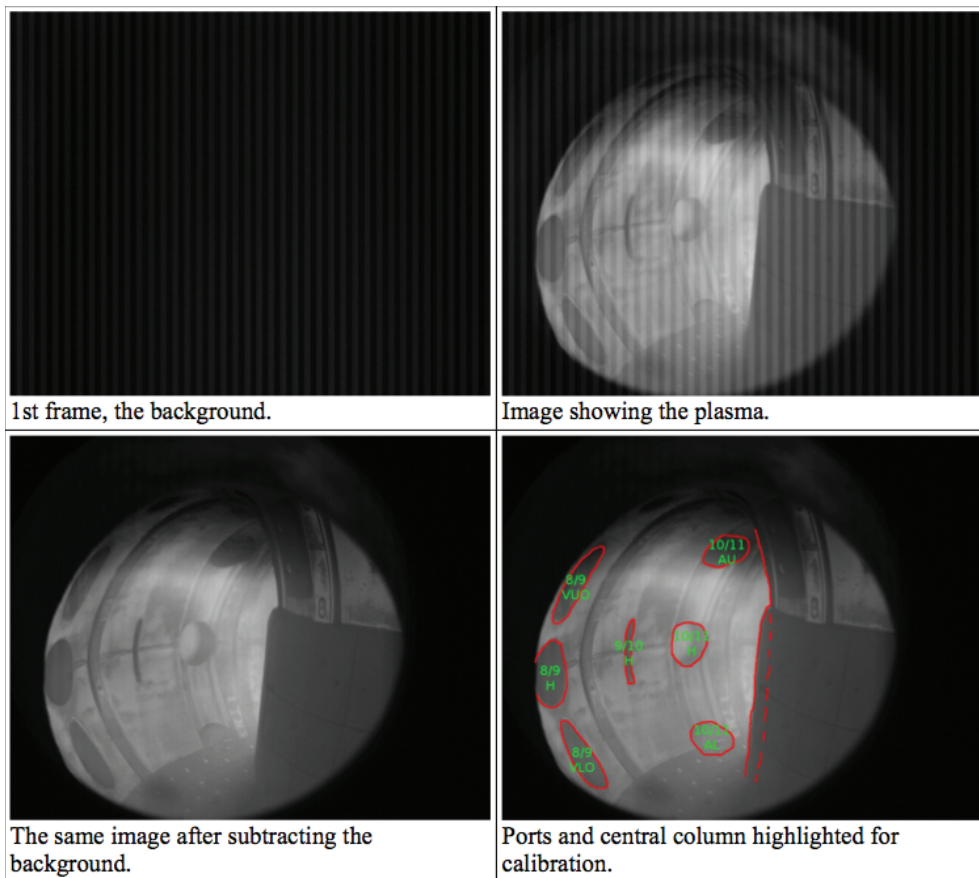
1.1. Introduction: the EDICAM camera

1.1.1. Hardware specifications and capabilities

The camera was initially designed for W7-X, as a monochrome visible camera with fast readout. Because the W7-X is expected to emit high neutron flux, the camera has been designed to be somewhat radiation resistant. Direct exposure will be avoided by using mirrors, but still, they will be present close to the plasma. This task can be eased by splitting the camera into two parts: the sensor, with just enough electronics to communicate with a data acquisition PC, and the PC itself with its software suite, which could be placed far from the radiation sources. The other design requirement for the camera was to have some "intelligence", hence the name. The cause was the length of the shots at W7-X: huge amounts of data would be generated by the many cameras, when all would be operated in full-frame. The other point of making the camera intelligent is to observe some phenomena (pellets, particle traces, fast events) with higher frame rates, by reducing the region of interest (ROI).

High frame rate was a natural requirement, as a camera with scientific applications.

To achieve fast readout, which was one purpose of the camera design, the EDICAM uses 16 separate channels, each with its own analog electronics and ADC. This limits



the ROI x offset and width to be a multiple of 16, and because of the slight differences between the analog offset and gain of the different ADCs, also generates periodic noise or stripes.

This striping is filtered in the processed images, but when evaluating data the unfiltered images should be used, because they contain the most information.

Subtracting the background is generally enough, but this only removes the differences in the offset, not in the gain, and per-pixel differences could also be present.

Practice shows that subtracting the first image shows decent results, but for proper calibration both dark and bright test pictures are needed. Taking a dark picture is easy, because it only requires closing the shutter, but to provide a homogeneous bright surface is a more difficult task. An Ulbricht sphere for example is considered as a very homogeneous light source.

The sensor itself is a 1.3 megapixel monochrome CMOS sensor, with 1280×1024 pixels of resolution.

1.1.2. Actual setup

At COMPASS, the EDICAM cameras are connected to their respective PCs by an optical link. The cameras are located in the tokamak hall, while the computers are in a diagnostic room. The first camera is located at port 5/6 HT, the second camera is not installed at the time of this writing. While the first camera is in a tangential port, it can see the ports on the opposing wall, and a part of the central column and some ports, as visible on the pictures. The location of these structures provide useful data for camera calibration. Their engineering parameters can be found in [1].

1.1.3. Operating instructions

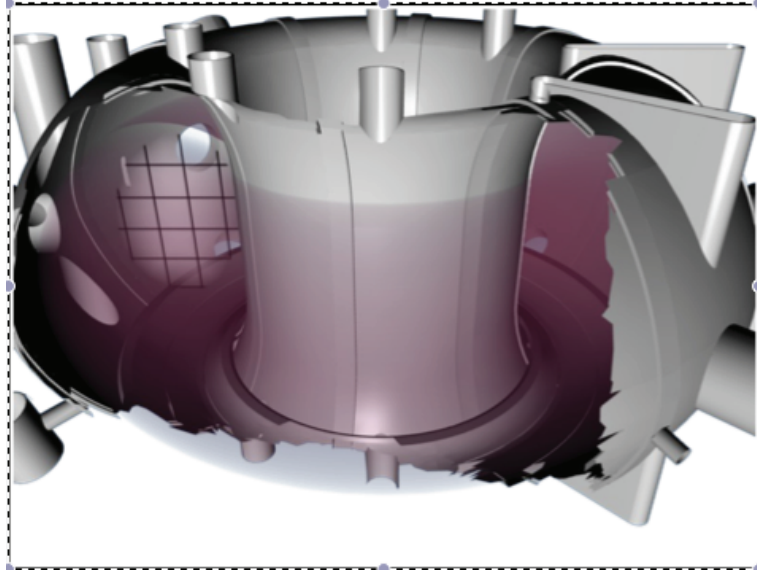
Please refer to the wiki page[4] of the COMPASS EDICAM camera, where the most recent manual can be found for the camera operation. Because the camera can be removed and installed at a different location, if such an operation is done, a new calibration is necessary. The change in the optical path also changes the magnification and the focus. Because there is generally enough light coming from the plasma (without spectral filters), the camera is usually operated with small aperture, to provide a wide depth of field, but for higher frame rates, it may be useful to increase it. When focusing the camera, the aperture should always be opened, to provide a small depth of field, otherwise least precise focusing would produce sharp images.

1.2. Video diagnostic applications

1.2.1. Reconstruction of the light profile

As mentioned, the tokamak is an axially symmetric device, so ideally no physical quantity has dependence on the toroidal angle. In reality, this is not always true, but during most of the operation, it is. One thing that can be measured by a video diagnostic, is the light emitted by the plasma. Based on the symmetry assumption, we can imagine a rectangular area, quantized as a matrix, emitting light, and by rotating this plane along its axis we get the visible light. The question is: how to reconstruct this half-plane, the poloidal cross section of the plasma?

The plane with the grid represents the light profile, which we would like to reconstruct based on the images. Even if we knew this light profile, and would like to generate the picture based on it, the process requires a complex technique called volumetric rendering, and that would still ignore reflections. Doing this backwards is even more difficult and time consuming, requires the use of optimization algorithms with a high set of parameters, with bound constraints (light intensity can not be negative for example). The problem can be solved easily to some degree, by using an approximation.



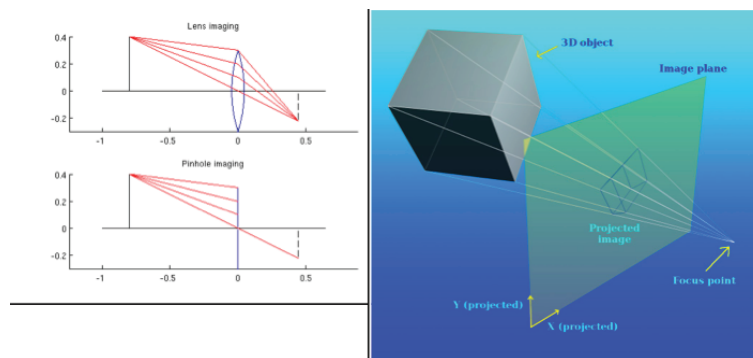
1.2.2. Brief introduction to 3D graphics

Before going into any detail, we need to discuss 3D graphics, because it will be necessary for the proper evaluation of the image data. 3D graphics has many applications in computing (like in games, CAD, even user interface nowadays), but its purpose is mostly to deliver 2D images from 3D models. Our screens and paper sheets are two-dimensional, and can not visualize 3D data directly ? even the so called 3D displays produce only stereographic images, which are two- dimensional. Direct 3D visualization is possible with volumetric displays. In our case the situation is different: we have two-dimensional data, which was projected to the sensor of the camera by the observation optics, and we would like to reconstruct the light intensity based on the images. As you might have guessed it by now, the conversion from the three dimensional models to 2D images is a surjection, and so it is not generally invertible. The camera has lines (or cones) of sight, each line (cone) corresponding to a point (pixel) in the sensor, but the depth information is lost during the measurement.

Without the use of more observation points and tomographic routines, it is impossible to determine the distance of any object. However, in our case we have a symmetry, that simplifies the situation: the toroidal symmetry of the tokamak. By knowing this symmetry, inversion is possible to some extent.

The ?transfer function? from the three dimensional world to our vision is provided by the optics of our eyes, which focus the incoming light from a certain location to a certain point in the retina. If our eyes are out of focus, then the point becomes a spot, and our vision becomes blurred. Cameras work the same way, they just use optics made of glass and an electronic sensor.

A pinhole optic always provides a clear picture, because any sensitive point in the



sensor can only be reached by light coming from a single direction. Now you understand, if not have already understood, that why photographic lenses contain variable apertures: by placing an aperture after the lens, the lens starts to behave less like a lens and more like a pinhole. The depth of field increases at the expense of blocking some part of the light.

A pinhole collects much less light than a typical lens (an ideal pinhole of zero diameter blocks all), but a lens has to be in focus in order to provide a clear picture, and it can do it only from a certain distance. They also suffer from distortion, aberration, etc. However, when they are in focus, their projection is very close to a pinhole optic, and this is called the perspective projection.

The figures illustrate how the projection works: every three dimensional point is connected to a focus point, and an image plane is placed between them. Where the lines of sight intersect with the image plane, we get the projected coordinates.

As you can see, it is very similar to the projection of the pinhole optics, the only difference being the placement of the image surface (in front of the focus point, rather than behind it). To project 3 dimensional coordinates to a surface, the following parameters are required:

- The coordinates of the camera (focus point) (c)
- The orientation of the camera (θ)
- Definition of the field of view (f)
- The relative shift of the projection plane (e)

The focus point will be the location of the imagined camera. The first step in the projection is called the 'camera transform', which consists of a shift and three adjacent rotations. If the camera was placed at the origin of the coordinate system, only a rotation would be necessary to set the camera (transform the objects) to the proper direction. Because in general it can be placed at any point, the shift of the object coordinates is also needed before the rotations.

Because the rotations in three dimensions do not commute, they have to be applied in a certain order. There are many conventions, we will use the Tait-Bryan convention, which is widespread in computer graphics. The mathematical formulas (from [5]): The camera transformation:

$$\begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{pmatrix} \cdot \begin{pmatrix} \cos \theta_x & 0 & -\sin \theta_x \\ 0 & 1 & 0 \\ \sin \theta_x & 0 & \cos \theta_x \end{pmatrix} \cdot \left(\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} - \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \right)$$

And the projection:

$$\begin{aligned} b_x &= (d_x - e_x) \frac{e_z}{d_z} \\ b_y &= (d_y - e_y) \frac{e_z}{d_z} \\ b_z &= b_z \end{aligned} \tag{1.1}$$

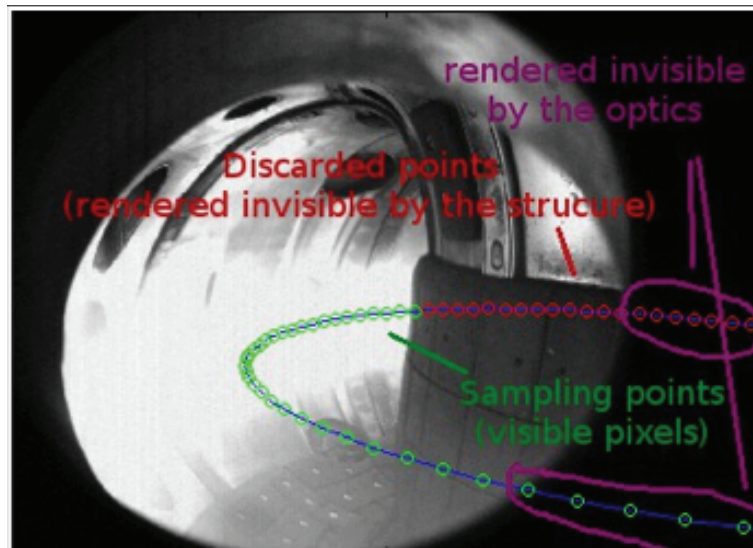
The vector \mathbf{b} contains the projected coordinates. Only the x and the y components are used for drawing, but the z (distance) component will also be important for our task. Now we have an algorithm to convert 3D coordinates to 2D. Let us go back to the image processing! 2.3. Intensity measurement by line averaging A simple way to calculate the light profile, is to average the light measured by the camera. Since we wish to reconstruct the poloidal cross section of the light profile, by using the toroidal symmetry, we must average the light on toroidal trajectories, which are circles. A 3D circle (or more generally an ellipse) can be defined by the following equation:

$$\mathbf{r}(\varphi) = \mathbf{O} + \mathbf{A} \cos \varphi + \mathbf{B} \sin \varphi$$

The idea is to sample the light value on the image on these toroidal circles.

The average intensity will correspond to the emitted light. Please note that this is a very simplified algorithm, with the intent to introduce you to image processing. The physical correspondence of this method is clearly questionable. For example we have ignored the light reflected from the walls, ignored that the hottest part of the plasma emits no visible light, ignored that in reality one should follow the magnetic field lines instead of toroidal trajectories, and there is not really a mathematical relation between the averaged intensity and the light profile. In spite of these, it generally gives decent results.

This algorithm is very simple, but there is a catch: we have to exclude the invisible points from the averaging. When a generated coordinate is not visible from the viewpoint, it has to be excluded, otherwise it would falsify the result. The solution of this task relates to yet another problem in computer graphics, called the visibility problem.



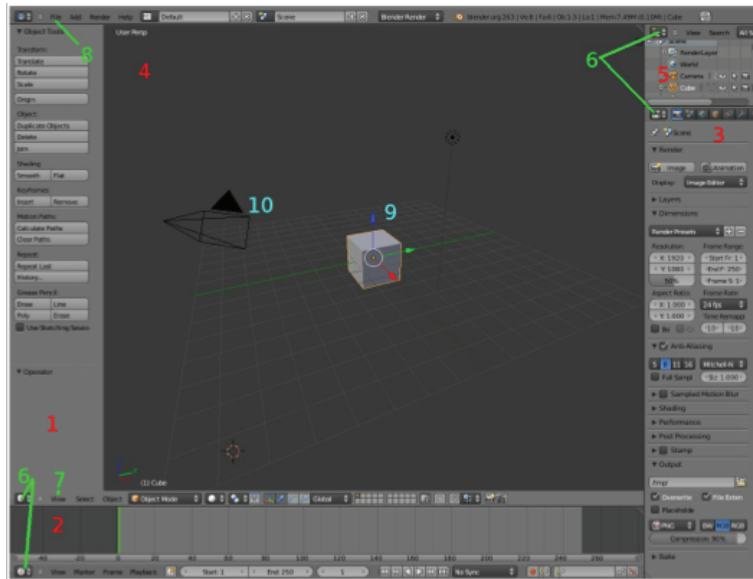
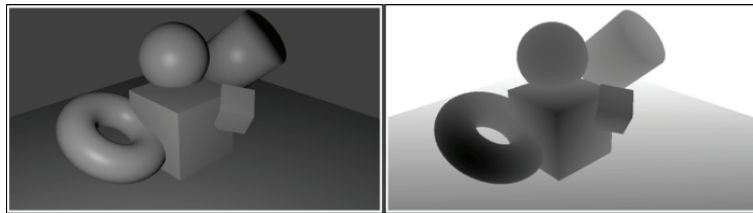
1.2.3. The solution of the visibility problem by introducing the Z-buffer

When the coordinate transformation from the three dimensions to two is done, the transformed X and the Y coordinates are used as the screen coordinates, but the Z coordinate is discarded, since it contains no information about the location of the vertex on the screen.

However, it is produced by the transformation, and it represents the distance of the object from the camera.

One solution of the visibility problem is to store these values in the memory. This way when the image is being produced, not only the memory area representing the visible picture is filled with information, but another memory area is also being filled, which is called the 'Z-buffer'. When a pixel is drawn, it is checked whether the Z coordinate of the pixel is smaller or larger than the already present value in the Z-buffer. If it is, then the pixel is visible, and so it is drawn, because it is closer to the camera than any previously rendered pixel with the same X and Y coordinates. But if it is not, then it is discarded, because a pixel that is closer to the camera is already present at that location. Every time a new pixel is drawn on the screen, the Z-buffer is filled with its distance value.

This is a very common and robust solution to the visibility problem, however we have ignored some difficulties with edges and transparent pixels, but these do not affect our problem. In GPUs this process has long been hardware accelerated, but we shall use a software implementation. If we had this depth information, then our exclusion problem would be very simple: we could just do a Z-buffer check on the transformed coordinates. But the camera images contain no depth information, so we have to use



a different solution, we will generate the depth map seen by the camera, from the 3D model of the tokamak! This requires the calibration of the camera.

1.3. Blender manual

We shall use the Blender software for a lot of tasks, so its basic knowledge is necessary. Blender is a free and open source software, available for download at <http://www.blender.org/>. We shall give only a basic introduction, its capabilities are greatly beyond our needs. This manual covers the 2.6 version. For any topic not covered in this manual, or for a more detailed description, please refer to the official wiki at: <http://wiki.blender.org>.

1.3.1. Simple functions

After starting the program, we see the following screen by default:

The interface is divided into panels. Panels can be created and hidden with mouse grabbing. Squared brackets mean the shortcut keys, in this case for hiding and showing panels. The default panels are:

1. Left sidebar: tool shelf, [T],
2. Bottom sidebar: timeline,
3. Right sidebar bottom: properties menu (not to be confused with object properties!),
4. 3D view,
5. Right sidebar top: outliner.

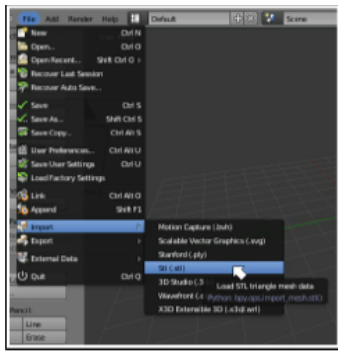
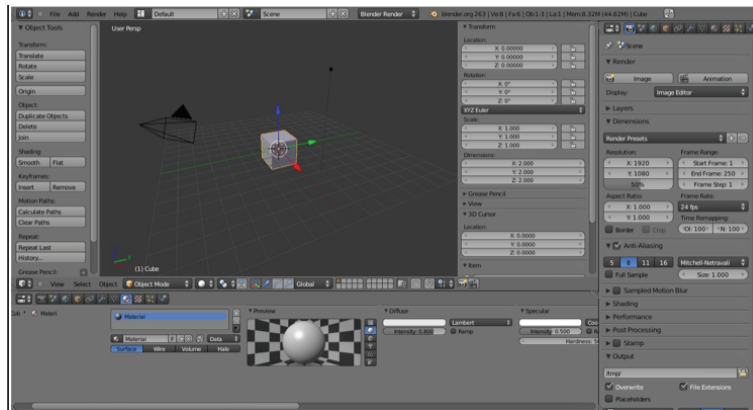
Other numbered UI elements are:

6. Window type selector,
7. View menu,
8. File menu,
9. 3D objects
10. Camera

Because we will not be using the motion picture generation features of Blender, we should set a different view: change the bottom sidebar to properties and hide the outliner by pulling up the right bottom sidebar over it by its handle. Bring up the object properties bar by pressing [N]. By right-clicking on the empty panel areas of the sidebars, their arrangements may be changed between horizontal and vertical. You can select objects in the 3D view by right clicking on them. If objects overlap, and not your desired object became selected, press the right mouse button, while holding the cursor at the same location, until it is. Right clicking cycles the selection. You can rotate the 3D view by mouse grab while holding down the middle mouse button. You can shift the view by the same gesture while you hold down the SHIFT key. You can also zoom by this way, while holding down CTRL instead of SHIFT. The size of the interface elements can also be changed this way: if you consider the buttons too large or too small, for example in the object properties bar, try zooming them! Note that it works a bit differently for horizontal and vertical panels. Zooming in the 3D view is also possible with mouse scrolling. Experience with the controls for a while! Also try the view menu.

If you select an object (only the cube is in the view so far, but the camera can be selected too), its properties show up in the object properties bar. As you can see the default cube is in the center, has dimensions of 2 in all directions, it is not rotated.

Try to change these values, and note their effects! You can change the value by either clicking on the corresponding field and entering a new number, or pressing down the left mouse button while the mouse is over the entry field, and grabbing it. When you grab, the effect is immediate.



1.3.2. Manipulating objects

Blender supports different types of objects, like meshes, curves, armatures, lamps and cameras. We are interested in meshes and cameras only for our task. Try the Add in the file menu, and try to add some objects! Some objects have properties that can only be set when adding them: spheres, cylinders, torii and cones can be constructed from a lot of segments, you can set exactly how many in the tool shelf. If you apply any transformation to the object, further setting of this value becomes impossible. You can also move, rotate and scale the objects in the 3D view. For translation press the [G] button and move the object with the mouse. Naturally you can only move the object in your view, not in depth, so for 3D positioning at least two viewports are necessary. Blender supports 'quad view', you can switch it on and off in the view menu. Scaling is possible after pressing [S], for rotation press [R]. If you press [R] twice, a different rotation becomes possible, and the cursor also changes.

You can select all objects with [A], and delete the selected objects with [Del] (you have to confirm the deletion). Naturally you can import meshes from other software. Many file formats are supported, but most of them are extensions, and they have to be enabled before use. In the default installation, blender handles .stl files among a few others, and we have a model of the COMPASS in STL.

Sometimes a mesh is made of too many triangles, and working with it slows the computer down. You can find a tab called object modifiers on the properties panel. You can add several modifiers and their effect will be shown in the 3D view while you edit the parameters. As long as you do not press apply, their effect is always computed, the result is not stored, therefore blender may slow down because of it. Now we are trying to reduce the resource cost of our operations, so when you have set the parameters, pressing 'apply' would be a good idea. There is a modifier called 'decimate', which can reduce the triangle count of a mesh by a given factor, which can be set in the 'ratio' field. For example: 0.05 means only 5% of the triangles will remain. Try it on a more complex object, like a sphere or a torus with a lots of segments!

1.3.3. Working with cameras

Cameras behave like meshes regarding location and rotation. They can also be added, selected and deleted just like meshes. The important difference is that you can see through them! Try to add a camera, and then try to move and rotate it! It will be placed at the location of the 3D cursor, directed down. You can view the scene through the active camera, by selecting camera in the view menu. Please note that the active camera is not necessarily the selected camera! The active camera can be selected in the properties panel, in the scene tab.

Set the newly added camera to active, and change the view to the camera. Now you see the scene from the focal point, the visible part is highlighted in the center of the view.

If the camera was selected, then you can transform it using the object properties bar, and you can see how the field of view changes as you move and rotate the camera.

The size of the viewing screen can be set on the render tab on the properties panel, under dimensions. Changing these values may cause the aspect ratio to change.

The seventh parameter, the focal length can also be set on the properties panel, on the object data tab. Make sure that the active camera is selected, then change to the camera view if you are not still in it. Under lens you can choose projection type, perspective or orthographic. Stay with perspective, if it is selected, then you can change the focal length and observe how the field of view shrinks or expands with it. The angle of view can be calculated by the formula:

$$\begin{aligned}\alpha &= 2 \tan^{-1}(d/2f), \\ e_z &= \frac{2f}{d},\end{aligned}$$

where d is the size of the sensor, f is the focal length, e_z is the distance from the view screen.

Still another useful feature, is that you can assign a background image to a camera, which will be visible in the camera view. With the camera selected, in the object proper-

ties bar enable background images and click on add image. For now select any image file you like by clicking on open.

From version 2.63 you can select the option show on foreground, which speaks for itself. Use a lower opacity in this case to see the model.

A comfortable way to position the camera is called fly navigation, you can turn it on in view ? navigation or with the hotkey [SHIFT+F]. In order to use this navigation to position the camera you must be in camera view!

The controls are the following:

- Forward/backward (actually increase/decrease speed): mouse scroll or Numpad +/- . Dodge (left/right/up/down): middle mouse button + drag.
- Rotate (left/right/up/down): moving the mouse in the desired direction.
- Cancel: [ESC], Confirm: [ENTER].

An important part of the projection, is the near and far clipping. It might be needed to hide polygons too close to or too far from the camera. The clipping can also be set on the object data tab, under lens.

1.3.4. Node system

The last functionality needed for our task is Blender's ability to generate the Z-buffer information about our image. Again, not going into deep details, we will use the node system of blender to perform this for us. Nodes are essentially functions, that operate on data, either generated by the rendering engine, or input by hand, or produced by another node.

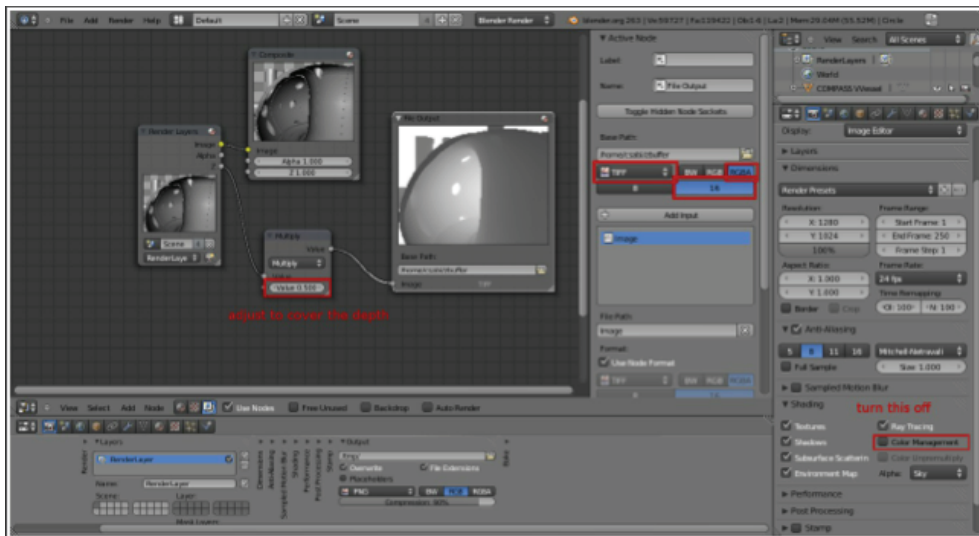
Select node view instead of the 3D view, so that the main part of the screen shows the nodes only. Or rather their place, because nodes are disabled by default.

After you changed the view, the menu also changes, you have some menu points and some new icons. Select render nodes from the icons (this is the last one), and use nodes from the recently appeared checkboxes. Two nodes should show up, render layers and composite. If they do not show up or you delete one them, you can bring them back with Add → input→render layers and Add→output→Composite respectively.

Notice that render layers has more outputs, image, alpha and Z by default, but many others can be switched on in the render tab under layers→passes.

By default, the image output of render layers is connected to the image input of composite. You can make connections with mouse dragging, one output into one input.

A rendering is required for the results to appear. You can achieve this by pressing [F12]. We wish to save our results, so add an output file node. Just like in the 3D view, the node details can be brought up by pressing [N]. If you select the output file node, you can set the name and the format of the file. We need better accuracy, so select TIFF



as the output format, and set it to 16 bits. Also set the color profile to RGBA, BW and RGB does not work always correctly with 16 bits, even though we do not wish to encode color information to the picture. With mathematical functions it would be possible to utilize all the four channels, and output 64 bit Z buffer, but 16 is enough for our task, so we will not bother with binary coding.

The Z output of the render layers has values measured in Blender's units. So for example if a pixel was 3 units from the camera, then the Z value at that pixel will be 3. The clipping of the camera has no influence on this behavior. But the inputs of the output nodes must fall between zero and one to produce a valid (not clipped) image.

You can extend the range beyond this one unit by using math nodes. They can be added in Convertor → Math. The operation can be selected, for example we could use multiply. Connect the Z output to the first input of the math node, connect the output to the file output. If an object is at most 10 units from the camera, then multiply it with 0.1, and the values now will fall in the range. Render, and you should see the Z buffer as an image in the output node. It should also be saved at the specified location.

Very important note: there is an option in the render tab on the properties panel, called color management. It can be found under shading. Be sure to TURN IT OFF! It was incorporated into Blender to produce images with more realistic colors, but it alters the levels of node outputs as well! It applies a non-linear scaling to all output images, and so the generated Z buffer will have false values! If it is turned off, the correspondence between the intensity values in the generated file and the distance from the camera at a given location will be linear.

	1	2	3	4	5	6	7	8	9	10	11	12	row
1	1	2	3	4	1	2	3	4	1	2	3	4	
2	1	2	3	4	1	2	3	4	1	2	3	4	
3	1	2	3	4	1	2	3	4	1	2	3	4	
4	1	2	3	4	1	2	3	4	1	2	3	4	
col													

1.4. Exercise tasks

Typically 2 or 3 person work in the camera group, and the tasks should be divided among them.

1.4.1. Camera operation

The first and most important part is to learn to operate the camera. This should be done by every member. For this please refer to the head of the camera group and the wiki pages.

The first task is to take pictures during shots. Then try to take advantages of the capabilities of the camera! Try to define ROIs around interesting areas and measure with higher framerate. Keep in mind that you will see only blackness with a closed shutter!

1.4.2. Image filtering

As mentioned in the introduction, the raw output images need filtering before further processing, due to the design of the camera. Try various methods, like background subtraction, FFT filtering (because the difference in the gain is periodic), and compare the results. Try to calibrate the offset and gain on a per ADC and on a per pixel basis.

This figure illustrates how the ADCs are connected, if there were only four of them: each pixel has its ADC numbered inside. The multiplexing is done by the image sensor. First, the CMOS sensor has internal gain, so its response to incoming light will be like the following:

$$U(xy) = t \cdot [I(xy) \cdot \text{Intgain}(xy) + \text{Intoffset}(xy)].$$

I is the light intensity on the pixel with indexes x and y . U is the voltage output of the cell and t is the integration (exposure) time.

Which is further complicated by the similar behavior of the ADCs:

$$N(xy) = U(xy) \cdot \text{ADCgain}(x \bmod 16) + \text{ADCoffset}(x \bmod 16).$$

At first, per pixel differences can be ignored. With a closed shutter no light can reach the sensor, and the output will be:

$$N(x \bmod 16) = t \cdot \text{Intoffset} \cdot \text{ADCgain}(x \bmod 16) + \text{ADCoffset}(x \bmod 16)$$

Because the exposure time can be set to very short, the ADC offset can be determined, since the first part will be close to zero. With a long exposure, on the contrary, the first part becomes dominant. From two images, taken with different exposures, both values can be calculated:

$$\begin{aligned} \text{ADCgain} &= \frac{N_2 - N_1}{(t_2 - t_1)\text{Intoffset}}, \\ \text{ADCoffset} &= N_2 \left(1 - \frac{1 - \frac{N_1}{N_2}}{1 - \frac{t_1}{t_2}} \right). \end{aligned}$$

Since the internal offset is not known, the calculation is not straightforward for absolute calibration, but we are only interested in the relative differences between the gains of the ADCs ($\text{ADCgain} \cdot \text{intoffset}$), which can be easily calculated, without knowing the internal gain.

Since these formulas can be calculated on a per pixel basis, try to do it for every pixel, and try to analyze statistically its periodicity!

Try to derive the formulas for calibration with a homogeneous light source! The ratio of $\text{Intgain}/\text{Intoffset}$ can now be calculated. How?

1.4.3. Calibration of the camera coordinates and generation of the Z buffer

We have the three dimensional model of the tokamak, and know the R-Z coordinates of its poloidal cross-section. We have also discussed how to project three dimensional coordinates into two dimensions, and know how to deal with the visibility problem. However, as mentioned in the introduction, a definition of a camera is required for the projection. The direction, the location and the focal length of the camera has to be calibrated for our measurement.

This means a seven parameter fit, which is not an easy task. This could be done based on the optical design, but in reality the actual parameters always differ from the initial design, because for example, the camera can be placed anywhere on a rail.

For the calibration a recent image from the camera is needed, because during removal and re- installation the exact location usually changes.

Once the calibration is done, the Z buffer can be generated by 3D modeling software (such as blender). Open a new file, delete the default cube.

Try to import the model of the tokamak with file \rightarrow import \rightarrow stl. It is a huge file, and importing it can consume time and memory. Working with this model is also very resource expensive. If you managed to import it, select it with the right mouse button, scale it down first. The units are in mm, so a 1/1000 reduction will make it fit into our drawing area better. Select the imported object, then use the scale in the object

properties to shirk it down. Since the model is much more detailed than our need, use decimate, and reduce the polygon count to around 10-20% of the original. Depending on the resources of your computer you can use either higher or lower setting. Click on Apply to perform the changes.

The tokamak's model is not yet in the center. Go to object → transform → geometry to origin. Now the model is almost in the center, but not exactly. It has centered the object in its bounding box, but since the model is not exactly symmetric, this center is at a slightly different location than where it should be. Move it to X:0.0012, Y:0.0015, Z:-0.0514 by altering location in the object properties.

Set the resolution on the render tab in the properties panel to 1280x1024. The X and Y aspect ratios should be both set to 1, and the render scale to 100%.

Define a camera roughly at the location of the camera port. You can find in [1], you should look for port 5/6 HT. Please keep in mind that while the locations of the ports have not changed since this document was made, their functions may have. You can convert the toroidal coordinates in the port list to X-Y-Z Cartesian, based on the illustrations:

$R = R_0 + r \cos \theta$	$X = R \cos \theta$	$Y = R \sin \theta$	$Z = r \sin \theta$
---------------------------	---------------------	---------------------	---------------------

The major radius (R_0) is 0.56 m for COMPASS.

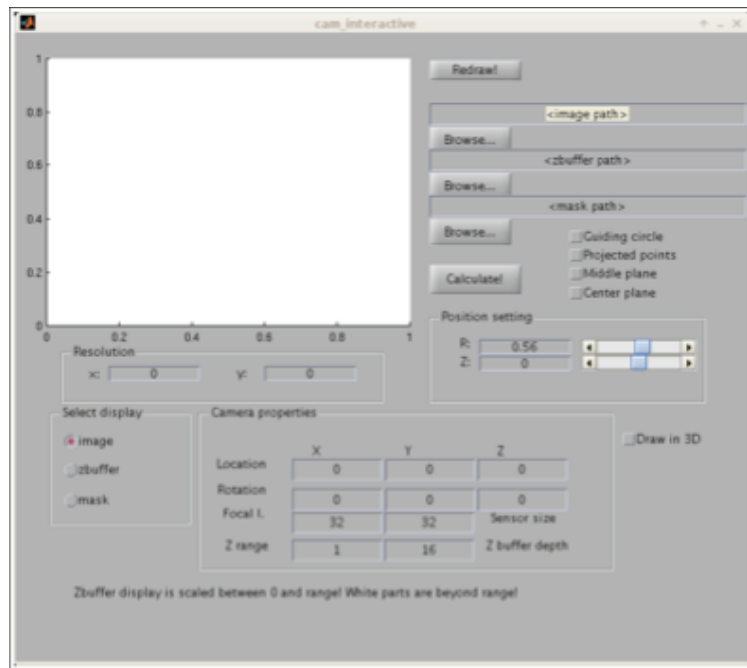
Roughly set the rotation of the camera to show the same as pictured by the EDICAM. Now try to calibrate the camera! Switch to camera view in the view menu, set an image from EDICAM as a background image, put it in the foreground by checking show on foreground . Please use a full frame image for this! Change the opacity until both the image and the model are clearly visible. You can edit the image to highlight the ports as in the introduction, for example with red. If you are more familiar with graphics software then you could draw the highlights on a new, transparent layer, and then just save this layer in a format readable by Blender that keeps the transparency (like PNG).

Modify the location, rotation, focal length and even the clipping if necessary. When the parameters are set roughly, you can use fly navigation for the fine tuning.

If you have the camera calibrated, write down its settings, location, rotation and focal length. Render the image, generate its Z-buffer, and save it as a 16-bit TIFF image. A lot of graphics applications have trouble with 16 bit TIFFs, but MATLAB and IDL can read it.

1.4.4. Light profile reconstruction

This task consists of creating poloidal light profiles from the images, based on the discussed line- averaging method. A matlab program was written to ease the task, but you are encouraged to implement it from scratch, based on this document. However, we shall discuss its usage and functionality. The program starts by calling



cam_interactive

It shows up the following screen:

The controls speak for themselves. You must load an image, a zbuffer and a mask, in order to perform the calculation. The program tries to read

calibration_data.txt

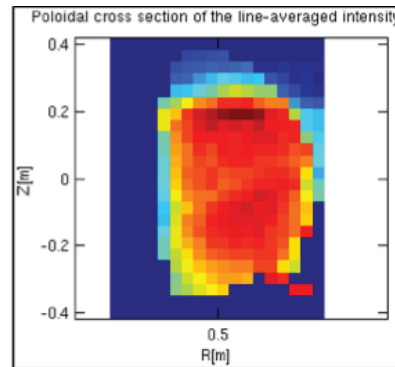
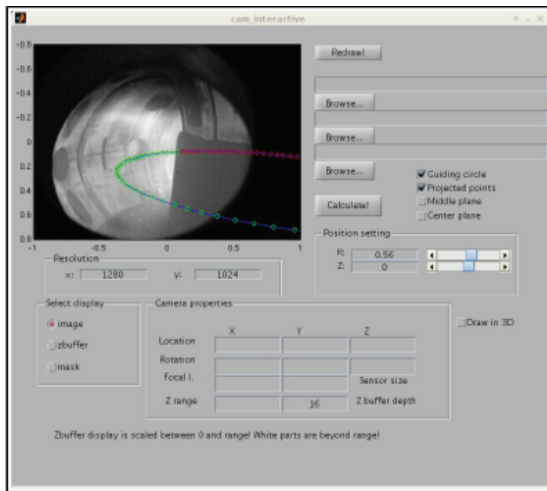
If it succeeds, the location and the rotation values are set upon initialization. The file should contain 2 rows and 3 columns, in the order visible on the screen (first line: location, second line: rotation). Note that the program expects the angles in radians (Blender can also be set to use radians)!

If you load an image, it will get displayed on the graph. If you also load a zbuffer image, you can use the checkboxes beside the calculate button. It will show various types of three dimensional projected coordinates, based on the calibration. You can use the position setting to vary the location. The program will show the visible vertices with green, the invisible (because of the 3D structure) with red.

You also have to load a mask to perform the calculation. It must be a black and white picture, but saved in 4 channel RGBA (all images are expected in it).

The black parts will be excluded from the averaging. The part seen by the camera should remain white.

Do not forget to set the depth and the range of the Z buffer, and also specify the focal length and the sensor size. These parameters should from the calibration.



If you have loaded everything, confirmed that the 'projected points' distinguishes well between visible and invisible points, you can press 'calculate'.

The code will generate a 18x24 poloidal grid, and do the averaging on 100 points on each trajectory. The results will be displayed and saved as

`result_average.matrix`

The further tasks are: examine how the program works, how projection is made, and how is Z buffer check performed. Implement the same algorithm in a more native language like C or Fortran, or try to create your own, more optimized matlab code. Generate videos of these data. Do not forget to normalize all images to the same level!

Try to determine the location of the plasma edge and the center based on these images. Compare the results with other diagnostics.

2. fejezet

Principles of Li-beam Measurement

Lithium ions are extracted from a thermionic ion source and accelerated in an ion optic. The ions are neutralized in Sodium vapour. The given neutral atoms are shot into the plasma where they will be excited to the 2p state by the plasma electrons. The excited state decay with the emission of a photon of characteristic wavelength ($\lambda = 670.8\text{nm}$). These photons can be measured using the appropriate Lithium filter and different observation methods (CCD camera, Photomultiplier, Photodiode or Avalanche Photo Diode).

The typical beam energy of these systems is around 30-60keV, which provides about 10-20 cm penetration into the plasma. The beam energy is limited by the neutralization efficiency (goes down to about 70% at 60keV) and the finite lifetime of the observed atomic state.

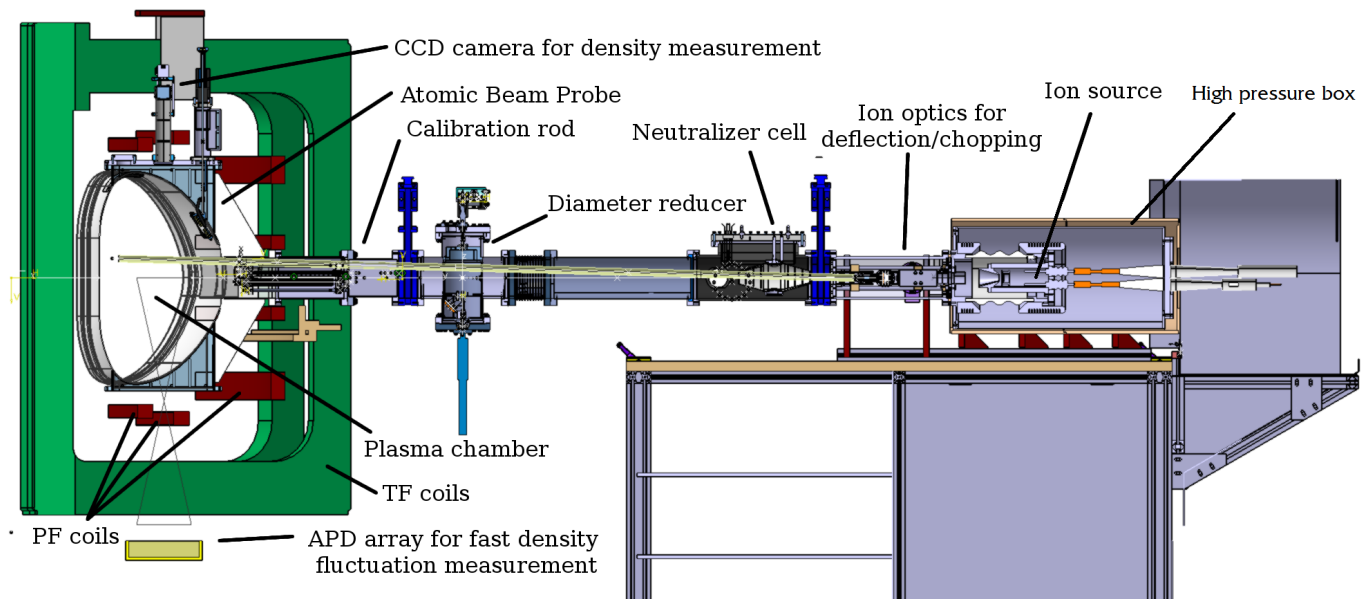
Observing the intensity and the fluctuations of this Li resonance line along the beam, it is possible to reconstruct the density profile and the two-dimensional correlation of the electron density fluctuation. Using a pair of deflection plates the beam can be either chopped or poloidally deflected. The switching frequency of this system can be increased up to a 400 kHz. Scanning the beam poloidally it is possible to achieve a quasi twodimensional measurement. Additionally poloidal flow velocity can be measured with this technique.

2.1. Li-beam

At COMPASS, which is a small size ($a=20$) tokamak, there is still a unique possibility to measure the turbulence via the full small radius (at low plasma density) following its changes from the edge till the core.

The aim of this diagnostic is:

1. Measurement of radial profile of plasma density with $\sim\text{cm}$ spatial and ~ 10 milisecond temporal resolution (standard density diagnostic tool).



2.1. ábra. Set-up of the diagnostic beam

2. Characterization of density fluctuations in radial–poloidal plane on time scale characteristic for plasma turbulence (advanced BES diagnostic tool)
3. Measuring poloidal flow velocities in the plasma.
4. Measurement of edge plasma current and its perturbation (ABP)

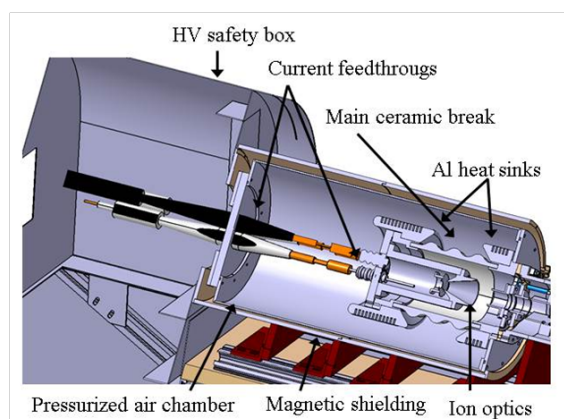
In the following chapters the beam hardware modifications (high voltage, magnetic shielding, ion source improvement), the neutralizer, the beam modelling (RENATE) and based on that- the schema of the observation system and the Atomic Beam Probe concept are described. First measurements are also showed followed by the summary and the conclusion of our results.

2.2. Hardware modifications

The system can be seen on Fig. 1. It consists of the accelerator (with the ion optic), the beam manipulation chamber (with two deflection plate pairs and a Faraday cup), the neutralizer chamber, a flight tube, a diaphragm (with a second Faraday cup) and a calibration rod.

2.2.1. High voltage system

To enhance the voltage up to 120kV several modifications were necessary—comparing it with the conventionally used Li-beam systems (ASDEX, JET). Figure 2. shows the HV part of the system.



2.2. ábra. High voltage system

The system consists of two high voltage power supply, one 120kV/10mA and the other is 120kV/0.25mA. The first one is connected to an isolation transformer (Powersources, HVTT-1K-120K) which is followed by a second transformer to produce the necessary heating current (about 70A). The heating current is controlled by a thyristor unit (Eurotherm, 7100A Single-phase power thyristor). It means that the heating current is enhanced to the main high voltage. This line is connected to the ion source, which takes place in the Pierce electrode (see Fig. 4).

The second power supply is connected directly to the extractor electrode. The second power supply is on the acceleration voltage, the difference between the two power supplies is the extraction voltage.

The third part of the ion optic, called puller is connected to ground.

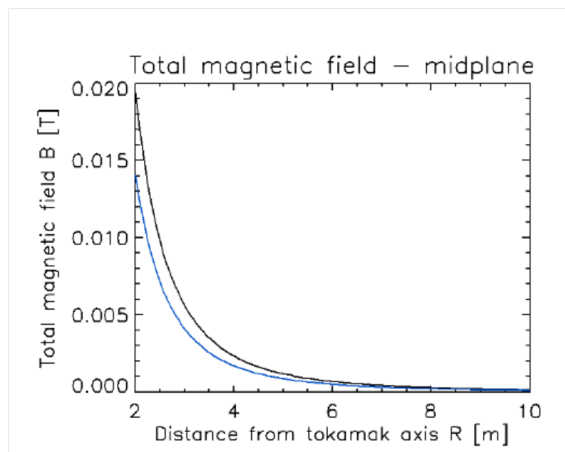
Both power supply is placed in the table, under the vacuum system.

The main ceramic insulator is 140 mm long made of china (in Hungary). There are aluminium heat sinks on both sides. The pressurized air chamber placed around the ceramic break. It is filled up to +1.5 bar (air) to isolate the 120kV voltage. The current (and HV) feedthroughs on the pressurized air chamber do not break the isolation of the HV cables (they only fix the cables and close the pressurized chamber).

2.2.2. Magnetic shielding

The system has to be magnetically shielded because of the stray magnetic field around the COMPASS tokamak, see Figure 3.

This value is calculated, but some measurements resulted in the same magnetic field value. The Lithium beam system is at about $R=3.5-5$ m.



2.3. ábra. Stray magnetic field of the COMPASS tokamak

Because of the stray magnetic field is basically vertical, the Lithium beam system consists of only horizontal ports till the neutralizer.

The beam manipulation and the neutralizer chamber are made a kind of soft iron with minimum 10mm thickness. All the connecting ports and flanges are made from the same material. To avoid oxidation—after welding—20 μm nickel coating were applied (using chemical coating). The magnetic shielding of the accelerator is placed around the pressurized air chamber with 10mm thickness using the same soft iron material.

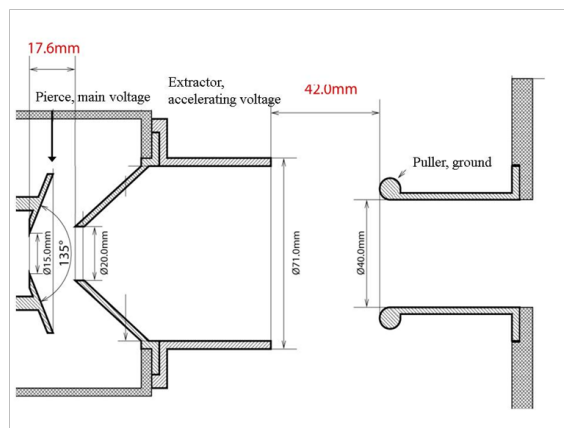
The magnetic field attenuation depends on the thickness and the radius of the soft iron cylinders, the first approximation is $A = \mu \frac{d}{D}$ (where μ is the magnetic permeability, d is the thickness; D is the diameter of the cylinder). We can assume 1000 for μ (conservative estimate) in case of a soft iron material.

The diameter and thickness of the beam manipulation and the neutralizer chamber are about the same, these are 140mm and 10mm. The diameter and thickness of the shielding around the accelerator are about 400mm and 10 mm.

This way we can assume 70 times weaker magnetic field in the beam manipulation and in the neutralizer chamber, while the magnetic field will be about 25 times weaker in the accelerator. The calculated magnetic field is about 3 mT (at the neutralizer) and is about 1 mT at the accelerator. The attenuated fields are than about the same, about 0.04 mT.

Calculating the deviation of the ion beam we have to take into account that in the ion optic the beam has different energies, see Figure 4.

The first investigated part is the extracting area, between the ion source (on main energy) and the extractor electrode (on the accelerating energy). The beam ions are



2.4. ábra. Ion optic of the COMPASS Lithium beam

accelerated by the extracting voltage, which is about 2-12kV at 20-120kV main beam energy. We have to calculate the deviation in the worst case, i.e. at the lowest possible beam energy (1keV average). The distance is about 100mm till the end of the extractor electrode, where the second acceleration starts (approximately). The deviation in this magnetic field range can be seen on Figure 5.

It can be seen that in this magnetic field range, below 0.1mT, the deviation—in this worst case—is still less than 2 mm at 4m distance from the ion source. This deviation is negligible.

The second part of the ion optic is from the extractor till the puller electrode, where the ions accelerate to the main energy. The distance of it is about 100mm, the lowest possible beam energy is about 10keV (the ions accelerate from 2 keV till 20 keV, the average is about 10 keV)

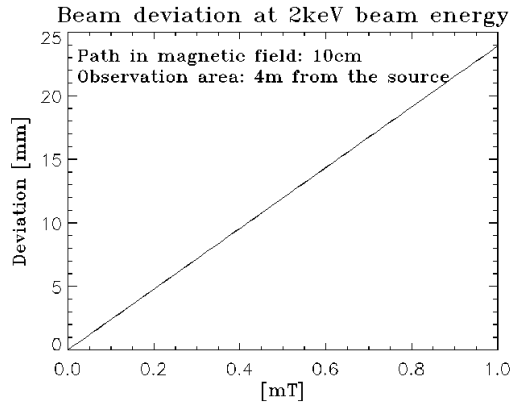
The third part, which has to be taken into account, is from the end of the ion optic till the neutralizer, about 500 mm. The smallest possible beam energy is about 20 keV.

The deviation we have to calculate is a 1 keV beam along 100 mm, plus 10 keV beam along 100 mm and plus a 20 keV beam along 500 mm, 4 m away from the ion source.

This means about the same—negligible—deviation. In this calculation there is still a double safety factor, in case of the magnetic field attenuation is only half of the assumed value, the ion beam deviation is still acceptable.

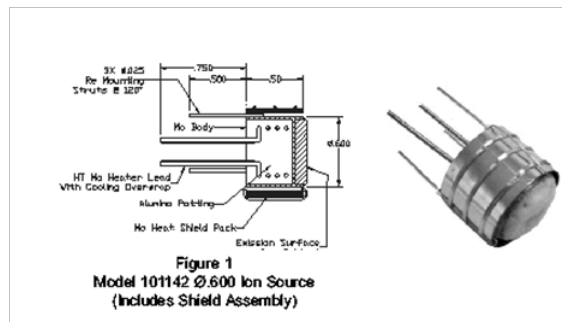
2.2.3. The ion source

The Lithium beam diagnostics usually apply the same ion source made by the Heatwave company, see Figure 6. This is one of the crucial part of the guns. The heating wire is a Molybdenum filament embedded into high purity alumina. The main part of these emitters is the porous Tungsten plug, which has 50 % porosity and the diameter of the



2.5. ábra. Ion beam deviation at 2 keV beam energy

holes are about 50 μm . The constituents of the β -eucryptite ($\text{Li}_2\text{O} + \text{Al}_2\text{O}_3 + 2 \text{SiO}_2$) or the spodumen, ($\text{Li}_2\text{O} + \text{Al}_2\text{O}_3 + 4 \text{SiO}_2$) which are the two mostly used emission material, were coated into the Tungsten plug in a vacuum vessel. These process results a thermionic ion source, which can emit maximum about 2 mA, and the lifetime of this is about 2 mAh.



2.6. ábra. The ion source

The temperature limit of the Heatwave ion source is 1200 $^{\circ}\text{C}$, fixed by the manufacturer, while the working temperature of the ion emission material is well above 1300 $^{\circ}\text{C}$. It means that this ion sources are used well above its permitted temperature. The working parameters of this source are about 5V/30A.

The main problem with these sources is the limited ion emission capacity and that the operability highly depends on the external conditions (temperature, fixing). A little change (which is not always known) can effect the decrease of the ion current or a break down.

Our newly developed thermionic heater does not contain any heating filament. The house is Molibdenum, the heater itself is a SiC disc (contaminated with some carbon).

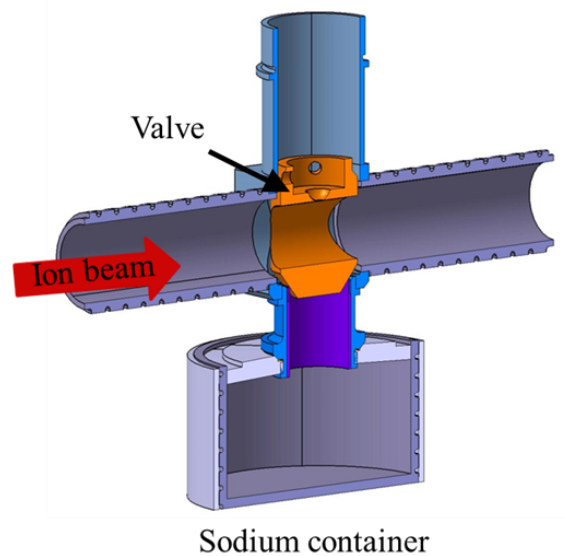
The current is led via a carbon conductor. On the top a porous Molybdenum plug can be found which can be filled up with the same ion emission material, see Figure 6.

The main difference is that this heating system does not contain heating filament which can fail during the long term high temperature application.

The working parameters of this source are about 3V/70A.

2.2.4. Recirculating neutralizer

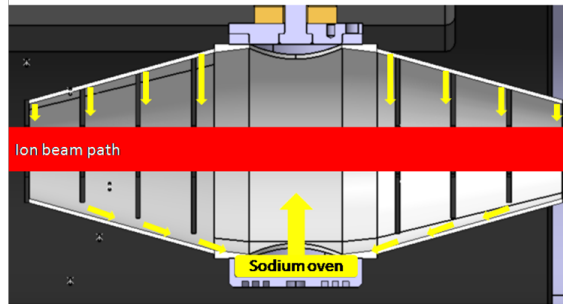
The originally developed neutralizer has a Sodium container which can be closed by a simple valve. The container continuously heated up to 280 C degrees (with closed valve) while the tube above is heated up to 300 C degree to avoid Sodium condensation in that area. Just before the discharge the valve is opened and Sodium vapor reaches the tube where the ion beam goes through and get neutralized by charge exchange effect. The loss is proportional to the pulse time (and above about 10s one have to enhance the temperatures of oven and the tube to have the appropriate Sodium vapor pressure).



2.7. ábra. The original neutralizer

The base of this newly developed neutralizer is to heat up the Sodium, produce a sodium vapour pressure in a cell and minimize the loss by condensing the sodium vapor (outside the neutralization volume). Fig. 8. shows the set-up of the COMPASS Li BES neutralizer.

The temperature of the oven is about 300 C degree, the temperature of the cones is about 150 C degrees (using air cooling). It means that part of the Sodium condensates on the diaphragms and flows back to the oven (gravitationally).



2.8. ábra. Scheme of the COMPASS Li-BES neutralizer

The Sodium oven can be kept continuously on 200 C degree, when the vapor pressure is still negligible, and 100 s is enough to enhance its temperature up to 300 C degree just before the plasma discharge. This process decreases the Sodium loss, too.

2.3. Slow Detector

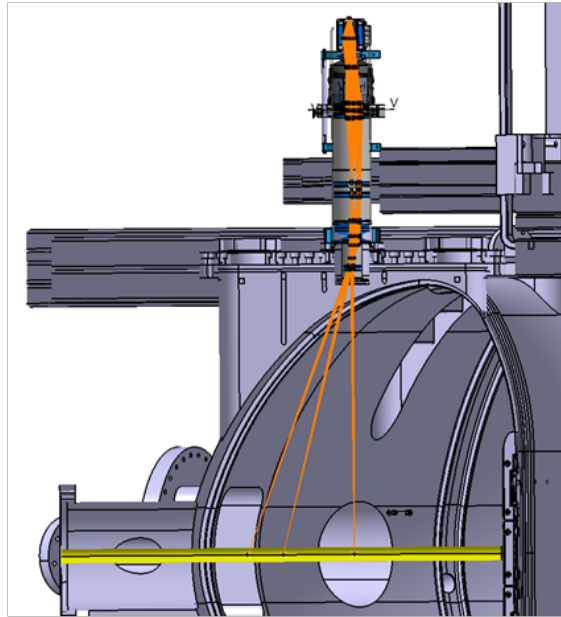
The Lithium beam diagnostic uses the complete 1/16 section. The beam entrances into the plasma horizontally from the LFS. The observation system is placed on the upper and bottom oval ports in the same poloidal cross section.

The observation of the beam is accomplished by means of two, independent optical observation systems. One of the observation systems has a modest time resolution (of the order of milliseconds) and is using a conventional CCD camera as the photon detection device (designated as slow branch); and the other observation system has a high temporal resolution (of the order of microseconds), using avalanche photodiodes for photon detection (designated as fast branch). Both branches look under 90 degrees at the beam, the slow one looking from up to down, and the fast one looking from the bottom of the tokamak upwards.

The slow branch serves the general observation of the beam, and hence an aberration corrected image is necessary. The starting element of the design is a telecentric objective, the intermediate image of which is relayed to the final objective of the CCD camera. Figure 10. shows the schematic of the design.

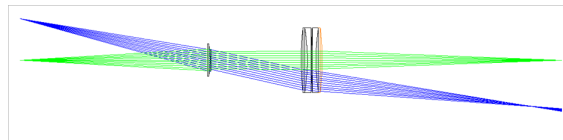
2.4. APD detector

Since the aim of the fast branch is the detection of plasma fluctuations, that requires the detection of photons in numbers as large as possible, this branch uses only a few lenses with less care taken about the aberrations of the image. Any eventual image aberrations (spherical aberration first of all) are corrected to some extent by (1) shifting



2.9. ábra. The CCD observation system.

the individual photodiodes to match correct focal positions and (2) using cylindrical lenses in front of the photodiodes. Figure 9. shows the schematic of the design.



2.10. ábra. The optical setup of the fast branch: the object is to the left at a distance of 400 mm from the 75 mm diameter $f=500$ mm front lens. The object is practically imaged 1-to-1 to the photodiodes, that are shifted normal to the optical axis to correct for the spherical aberrations. An interference filter is attached directly to the first lens, as well as the cylindrical lenses, not shown, to the photodiodes.

Since the aim of the fast branch is the detection of plasma fluctuations, that requires the detection of photons in numbers as large as possible, this branch uses only a few lenses with less care taken about the aberrations of the image. Any eventual image aberrations (spherical aberration first of all) are corrected to some extent by (1) shifting the individual photodiodes to match correct focal positions and (2) using cylindrical lenses in front of the photodiodes. Figure 9. shows the schematic of the design.

2.5. Measurement tasks

After getting familiar with the technical details of the diagnostic now we list here the measurement tasks to be performed during the measurement sessions:

1. Beam performance tests

The goal of these measurements is to learn how to operate the Li-bem. on the one hand and to measure the overall performance of the beam. These measurements do not need plasma discharges. These measurements are done using the Faraday-cup (FC) placed at the tokamak end of the flight tube. The Faraday-cup consists of a Ti-plate and a housing around it. The housing is isolated from the Ti-plate. The housing can be biased to negative or positive voltage repelling or attracting secondary electrons in order to determine the neutral current and the neutralization efficiency. Now, the specific tasks are the following:

- Centralize the ion beam path and optimize the beam focusing using the poloidal deflection plate.
- Measure the extracted current as a function of extraction voltage (Child-Langmuir curve) for different emitter temperatures.
- Calculate the neutralization efficiency biasing the FC-house and using permanent magnets to deflect the remaining part of the ion beam.

2. Light profile measurements

Light profile measurements can be performed in both vacuum shots and regular tokamak discharges. During the vacuum shot all the magnetic fields are switched on, but no plasma current is driven. The pressure for beam test in H is about $4 \cdot 10^{-4}$ mbar. For observation of the intensity distribution we use the CCD camera located at the top-central port in the beam cross-section.

- Compare the light intensities measured in vacuum shots and gas-shots (no magnetic field, no neutralization). Use the CCD camera with 100 ms integration time. Estimate the neutralization efficiency and compare it to FC measurements.
- Determine the light profile in different density L-mode shots and compare with H-mode shots. What is the signal-to-noise ratio?

3. Basics of the plasma fluctuation measurements

Fluctuation measurements can be performed using the APD detector array (18 channels), covering the outer 17 cm of the plasma. The spatial resolution is about 1cm and the temporal resolution is set to 1 microsecond. Use fast chopping for proper determination of the background level.

- Calculate the basic statistical moments of fluctuations as a function of time (how steady the discharge could be considered?).
- Calculate the relative fluctuation amplitude as a function of the radial coordinate.
- Calculate the power spectra as a function of radial coordinate. What kind of modes are seen?
- Calculate the radial correlation map of light fluctuations.

Tartalomjegyzék