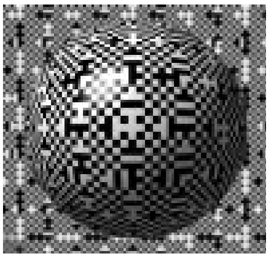
NIKHEF

NATIONAL INSTITUTE FOR NUCLEAR AND HIGH ENERGY PHYSICS



ETR 94-10

The coding of the mask for CCD_Rasnik

Project no.: 884.12

H.L.Groenstege

email: h.groenstege@nikhef.nl

Description of the mask pattern used in the CCD_Rasnik alignment monitoring system.
(patent pending)

- **A bit of history.**
- **Description of the code.**
- **Hints for designing new models.**

NIKHEF, DEPARTMENT OF ELECTRONIC TECHNOLOGY

P.O.box 41882, NL 1009 DB AMSTERDAM

CCD_Rasnik straightness monitoring system

Our work on the straighness monitoring system has been presented in “The third international workshop on accelerator alignment”, september 1993. The paper presented there was: “The Rasnik/CCD 3-dimensional alignment system”. Authors: H.Dekker, H.vd.Graaf, H.Groenstege, F.Linde, S.Sman, R.Steensma. From: Nikhef-H, Kruislaan 409, 1098 SJ Amsterdam, the Netherlands.

Since than, a lot of work has been done on improving *the mask* and on the reconstruction software. Also systematical errors were investigated and methods were developed to minimize the effects.

The code is explained in “The coding of the mask for CCD_Rasnik” by H.L.Groenstege from NIKHEF. This is a NIKHEF paper, registered in 1994, numbered ETR_94-10. This paper can be obtained from the NIKHEF library or can be downloaded as a postscript file. By Browsing the WWW version one will have access to more related information.

In the L3 experiment at Cern, Geneva one part of the detector is the muon detector. This large mechanical structure could not be build stable enough to reach the accuracy required. Instead the relative positions of the components are monitored continuously. For this purpose the Rasnik system was developed. The relative positions of the three layers of muonchambers are monitored, this is called the vertical Rasnik. The monitoring of the middle wire support frame in each wire chamber is monitored by the horizontal Rasnik system. Three systems are used here for redundancy reasons.

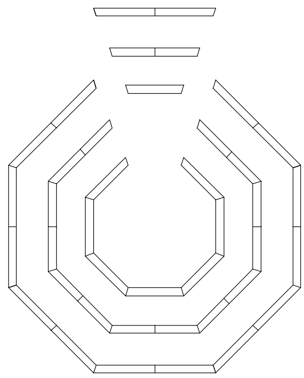
The lightspot is a homogeneously illuminated square or circle. The lens in the middle produces an image on the four-quadrant diode. The amount of light on each quadrant is measured. From the ratio of these figures the deviation from the original location of the lightspot can be calculated.

The measurements are converted into off-line corrections for the chamber data.

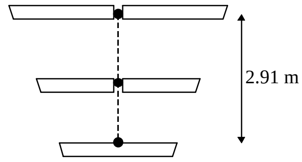
The Rasnik system has been used in several set ups after this. The system is described more extensively in the NIM publication "P.Duinker et. al., Nuc. Instr. and Methods, A273 (1988), page 814-819". See also "H.Dekker et. al., Proceedings of the Third International Workshop On Accelerator Alignment: The Rasnik/CCD 3-D Alignment system".

The L3 muon detector

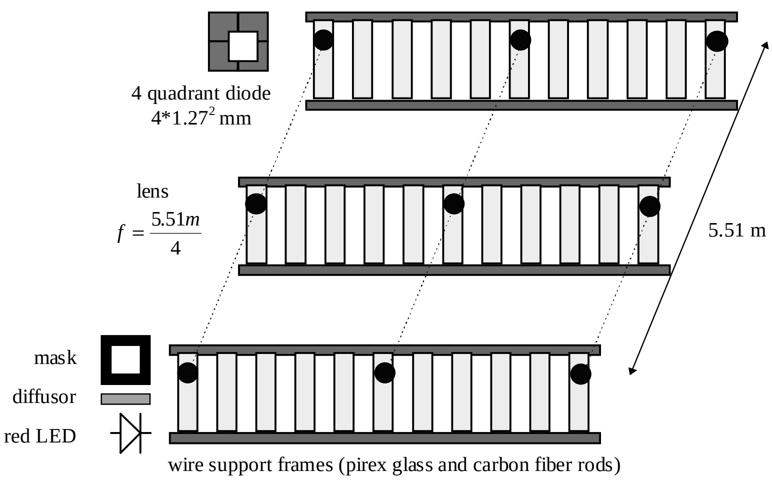
5 muon chambers form 1 octant.
8 octants form a wheel.
2 wheels form the muon detector.



4 systems in each octant
(vertical systems)



3 systems in each muon chamber
(horizontal systems)



The Rasnik system measures in two directions. The range is mainly limited by the size of the four quadrant diode. Large four quadrant diodes are available but are rather expensive. Non-linearity is caused by the non homogeneously lit mask. Improving this aspect is tedious.

For experiments to be build in the near future, alignment monitoring in larger range is needed. With the price of CCD cameras going down, this was a tempting alternative. Much more information becomes available from the large amount of pixels. The range is no longer limited by the sensor size but rather by the coding of the lightsource. The set up is briefly described in CCD Rasnik set up.

Experiments in this direction have proven that the accuracy can be satisfactory and the measurement range is large when proper coding of the light source is used. In the first experiments a sort of dartboard was used as a mask. Positive results were obtained but the accuracy depends on the part of the mask that is seen by the camera. Further from the center the accuracy becomes less. Also reconstruction of this circular shape from the frame grabber data is not trivial. These experiments lead to the idea that the mask should just have a lot of black white transitions in X and Y direction and some code for the rough position.

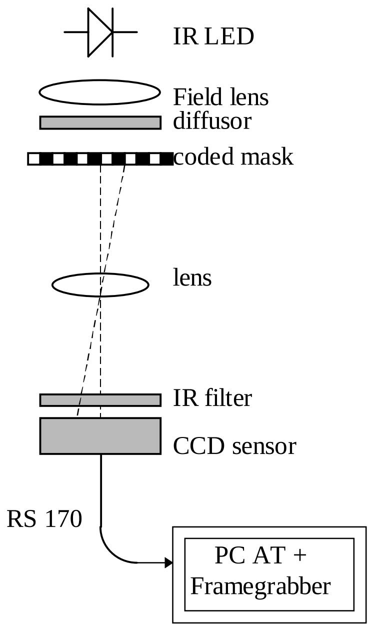
A barcode like pattern can be created by using this code for X and Y and logically exclusively or these to create the mask. Information density for precise fitting the grabbed data varies with the position. Creating large area masks requires a complex bar-code. Several ideas have been discussed and finally lead to the pattern described in this paper. The idea is that a chess board type mask contains many black white transitions which makes fitting of the acquired data more precise. Than we need some code to determine the rough location on the large chess board. At regular intervals codes are inserted for the rough X and Y location.

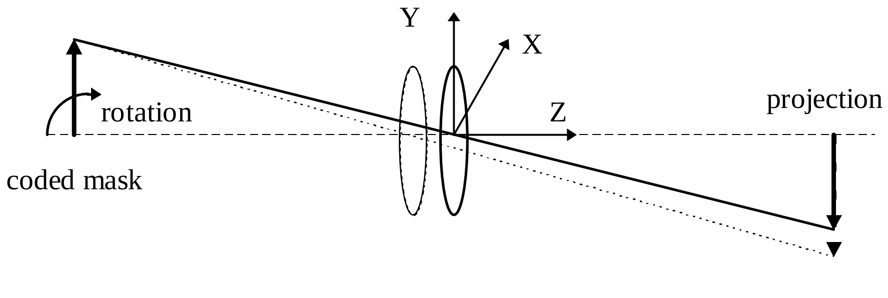
Principle

The basic idea is to create an image of a coded mask on a CCD sensor by means of a lens. The mask is lit by an infrared LED. A collimating lens increases the amount of light that reaches the projection lens. The diffuser is used to minimize effects of imperfections of the lightsource. The standard video signal is digitized by means of commercially available frame grabber.

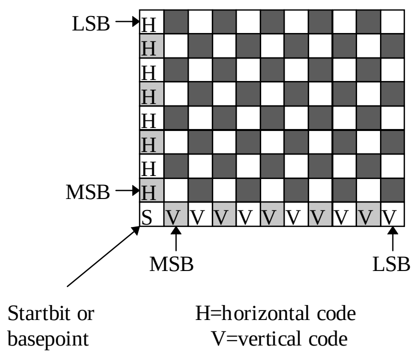
In this set up the relative position in X and Y direction is measured along the line mask, optical center of the lens and the CCD sensor. Also (relative) rotation of the mask or the sensor can be measured.

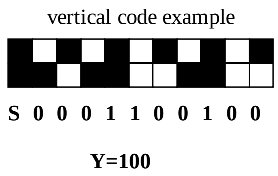
By calculating the actual image spotsize and comparing this with the mask spot size, the position of the lens along the Z-axis can be calculated.





A mask is generated by stacking position coded *basic building blocks*. We will call such a block a B^3. The horizontal and vertical codes give the position of the basic building block in the total mask. Counting starts at 0,0 in the lower left corner of the mask. A startbit is used to make recognition and decoding easier. It is the basepoint of the building block. The ones in the binary position code invert the location color. The startbit is also inverted in respect to the basic chessboard.[1]



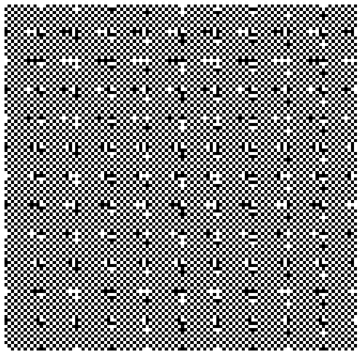


At some point it seemed that, instead of inverting the code locations, making them gray was favorable. The average illumination is more stable then. This idea was dropped because gray had to be defined as a fine black and white pattern. Then the definition becomes dependent on the type and quality of the plotting device. Experiments written in postscript and plotted on a high quality machine showed poor results.

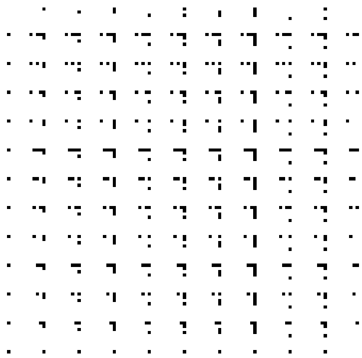
Also postscript was abandoned for this purpose because postscript interpreters use floats instead of integers. One had to choose scaling factors in a way that the interpreter was forced to round figures calculated internally in a consequent direction.

Gerber files are now used to define the mask. This is a well known format for photoplotters, used to create masks for integrated circuit production. The files generated are in ASCII format, so they are easy to transport.

[1] The thin lines in the example are not implemented on the real mask.



This mask is build by using basic block of 11 times 9 fields. Recognition of the code lines and thus the B3’s is easy due to the startbits.



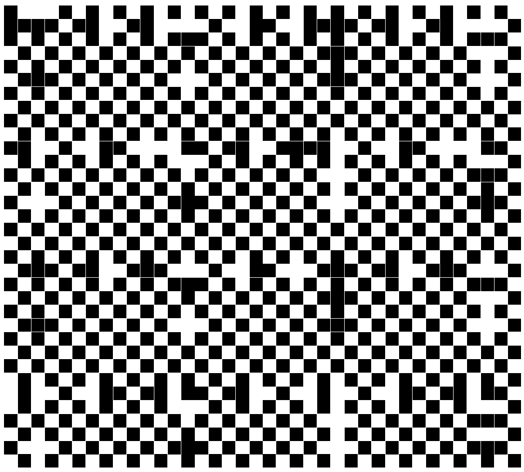
Below only the codebits are shown. This is done by logically exclusively ORing the coded pattern with a normal chessboard pattern with the size of the mask. The origin 0,0 is in the lower left corner in this case.

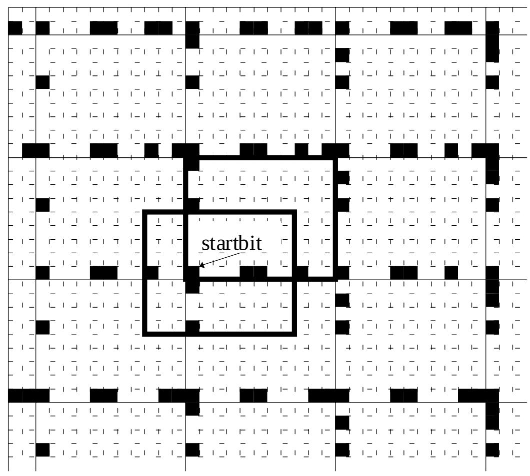
One could also define the origin in the center of the mask or anywhere else. In that case the negative coordinates are coded in two's complement.

Decoding

To find the location of the mask's origin, or better the projection of it, in respect to the camera's sensor position, is a process of reverse refinement. First an acurate fit is done to find the chessboard pattern's position, magnification and orientation. The inverted locations can than be found. From the locations of the codelines in respect to the sensor's center (for example) the offset in dots is determined. Finally the position code of this startbit (crossing of the codelines) is determined.

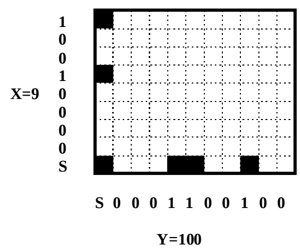
Below a piece of a mask is shown, with B3’s of 11x9. Next to it, only the recovered codebits and some guidelines are shown.



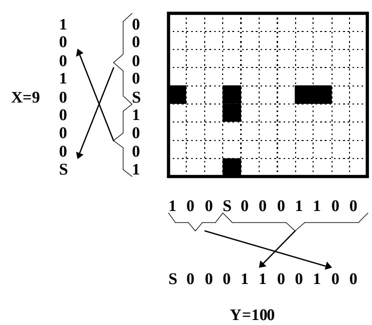


Analyzed are two methods for recognition of the position code. A pattern of 11*9 fields is needed to find the code. One method uses a complete B3. The other method uses the same size but does not include a complete B3. The examples include the same startbit, or B3 basepoint. For easy recognition only the codebits are shown.

In this case the coordinate of the B3 can be found by searching a 11*9 pattern, with codebits in the bottom line and left edge. The code can be read directly.



In the second case one just takes an arbitrary pattern of 11x9. The codelines are found by searching the horizontal and vertical lines that contain more than one codebit within the size of one B3. Then we know the location of the startbit in the pattern seen by the camera. Since the vertical code -in the horizontal line- is the same in the two partially seen B3’s we can merge the partial codes. This is shown here. The same goes for the horizontal code.



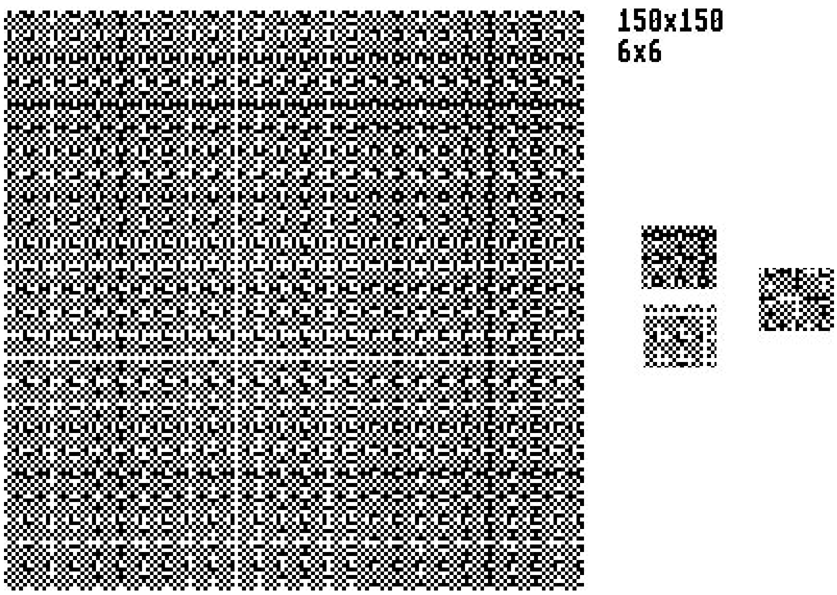
So we do not have to see a complete B3 to find its coordinate. This allows a design of the mask with fewer codelines and thus large undisturbed chessboards. The number of bits in the code then also increases, allowing for large masks to be coded.

Choosing a basic building block depends on a number of requirements. It should be big enough to contain a sufficient amount of codebits to encode the complete mask. It must be small enough to ensure that a little more than one blocksize of the mask is seen by the camera. Or more than one and a half if redundancy is required.

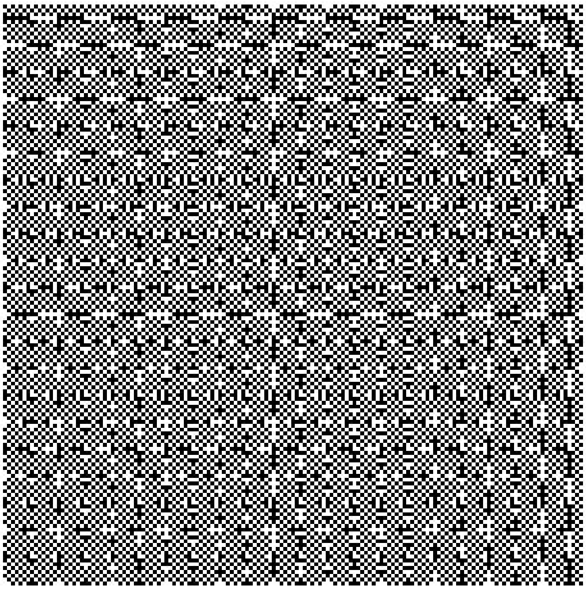
The size of the dots depends on the camera used. The image of the mask on the camera will not be completely sharp due to diffraction. In the data generated by the framegrabber, the dots must big enough to be recognized. Recognizing flat tops for black and white in the data all over the view seems good practice. On the other hand, smaller dots will generate more edges to do a precise fit. So the choice of the size of the mask pixels depends on the quality and aperture of the optics, but also on the camera-framegrabber combination.

It is good practice to be able to recognize a codeline twice or more, to compensate for dust, damage etc.. When the orientation of the mask in respect to the camera is relatively fixed and the projection is very sharp, the width to height ratio of a B3 could be about 4/3 for a standard camera. A square B3 is mirror symmetrical over the line X=Y. This allows to make contact copies of the mother mask, without the need to rotate the origin of the copy.

It is good practice to choose an odd numbers of dots for both horizontal and vertical B3 size. In that case the resulting pattern of the coded locations are inverted every next B3. This results in a more stable average illumination of the camera when scanning over the mask. Also completely black or white codelines are avoided. This is illustrated below. The mask sizes 150x150 dots. The B3 is 6x6. The 'cameraview' shows a little more than two B3s in both directions. Two views (next to the mask) show a clear difference in dot density. The view next to them is a sample of the same size of a mask with B3s of 7x7. This mask is shown on the next page.



Example



150x150
7x7

A 150x150 mask with B3s with an odd number of fields in X and Y direction gives a more stable average illumination of the camera when scanning over the mask. Also completely black or white codelines are avoided.