Req 5.7.1

Title:

Camera focus/tilt

Objective:

Determine and verify the camera focus.

In a series of exposures both M2 will be moved and charge will be moved over a single chip, so that only one read-out is nessecary (TS 3.3.12).

Verify once for each filter that they have the same optical thickness (15mm physical thickness). Do this by measuring the "filter focus offset".

The tilt of the detector plane with respect to the focal plane and its dependency on the orientation of the telescope shall be determined both from:

CalFile– 554 *PSF anisotropy*
and from the matrix of best focus positions provided by the present requirement.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

verify focus: CP and RP. Filter thickness and tilt only once during comissioning.

Outputs:

for filter offsets:
-focus offset values to be transferred to INS data base

-Conformance flag for optical thickness of all filters

For tilt during CP:
-Tilt value

Estimated time needed:

Observation:
- Focus offset and tilt: 2 hours during CP
- Verification filter focus offset 10-30 min/filter (Commissioning)

Reduction: 1 min/CCD.

Priority:

essential

TSF:

TSF– OCAM_img_tec_focusseqwhich stores data for one chip. For the special occasion of the tilt determination data of all chips will be stored (a trivial mod).

Recipe:

Recipe– Focus Standard Paranal focus MIDAS procedure

Needed functionality:

tplTFocus.prg
standard MIDAS

CA:

Verification of focus:
this is an on-line activity which is performed using the RTD and the standard Paranal MIDAS focus procedure tplTFocus.prg. This relatively simple MIDAS procedure asks the user to point on the screen at a non-saturated, well exposed star. It will then compute for a sequence of stellar images the optimally focussed images and returns in keywords outputr(1) = sigmax outputr(2) = sigmay outputr(3) = focus. Plots are made and tables are created for further interactive analysis (if wanted).

Standard RTD allows to make cross cut plots and to ”pick” an object interactively for which position, FWHM in X, Y, and other parameters are computed. It can be used to inspect any CCD, e.g., by reading in RTD the complete file that is created after merging of the data from both mosaic halves.

The MIDAS procedure forms the core of the data reduction of all focus determinations. In ”stand alone” mode it provides the optimal focus values, and the values for focus offsets for other filters.

For the tilt determination, which is to be done only once during commissioning, tplTFocus.prg will be put in a loop in a supervising MIDAS command procedure. While running this superviser loop, well exposed stars will be selected over a large number of CCDs (if not all) and this output (best focus value) will be merged into one big MIDAS table. Table will be converted into an image and a big x-y contour plot of best focus values will directly show the tilt. An regression analysis on the table will result in the finally adopted orientation angle and amplitude. Since this procedure is run only once, it will not be automized but entirely interactively processed with the help of a handful of simple MIDAS commands.

CAP:

```
! "@(#) $Id: req571.tex,v 1.30 2004/11/12 08:35:53 danny Exp $"
!# -*- tcl -*-
!#**********************************************************************
!# E.S.O. - VLT project
```

```
!#
!# tplTFocus.prg
!#
!# who        when      what
!# ---------- --------  ---------------------------------------
------
!# alongino   25/10/00  renamed from TC
!#
!#**********************************************************************
!#   NAME
!#   tplTFocus
!#
!#   SYNOPSIS
!#
!#
!#   DESCRIPTION
!#
!#
!#   FILES
!#
!#   ENVIRONMENT
!#
!#   RETURN VALUES
!#
!#   CAUTIONS
!#
!#   EXAMPLES
!#
!#   SEE ALSO
!#
!#   BUGS
!#
!#-------------------------------------------------------------
-----------
!#
!.KEYWORDS
! ------------------------------------------------------
!
```

```
! PARAMETERS:
!    P1: start-x (default = 0.)
!    P2: step-x  (default = 1.)
!
!define/parameter p1 0. N "Enter start-x : "
!define/parameter p2 1. N "Enter step-x : "
!
define/local centin/c/1/6 CURSOR
!
! initialization
define/local focx/d/1/1 0.0
define/local focy/d/1/1 0.0
define/local sigmax/d/1/1 0.0
define/local sigmay/d/1/1 0.0
define/local focus/d/1/1 0.0
define/local aux/d/1/1 0.0
define/local fox/d/1/1 0.0
define/local sign/d/1/1 0.0
create/tab tfocus 15 0 null
!
! setup the graphics window
create/graph
set/graphics default
set/graphics pmode=1
!
! getting the coordinates...
center/gauss {centin} tfocus
if outputi(1) .eq. 0 then
    write/out
    write/out Invalid number of coordinates.
    return
endif
!
set/midas output=no ! we don't want output on the screen...
!
!
! fit-x computation
compute/table tfocus :step = SEQ
```

```
regression/polynomial tfocus :xfwhm :step 2
save/regression tfocus test
comp/regression tfocus :fitx = test(:step)
focx = {p1} + {p2}*({outputd(2)}/(-2.*{outputd(3)})) - {p2}
!
! fit-y computation
regression/polynomial tfocus :yfwhm :step 2
save/regression tfocus test
comp/regression tfocus :fity = test(:step)
focy = {p1} + {p2}*({outputd(2)}/(-2.*{outputd(3)})) - {p2}
!
! compute the scaling for plots (big job ;{ )
!
compute/table tfocus :maxim = MAX(:xfwhm,:yfwhm)
sort/table tfocus :maxim(-)
define/local maxim/r/1/1 {tfocus,:maxim,1}
maxim = {maxim} + 0.2  ! let's give a margin value
compute/table tfocus :minim = MIN(:xfwhm,:yfwhm)
sort/table tfocus :minim
define/local minim/r/1/1 {tfocus,:minim,1}
minim = {minim} - 0.2  ! let's give a margin value
set/graphics yaxis={minim},{maxim}
!
! plot (x,y)-value versus (x,y)-sigma
sort/table tfocus :step
set/graphics stype=6 colour=1 ! cross(x), black
plot/table tfocus :step :xfwhm
set/graphics stype=5 colour=4 ! cross(+), blue
overplot/table tfocus :step :yfwhm
!
! plot of the fit
set/graphics stype=0 ltype=1 ! no symbol, solid line
set/graphics colour=1 ! black
overplot/table tfocus :step :fitx
set/graphics colour=4 ! blue
overplot/table tfocus :step :fity
!
! focus computation
```

```
aux  = {focx} * {focy}
aux  = M$ABS({aux})
fox  = M$ABS({focx})
sign = {focx}/{fox}
focus = {sign} * M$SQRT({aux})
!
! residual computation
compute/table tfocus :residx2 = (:xfwhm - :fitx) ** 2
compute/table tfocus :residy2 = (:yfwhm - :fity) ** 2
statis/table tfocus :fitx
sigmax = outputr(1)
statis/table tfocus :fity
sigmay = outputr(1)
!
set/midas output=yes ! enable the screen output...
write/out
write/out
if mid$sess(16:20) .eq. "94NOV" then
   set/format f12.2
else
   set/format ,f12.2 !for double keywords
endif
write/out "X: BLACK    y: BLUE"
write/out
write/out "FOCUS:  X =  {focx} " "Y =  {focy}  " "MEAN =  {focus}"
write/out
if mid$sess(16:20) .eq. "94NOV" then
   set/format f12.3
else
   set/format ,f12.3 !for double keywords
endif
write/out "sigma-x = {sigmax}   sigma-y = {sigmay}"
set/graphics default

delete/tab tfocus no

!
outputr(1) = sigmax
outputr(2) = sigmay
outputr(3) = focus

return
```