# Documentation on TELOFFSET
(Mosaic (telescope offset) program for use with ALFOSC).
Version 0.1

Jacob Wang Clasen
Nordic Optical Telescope

October 19, 1997

# Contents

# Chapter 1

# Description

The purpose of this program (TELOFFSET) is to provide a command to offset the telescope. This command is made available from the CCD control program BIAS. With this program it is possible to make mosaics of images as well as small offsets between individual exposures.
It's usage in combination with the CCD aquisition program (BIAS) works via a new command: `teloffset`. This command asks the user for the offset size in the cardinal directions of the CCD's rows and columns. Then the TELOFFSET program is called with these parameters. It is not possible to perform other actions within BIAS while TELOFFSET is executing.
TELOFFSET is called with the following parameters (also see 'usage')

- IP-number
  The IP number of the TCS/VME computer.
- Accesscode
  The current accesscode for communicating with the TCS.
- X movement
  The amount to move the telescope in the CCD pixel X direction.
- Y movement
  The amount to move the telescope in the CCD pixel Y direction.

The program now takes action as follows

- Makes a connection to the RPC server specified in argument 'IP-number'. If this does not succeed, the program returns the exit status -2.
- Gets the current field rotation from the telescope and checks if the telescope is being autoguided.
- Calculates the amount to move the guideprobe. Determined as

$$X_{new} = X_{old} - \frac{\Delta_\delta}{0.0073\frac{asec}{step}}$$

$$Y_{new} = Y_{old} - \frac{\Delta_\alpha}{0.0073\frac{asec}{step}}$$

- The new guideprobe position is checked to be within limits.
- Switches autoguiding off.
- Determines the the amount to move the telescope in seconds of arc in a non rotated coordinate system, i.e. a system with the normal ALFOSC field of -90°. The $zeropoint_{field}$ has therefore the value 90.
- Finds the amount to move the telescope in the coordinate system rotated "field" degrees:

$$\Delta_\alpha = \Delta_x \cdot cos(field + zeropoint_{field}) - \Delta_y \cdot sin(field + zeropoint_{field})$$

$$\Delta_\delta = \Delta_x \cdot sin(field + zeropoint_{field}) + \Delta_y \cdot cos(field + zeropoint_{field})$$

- The value of the $\delta$ movement is negated because increasing values of CCD Y corresponds to descreasing declination.
- The telescope is moved.
- The guideprobe is moved.
- Waits until the guideprobe has reached its new position.
- Waits until the telescope is tracking.
- Switches autoguiding on.
- Returns to caller with exit status 0 (zero).

If the offset size is bigger than a certain size, the guideprobe will end outside its limits. It is desirable to be able to make such big offsets, so the program is able to do that.
If the telescope is being autoguided before the program is executed, the actions described above will be taken. If, however the telescope is *not* being autoguided before the offset program is run, the program will do as follows

- Calculates the amount to move the telescope, given the field rotation.
- Moved the telescope.
- Waits for the telescope to be tracking again.

,witch altogether leaves out the autoguider and guideprobe.

# Chapter 2

# Compilation

Change directory (command `cd`) to `~alfosc/src/mosaic-offset`. The directory should contain the files

- `teloffset.c`
- `Makefile`

A number of other files are needed to be linked with TELOFFSET and these should be located in a known directory before proceeding. The files needed are

- hostintf2.o, hostintf2_clnt.o, hostintf2_xdr.o
- gettcs2.h

Edit the makefile to make `RDIR` point to the directory containing above files (on LISA the location is `/software/rpc/hostintf2`.
Type `make teloffset`. The directory will now also contain the file

- `teloffset`

The compilation is completed.

# Chapter 3

# Installation

Copy the file `teloffset` to the `bin` directory where the current BIAS version resides. Currently `~alfosc/bin.v4` (Oct 19. 1997).

# Chapter 4

# Usage

The normal usage of TELOFFSET is hidden internally in BIAS. BIAS makes a system call (system(3)) to call TELOFFSET.
To use the mosaic/offset software from within BIAS, use the command `teloffset`. This command takes two (2) arguments, namely the offset size in the CCD pixel X direction and the offset in the CCD pixel Y direction. An example:

```
teloffset 10 -10
```

This will move the CCD field of view upwards and to the left by 10 arc seconds (when the field is -90 this is 10 asec to the north and 10 asec to the west, but with other fieldrotations the notion of north and west is lost and up and left are more natural).
This command can also be used in BIAS batch runs together with all the other BIAS commands.

The program can also be executed manually from the commandline, but this is normally intended for testing. The full usage of the program is

- `teloffset` IP-number accesscode X-offset Y-offset

,where
IP-number is the IP number of the TCS computer,
accesscode is the currently valid TCS accesscode (command `sh 84` on TCS),
X-offset is the amount to offset is the CCD X direction,
Y-offset is the amount to offset in the CCD Y direction.

Example:

```
teloffset 161.72.5.7 325911657 10 -10
```

The program returns an exit status according to the success of the execution. A number of things can go wrong, and the easiest way to report it back to the caller is by using the exit status. The following values will (can) be returned

- 0
  Normal termination (all ok).
- -1
  Unable to establish RPC connection to TCS.
- -2
  Telescope is not being autoguided.
- -3
  Guideprobe movement will result in X and Y value out of range.
- -4
  Guideprobe movement will result in X value out of range.
- -5
  Guideprobe movement will result in Y value out of range.

# Chapter 5

# Bugs

- The offsets might be inaccurate due to errors in the determination of the scale of the guideprobe. This should not affect the usage of this program.

# Chapter 6

# Program listing.

## 6.1 Main program.

```
#include <stdio.h>
#include <math.h>
#include "/software/rpc/hostintf2/gettcs2.h"
/* Functions used:
   FieldRotationDelta
   MoveRADelta
   MoveDECDelta
   AutoGuideOff
   AutoGuideOn
   MoveGuideprobeDeltaX
   MoveGuideprobeDeltaY
   MoveGuideprobeX;
   MoveGuideprobeY;

   Usage: teloffset IP-number accesscode Xmovement Ymovement

   - IP-number: IP address of the TCS computer.
   - accesscode: Required accesscode.
   - Xmovement: The amount to move the telescope in the CCD pixel X direction
                (in seconds of arc).
   - Ymovement: The amount to move the telescope in the CCD pixel Y direction
                (in seconds of arc).

   Return values:
     0: Normal termination (all ok).
    -1: Unable to establish RPC connection to TCS.
    -3: Guideprobe movement will result in X and Y value out of range.
    -4: Guideprobe movement will result in X value out of range.
    -5: Guideprobe movement will result in Y value out of range.
*/
#define DEBUG TRUE
/*******************************************************************/

main(argc, argv)
int argc;
char **argv;
{
  double Xmovement, Ymovement; /* The amount to offset */
  int OldGuideprobeX, OldGuideprobeY;
  int NewGuideprobeX, NewGuideprobeY;
  int AutoGuiding; /* 1: autoguiding 0: not autoguiding */
  int AutoguidingWasOn = 0; /* 1: Yes it was 0: No it wasn't */
  double field; /* Field rotation */
  double field_off = 90.0; /* Field zeropoint offset (field -90 is zero)*/
  double scalefactor = 0.007153; /* asec pr. ag step */
```

```
  int TCS_accesscode;
  int *FieldRotDeltaRes,
   *RADeltaRes,
   *DECDeltaRes,
   *AutoGuideOffRes,
   *AutoGuideOnRes,
   *GuideprobeDeltaXRes,
   *GuideprobeDeltaYRes,
   *GuideprobeXRes,
   *GuideprobeYRes;
  char *getserver;

  fieldrotdeltatype a, *fieldrotdelta_info = &a;
  radeltatype b, *radelta_info = &b;
  decdeltatype c, *decdelta_info = &c;
  agontype d, *agon_info = &d;
  agofftype e, *agoff_info = &e;
  guideprobedeltaxtype f, *guideprobedeltax_info = &f;
  guideprobedeltaytype  g, *guideprobedeltay_info = &g;
  guideprobextype h, *guideprobex_info = &h;
  guideprobeytype i, *guideprobey_info = &i;
  smallstatustype *telescope;

  getserver      = argv[1];
  TCS_accesscode = atoi(argv[2]);
  Xmovement      = atof(argv[3]);
  Ymovement      = atof(argv[4]);

  fieldrotdelta_info->AccessCode = TCS_accesscode;
  radelta_info->AccessCode = TCS_accesscode;
  decdelta_info->AccessCode = TCS_accesscode;
  agon_info->AccessCode = TCS_accesscode;
  agoff_info->AccessCode = TCS_accesscode;
  guideprobedeltax_info->AccessCode = TCS_accesscode;
  guideprobedeltay_info->AccessCode = TCS_accesscode;
  guideprobex_info->AccessCode = TCS_accesscode;
  guideprobey_info->AccessCode = TCS_accesscode;

  if (InitRPCService(getserver) == NULL) {
   fprintf(stderr,"Unable to contact the TCS!\n");
   exit(-1);
  }

  if ((telescope = GetTelescope()) == NULL) exit(1);
  AutoGuiding = telescope->AutoGuiding;
  OldGuideprobeX = telescope->ActualGuideProbeX;
  OldGuideprobeY = telescope->ActualGuideProbeY;
  field = telescope->ActualFieldRotation;
    fprintf(stdout,"Field rotation: %4.2f\n",field);

  /* Check if autoguiding is on */
  if (AutoGuiding) {
    AutoguidingWasOn = 1; /* Remember that it was on. */

    /* This is the amount to move the guideprobe opposite the tel. */
    /* This is only true if the angle between x,y and alpha,delta is 0 /*
    /* In any other case, calculate what the angle does */
    /* SIGN! */

    /* Sign is correct */
    NewGuideprobeX = OldGuideprobeX - (int)(Ymovement / scalefactor);
    NewGuideprobeY = OldGuideprobeY - (int)(Xmovement / scalefactor);

    /* Check if within limits, but only if we where guiding. */
    if ( ((NewGuideprobeX <= 1100.0) || (NewGuideprobeX >= 225000.0)) &&
((NewGuideprobeY <= 1100.0) || (NewGuideprobeY >= 165000.0)) ) {
fprintf(stdout,"Error: Guideprobe X and Y outside limits.\n");
exit(-3);
```

```
    }
    if ( (NewGuideprobeX <= 1100.0) || (NewGuideprobeX >= 225000.0) ) {
fprintf(stderr,"Error: Guideprobe X outside limits.\n");
exit(-4);
    }
    if ( (NewGuideprobeY <= 1100.0) || (NewGuideprobeY >= 165000.0) ) {
fprintf(stderr,"Error: Guideprobe Y outside limits.\n");
exit(-5);
    }
    /* Swich off autoguiding */
    fprintf(stdout,"Switching off autoguiding...\n");
    if ((AutoGuideOffRes = AutoGuideOff(agoff_info)) == NULL) exit(1);

  }

  radelta_info->RAdelta = Xmovement * cos((field + field_off)*(PI/180.0)) -
                          Ymovement * sin((field + field_off)*(PI/180.0));
  decdelta_info->DECdelta = Xmovement * sin((field + field_off)*(PI/180.0)) +
                            Ymovement * cos((field + field_off)*(PI/180.0));
  /* We need to invert the delta motion because south (DEC--) is y-value++ */
  decdelta_info->DECdelta = -(decdelta_info->DECdelta);

  /* Move the telescope */
  if (DEBUG) {
    fprintf(stdout,"Field: %4.2f\n",field);
    fprintf(stdout,"Commanded telescope movement: alpha: %4.2f   delta: %4.2f\n",
     Xmovement, Ymovement);
  }
  if (DEBUG)
    fprintf(stdout,"Resulting telescope movement: alpha: %4.2f   delta: %4.2f\n",
radelta_info->RAdelta,decdelta_info->DECdelta);

  fprintf(stdout,"Moving the telescope...\n");
  if ((RADeltaRes = MoveRADelta(radelta_info)) == NULL) exit(1);
  if ((DECDeltaRes = MoveDECDelta(decdelta_info)) == NULL) exit(1);


  if (AutoguidingWasOn) {
    if (DEBUG) {
      fprintf(stdout,"Old XY position:\n");
      fprintf(stdout,"           X: %d   Y: %d\n",OldGuideprobeX,OldGuideprobeY);
      fprintf(stdout,"New position in CCD system:\n");
      fprintf(stdout,"           X: %d Y: %d\n",
            NewGuideprobeX, NewGuideprobeY);
    }
    guideprobex_info->DesiredX = NewGuideprobeX;
    guideprobey_info->DesiredY = NewGuideprobeY;

    /* Make the offset */
    fprintf(stdout,"Moving the guideprobe...\n");
    if ((GuideprobeXRes = MoveGuideprobeX(guideprobex_info)) == NULL) exit(1);
    if ((GuideprobeYRes = MoveGuideprobeY(guideprobey_info)) == NULL) exit(1);

    do {
      if ((telescope = GetTelescope()) == NULL) exit(1);
      usleep(500000);
      if (DEBUG) {
        fprintf(stdout,"Waiting for guideprobe to reach TGT...\n");
        fprintf(stdout,"X: %d  Y: %d\n",telescope->ActualGuideProbeX,telescope->ActualGuideProbeY);
      }
    } while ( (telescope->ActualGuideProbeX != NewGuideprobeX) &&
            (telescope->ActualGuideProbeY != NewGuideprobeY) );
  } /* if AutoguidingWasOn */

  do {
    if ((telescope = GetTelescope()) == NULL) exit(1);
    usleep(500000);
    if (DEBUG) {
```

```
      fprintf(stdout,"Waiting for telescope to track...\n");
    }
  } while (telescope->TelescopeModeNumber != 4);

  /* Swich autoguiding back on */
  if (AutoguidingWasOn) {
    fprintf(stdout,"Turning autoguiding back on...\n");
    if ((AutoGuideOnRes = AutoGuideOn(agon_info)) == NULL) exit(1);
  }

  exit(0);
}
```

## 6.2 Makefile.

```
SHELL=/bin/sh
#DF = _CLASSIC_ID_TYPES
DF = __STDC__
CC = gcc
LD = ld
ODIR = .
RDIR = /software/rpc/hostintf2
PDIR = .
#TMPD = /dd/TEMP
CFLAGS = -O -c -D$(DF) #-Wall
LDFLAGS = -lm
#RDIR = .

RPCLIB   = rpcsvc

LINKLIBS = -l$(RPCLIB)

MOSAIC = teloffset

SERVEROBJS = $(RDIR)/hostintf2_svc.o $(RDIR)/hostintf2_xdr.o
CLIENTOBJS = $(RDIR)/hostintf2.o $(RDIR)/hostintf2_clnt.o $(RDIR)/hostintf2_xdr.o

HDRS = $(RDIR)/hostintf2.h $(RDIR)/gettcs2.h

MOSAICOBJS = $(CLIENTOBJS) $(PDIR)/teloffset.o

$(MOSAIC): $(MOSAICOBJS)
$(CC) $(LDFLAGS) -o $@ $(MOSAICOBJS)

$(CLIENTOBJS): $(HDRS)
```

## 6.3 Code inserted into BIAS.

```
##############################################################################
Correction made on 17. October 1997
Subject: Addition of command to do telescope offsets.
Files changed: lisa:~alfosc/src/alfosc_bias.v4/bias.src.new/commands.c
               lisa:~alfosc/src/alfosc_bias.v4/bias.src.new/comstr.c

-------------------------------------------------------------------------
commands.c / void command(void) line 37, line added:
extern int TCS_accesscode;
```

```
commands.c / void command(void) line 97, line added:
double teloffset_X = 0.0, teloffset_Y = 0.0;

commands.c / void command(void) @ line 1032, case added:
case TELOFFSET: {
dl = (double)teloffset_X;
token = strtok(0," ,");
if ( token == NULL ) {
  cgh_double(&dl,"Offset X size: ");
}
else dl = atof(token);
teloffset_X = dl;

dl = (double)teloffset_Y;
token = strtok(0," ,");
if ( token == NULL ) {
  cgh_double(&dl,"Offset Y size: ");
}
else dl = atof(token);
teloffset_Y = dl;

sprintf(xcbuf,"%s/teloffset 161.72.5.7 %d %4.2f %4.2f",
        TCS_accesscode, teloffset_X, teloffset_Y);
system_exec(xcbuf,FG);
} break;
------------------------------------------------------------------------
comstr.c / line: 74, line added:
#define TELOFFSET 98

comstr.c / line: 158, line added:
"teloffset",TELOFFSET,
########################################################################
```