Building Extraction from Digital Elevation Models

U.Weidner
e-mail: *weidner@ipb.uni-bonn.de*

BEX Manual
Version:
February 1995

Printed:

May 11, 1995

Institut für Photogrammetrie
Universität Bonn
Nußallee 15
D 53115 Bonn

Contents

List of Figures

List of Tables

Input parameter	Remarks
input filename	cf. Remark A.1
pixel size in x-direction pixel size in y-direction	cf. Remark A.2
resolution in height	cf. Remark A.3
first label	cf. Remark A.4
output data format	cf. Remark A.5
output filename	cf. Remark A.6
parametric models	cf. Remark A.7
prismatic models	cf. Remark A.8
write grid models	cf. Remark A.10
write data for each model	cf. Remark A.11
write additional data	cf. Remark A.12
show results	cf. Remark A.13

Tab. A-1: General input parameters

A User's Guide

A.1 How to use BEX

BEX needs a control parameter file as given in the next section. Therefore, **BEX** can be used in two different ways:

`bex`
: invokes the creation of a control parameter file *bex.ini*, which has to be edited by the user. If this file already exists, the program asks the user to edit the file.

`bex` *control parameter filename*
: runs the program with the control parameters in the named file.

The following sections describe the input data (Section A.2), the output (Section A.3), and error messages of the program (Section A.4).

A.2 Input Data

Table A-1 shows further input parameters which have not been discussed. An example of a control parameter file is given in the next paragraph.

Format	Remarks	Format	Remarks	Format	Remarks
IPB [50] [51]	 UCHAR FLOAT	ASCII [1]	*number_of_ rows* *number_of_cols* $z(x, y)$ ...	GNU [4]	# *number_of_ rows* # *number_of_cols* z(0,0) ... z(r,0) *empty line* z(1,0) ... z(r,c)

Tab. A-2: File formats

Remark A.1 input filename *specifies the name of the DEM-file. This file has to be in a format of those given in Table A-2.* **MATCH-T** *ASCII-files have to be converted using the programm* **inpho**. *For this, all grid points have to be defined in the* **MACTH-T** *ASCII-file.* •

Remark A.2 *Different sizes are allowed.* •

Remark A.3 *Up to now this parameter is not needed.* •

Remark A.4 *The user can fix the* first label *for the segments.* •

Remark A.5 *Different output formats for the files *.data* and *.build* are possible (cf. Table A-2) It is recommended to use the format IPB-FLOAT (51).* •

Remark A.6 output filename *specifies the file name for the output files. Extensions are set automatically. See Table A-3, Table A-4 and Table A-5.* •

Remark A.7 *Option* parametric models*: Inputs are*

0 *no parametric models*
1 *compute parametric models*

For output files *see Table A-3.* •

Remark A.8 *Option* prismatic models*: Inputs are*

0 *no prismatic models*
1 *compute prismatic models*

If only prismatic models *are selected, the programm searches for existing files*

- *.bound_box_dat
- *.valid_label_dat
- *.int_seg
- *.int_seg_lines

If these files exist in the current directory, only the ground plan extraction is performed without new detection of buildings. For output files *see Table A-4.* •

Remark A.9 *Pure detection of buildings is possible using 0 for the control parameters* parametric models *and* prismatic models. •

Remark A.10 *Option* write grid models*: Inputs are*

0 do not write grid models
1 write grid models

This option is only valid for parametric models. •

Remark A.11 *Option* write data for each model*: Inputs are*

0 do not write data for each model
1 write data for each model

This option is valid for both groups of models. •

Remark A.12 *Option* write additional data*: Inputs are*

0 do not write additional data
1 write additional data

For output files *see Table A-5. Some of the files are only written, if the corresponding option for the models is set.* •

Remark A.13 *Option* show results*: Inputs are*

0 do not show (some of) the results
1 show (some of) the results

The results are shown using visualization tools of IPB. During visualization the program is stopped. •

Example of Control Parameter File

```
# control parameter file BEX

flat_05.dhm # data
    0.5     # pixel_size in x-direction [m]
    0.5     # pixel_size in y-direction [m]
78909.0     # x-coordinate (lower left corner) [m]
44981.0     # y-coordinate (lower left corner) [m]
   50.0     # expected size of buildings [m^2]
  500.0     # area min/max filter
    4.0     # height for initial segmentation
    3.0     # minimal slope [degree]
    0.1     # resolution in height
    0.5     # threshold for polygon (triangle height) [m]
  100       # first label
    2       # additional margin for bounding box [pixel]
   51       # output data format for grid models of buildings
bextestf    # name (without extension)
    1       # parametric models
    1       # prismatic models
    1       # write grid models
    1       # write data for each model
    1       # write additional data
    0       # show results
```

A.3 Output Files

Lists of Output Files The output files depend on the control parameters *parametric models*, *prismatic models*, *write grid models*, *write data for each model*, and *write additional data*.

- The output files for *parametric models* are listed in Table A-3.
- The output files for *prismatic models* are listed in Table A-4.
- The option *write grid models* is only valid if *parametric models* are selected. The files *name*.build.*label* contain the grid models for each *label*.
- The option *write data for each model* is valid for both, either *parametric models* or *prismatic models*. The files *name*.data.*label* contain the original data within the bounding box for each *label*.

File name	Contents	Format
name.bound_box_dat	file with bounding box information	A
name.build.*label*	file of the grid model of each *label*	*
name.build_lines	file for the visualization of the extracted models' outlines	IPBU
name.dem_build	file for the visualization of the result: data and grid models of the extracted buildings	*
name.hase	file for HASE-parameters	A
name.int_seg	file for the visualization of the refined segments	**
name.int_seg_bin	file for the visualization of the refined segments	IPBU
name.int_seg_lines	file for the visualization of the refined segments' outlines	IPBU
name.labels	file for the visualization of the labels	**
name.prt	file of output parameters	A
name.segments	file of the initial segmentation	**
name.sel_labels	file for the visualization of the valid labels	**
name.valid_label_dat	file with code number of labels	A

Tab. A-3: Output files: parametric models

- The option *write additional data* is also valid for both groups of models. The output files are listed in Table A-5. The actual files also depend on the options *parametric models* and *prismatic models*.

- The formats marked in the tables are

A	ASCII file
P	POLY file
IPBU	IPB-format using *unsigned character*
IPBF	IPB-format using *float*
*	format is given by the user
**	format is determined by the number of labels, either IPBU or IPBF
***	format of the data file

File name	Contents	Format
name.bound_box_dat	file with bounding box information	A
name.first_poly	file of polygons (vectorization of outlines)	P
name.first_poly.*label*	file of polygon (vectorization of outline) of each *label*	P
name.int_seg	file for the visualization of the refined segments	**
name.int_seg_bin	file for the visualization of the refined segments	IPBU
name.int_seg_lines	file for the visualization of the refined segments' outlines	IPBU
name.int_seg_lines.*label*	file for the visualization of the refined segments' outlines of each *label*	IPBU
name.labels	file for the visualization of the labels	**
name.prt	file of output parameters	A
name.ref_poly	file of refined polygons	P
name.ref_poly.*label*	file of refined polygons of each *label*	P
name.segments	file of the initial segmentation	**
name.sel_labels	file for the visualization of the valid labels	**
name.valid_label_dat	file with code number of labels	A

Tab. A-4: Output files: prismatic models

File name	Contents	Format
name.bound_box	file for the visualization of bounding boxes	**
name.build	file of all grid models	IPBF
name.build_lines	file for the visualization of the extracted models' outlines in the (x, y)-plane	IPBU
name.dem_build	file for the visualization of the result: data and grid models of the extracted buildings	IPBF
name.difference	difference between data and filtered data	***
name.min	minimum filtered data	***
name.minmax	minimum-maximum filtered data	***
name.num_poly.*label*	file of polygons (number)	IPBU
name.poly	file of polygons (straight lines)	P
name.poly.*label*	file of polygon (straight lines) of each *label*	P
name.pose	file for the visualization of the extracted points of gravity and main axes	IPBU
name.sel_data	file containing only the data of the valid labels	***
name.size_of_label	file for the visualization of the initial segments' sizes	IPBF

Tab. A-5: Output files: additional data

Example of Output File

```
BEX report file

data file                              : flat05.dhm
number of rows                         :   481
          cols                         :   481

input variables

pixel size in x direction [m]          :     0.5000
pixel size in y direction [m]          :     0.5000
x-coordinate (lower left corner) [m]   : 78909.0000
y-coordinate (lower left corner) [m]   : 44981.0000
minimal size of buildings [m^2]        :    50.0000
area min/max filter [m^2]              :   500.0000
height for initial segmentation [m]    :     4.0000
minimal slope [degree]                 :     3.0000
resolution in height [m]               :     0.1000
threshold for refined polygon [m]      :     0.5000
first label                            :   100
margin for bounding box [pixel]        :     2

output variables

pixel area [m^2]                   :     0.2500
width for min/max-filter           :    45
last label                         :   139
number of labels                   :    40
number of valid labels             :    20

output for labels


...


label no. 129
non valid label : size =     0.5000 m^2

label no. 130
type of building                   :   200
size of label [pixel]              :  1386
              [m^2]                :   346.5000
bounding box row                   :   323
              number of rows       :    50
```

```
             col                  :   187
             number of cols       :    51
height       min of bobo          :   100.6383
             max                  :   111.2185
             min of label         :   105.9378
             max                  :   111.2185
             of interest          :   105.9284
             mean of segment      :   108.4103
             mean of box-segment  :   101.7229
size of interest segment [pixel]  :  1167
                         [m^2]    :   291.7500
outline length of segment[pixel]  :   172
number of polygons                :     1
number of polygon points          :    17

internal coordinate system
point of gravity r [pixel]        :   347.7673
                 c [pixel]        :   213.5216
orientation  phi [degree]         :    36.3166

object coordinate system
reference point  x [m]            : 79008.2344
                 y [m]            : 45057.8516
                 z [m]            :   101.7229
orientation [degree]              :   143.6834

building     lenght [m]           :    25.4000
             width  [m]           :    13.9000
             height 1 [m]         :     4.2149
             height 2 [m]         :     5.2807
             size   [m^2]         :   353.0600
             volume [m^3]         :  2420.3057
.
.
.

label no. 139
non valid label : bounding box


...
```

A.4 Trouble Shooting

Some of the error messages are given by code numbers, other are printed on the screen:

existing files
bex searches for existing output files of the specified output filename. If these files exist, the user has to move or delete these files. **bex** cannot search for those files (A, P) which correspond to a *label*. Nevertheless, the existance of such a file causes an exit of the program.

memory allocation
bex allocates a lot of memory for the computation, which is also freed again, if the information is not needed any longer. If an allocation of memory fails, the program will give a message on the screen, which indicates, which allocation failed (vector, matrix) and the calling subroutine, and the program will exit.

control parameterfile While reading the control parameterfile, errors may occur. Some of them are indicated by a short message and a code number. These code numbers are given in Table A-6, Table A-7 and Table A-8.

Code	Error reading ...
11	pixel size in x-direction
12	pixel size in y-direction
13	x-coordinate (lower left)
14	y-coordinate (lower left)
21	expected minimum size of building
22	area for min/max filter
23	height for initial segmentation
24	minimal slope
25	resolution in height
29	threshold for refined polyon
31	first label
32	additional margin for bounding box
33	output data format
34	option *parametric models*
35	option *prismatic models*
36	option *write grid models*
37	option *write data for each model*
38	option *write additional data*
39	option *show results*
80	data filename
81	data file
82	output filename

Tab. A-6: Error codes I

Code	Error during plausibility check ...	Allowed values ...
111	pixel size in x-direction	> 0
112	pixel size in y-direction	> 0
113	x-coordinate (lower left)	
114	y-coordinate (lower left)	
121	expected minimum size of building	> 0
122	are for min/max filter	$> sizeofbuilding$
123	height for initial segmentation	> 0
124	minimal slope	> 0
125	resolution in height	≥ 0
129	threshold for refined polyon	> 0
131	first label	> 0
132	additional margin for bounding box	≥ 0
133	output data format not allowed	cf. Table A-2
134	option *parametric models* not allowed	0,1
135	option *prismatic models* not allowed	0,1
136	option *write grid models* not allowed	0,1
137	option *write data for each model* not allowed	0,1
138	option *write additional data* not allowed	0,1
139	option *show results* not allowed	0,1

Tab. A-7: Error codes II

Code	Error during file check ...
801	output file **.prt* exists
802	output file **.hase* exists
803	output file **.first_poly* exists
804	output file **.ref_poly* exists
805	output file **.poly* exists
806	output file **.bound_box_dat* exists
807	output file **.valid_label_dat* exists
888	EOF of control parameterfile not found

Tab. A-8: Error codes III

B Programmer's Guide

B.1 Flow Chart

bex

→ **bex_in**
- task: read control parameter file
- alloc: bex.d
- free: -

if *shortcut*

→ **bex_seg**
- task: compute initial segmentation
  label segments
  compute segment size
  compute bounding box
- by: minimum filtering
  maximum filtering
  computing difference
  thresholding
  computing connected components
- alloc: bex.dmin
  bex.dminmax
  bex.ddif
  bex.seg
  bex.labels
- free: bex.dmin
  bex.dminmax
  bex.ddif
  bex.seg

→ **bex_sob**
- task: compute size of labels
- alloc: -
- free: -

→ **bex_bobo**
- task: compute bounding boxes
- alloc: bex.box_r
  bex.box_c
  bex.box_dr
  bex.box_dc
- free: -

→ **bex_rseg**

task: select valid labels
alloc: bex.valid_label
bex.rlabels
free: -

→ **bex_int**

task: compute refined segmentation
alloc: bex.max_height
bex.min_height
bex.max_height_l
bex.min_height_l
bex.mean_height_l
bex.mean_height_wol
bex.height_of_int
bex.size_of_intseg
bex.intseg
bex.intseg_lines
bex.dint
free: bex.labels
bex.rlabels

if *parametric models*

→ **bex_para**

task: compute parameters for parametric models
alloc: bex.pos_r
bex.pos_c
bex.phi
bex.length
bex.width
bex.ridges_height
bex.eaves_height
bex.base_height
bex.pose
free: bex.pose
bex.dint

if *parametric models*

→ **bex_hou**

task: compute grid models
alloc: -
free: -

if *write data*

→ **bex_data**

task: write data within bounding box for each model
alloc: -
free: -

if *parametric models*

→ **bex_hase**
task: compute parameters for program *HASE*
alloc: bex.x_ref
bex.y_ref
bex.z_ref
bex.ridges_eaves
bex.eaves_base
bex.ori_ref
free: -

else

read files
files: *.bound_box_dat, *.valid_label_dat, *.int_seg, *.int_seg_lines
alloc: bex.box_r
bex.box_c
bex.box_dr
bex.box_dc
bex.valid_label
bex.intseg
bex.intseg_lines

endif *shortcut*

→ **bex_poly**
task: compute polygons
alloc: bex.outline_intseg
bex.ref_poly_nop
bex.ref_poly_num
free: -

→ **bex_out**
task: write output file
alloc: -
free: -

free: bex.intseg
bex.intseg_lines

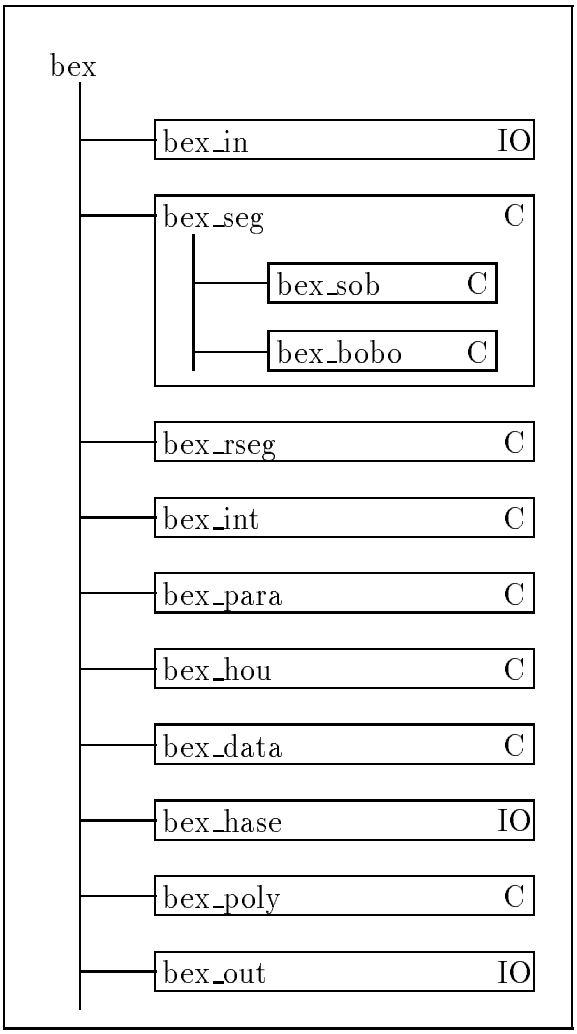

Fig. B-1: Flow chart

B.2 Header File

```
#define BEXSTRUCT static BEXTYP
   typedef
     struct
       {
        /* matrices                                                    */

        /* input:                                                      */

        MATTYP d;              /* data                                 */

        /* output:                                                     */

        MATTYP dmin;           /* MIN(data)                            */
        MATTYP dminmax;        /* MAX(MIN(data))                       */
        MATTYP ddif;           /* Differenz data - MIN(MAX(data))      */
        MATTYP seg;            /* initial segments (is)                */
        MATTYP labels;         /* labels of is                         */
        MATTYP rseg;           /* valid segments (vs)                  */
        MATTYP rlabels;        /* labels of vs                         */
        MATTYP intseg;         /* refined segmantation (rs)            */
        MATTYP intseg_lines;   /* outlines of rs                       */
        MATTYP dint;           /* reduced data                         */

        MATTYP pose;           /* position                             */

        MATTYP gb;             /* gradients (abs)                      */
        MATTYP gw;             /* gradients (ori)                      */
        MATTYP gwv;            /* gradients (var)                      */
        MATTYP hk1;            /* principal curv 1                     */
        MATTYP hk2;            /* principal curv 2                     */
        MATTYP hkw;            /* principal direction direction        */


        /* vectors                                                     */

        /* output:                                                     */

        VECTYP box_r;          /* r position of bounding box [pixel]   */
        VECTYP box_c;          /* c position of bounding box [pixel]   */
        VECTYP box_dr;         /* #rows of bounding box [pixel]        */
        VECTYP box_dc;         /* #cols of bounding box [pixel]        */
        VECTYP size_of_label;  /* size of label (is) [pixel]           */
        VECTYP size_of_intseg; /* size of label (rs) [pixel]           */
        VECTYP max_height;     /* max height in bounding box           */
        VECTYP min_height;     /* min height in bounding box           */
        VECTYP mean_height_wol;/* mean height of background in bounding box   */
        VECTYP height_of_int;  /* thresholds rs                        */
        VECTYP max_height_l;   /* max heights of labels                */
        VECTYP min_height_l;   /* min heights of labels                */
        VECTYP mean_height_l;  /* mean height of labels                */
        VECTYP ridges_height;  /* height of ridges                     */
        VECTYP eaves_height;   /* height of eaves                      */
```

```
VECTYP base_height;    /* height of base                              */
VECTYP length;         /* lenght                                      */
VECTYP width;          /* width                                       */
VECTYP eaves_base;     /* height 1 -> HASE                            */
VECTYP ridges_eaves;   /* height 2 -> HASE                            */
VECTYP x_ref;          /* x-coord. reference point                    */
VECTYP y_ref;          /* y-coord. reference point                    */
VECTYP z_ref;          /* z-coord. reference point                    */
VECTYP ori_ref;        /* orientation in reference system             */
VECTYP area_of_label;  /* Flaeche als robuste Mittelbildung           */
VECTYP pos_r;          /* r-coord. of point of gravity                */
VECTYP pos_c;          /* c-coord. of point of gravity                */
VECTYP phi;            /* orientaion in rc-system                     */
VECTYP eigen1,eigen2;  /* eigenvalues                                 */
VECTYP schiefe;        /* unsymmetry                                  */

IVECTYP valid_label;   /* code number                                 */
IVECTYP outline_intseg;/* number of outline points                    */
IVECTYP ref_poly_num;  /* number of polygons                          */
IVECTYP ref_poly_nop;  /* number of polygon points                    */

/* variables                                                          */

/* input:                                                             */

float size_of_build;   /* expected minimal size of building (m*m)     */
float xo;              /* lower left coord. in reference system       */
float yo;              /* lower left coord. in reference system       */
float xsize_of_pixel;  /* pixel size in x (m)                         */
float ysize_of_pixel;  /* pixel size in y (m)                         */
float area_of_int;     /* area MIN/MAX-filtering                      */

float segh;            /* height for initial segmentation             */
float slope;           /* min slope                                   */
float ridge_eave_h;    /* difference ridge - eave                     */
float height_res;      /* resolution of height                        */
float ref_poly_thresh; /* threshold for refined polygon               */

int boxrand;           /* additional margin for bounding box          */
int first_label;       /* first labels                                */
int param_flg;         /* flag parametric models                      */
int poly_flg;          /* flag prismatic models                       */
int out_flg;           /* flag write additional data                  */
int build_flg;         /* flag write grid models                      */
int data_flg;          /* flag write data                             */
int steppoly;          /* flag shortcut                               */

/* output:                                                            */

float size_of_pixel;   /* size of pixel (m*m)                         */


int last_label;        /* last label                                  */
int number_of_labels;  /* number of labels (is)                       */
```

```
        int number_of_rlabels; /* number of labels (vs)                    */
        int mimafilb;          /* window size MIN/MAX                      */
        int form;              /* format of input data file                */
        int form_md;           /* format for output files *.build *.data   */

        /* filenames                                                       */

        char *dfile;         /* data                                       */
        char *stfile;        /* control parameter                          */
        char *prtfile;       /* output                                     */
        char *bexfile;       /* project                                    */
        char *file;          /* (additional)                               */

        /* else                                                            */

        char *kommentar;     /* data comment                               */
        FILE *fp;            /* output                                     */
        int iniflag;

        int scr_flg;
        char *scr_ipbu;
        char *scr_ipbf;
        char *scr_plot;
        char *scr_cmd;
       }
     BEXTYP;

void bex_in();
void bex_out();
void bex_seg();
void bex_rseg();
void bex_int();
void bex_bobo();
void bex_sob();
void bex_para();
void bex_hou();
void bex_data();
void bex_hase();
void bex_poly();
void bex_scr();
```