Survey Management System

SURPAC CMS PROCESSING

for Stockpiles

SURV-GEN-20020528

Revision: A

Document Owner	**O. Glockner – Principal of G.E.M.S.**		
Date Originated	21 July 2003	Approval Date	
Deployment Date		Archive Date	
Next Revision Date		Printed	

Table of Contents

1 PURPOSE

This procedure describes a way of using an Optec CMS (Cavity Monitoring System) to survey a stockpile, and process the data into a useable format in Surpac software.

2 SCOPE

This procedure applies to:

- Any truck dumps, stockpiles, or pit wall scans.
- Optec CMS equipment.
- Surpac Software v4.1 and 5.0

A certain degree of prior knowledge of Surpac, surveying, and CMS processing is assumed.

3 DEFINITIONS

4 PROCEDURE

Field Routine:

Setup the CMS to get good coverage over the stockpile.

Several setups and scans may be required to reduce any data shadow effects.

CAUTION: The scans can only be done in light or no wind conditions. Rain and dust will also effect the scans.

Survey in the location of the head and boom for each setup, as well as picking up a boundary string around the stockpile base.

Office Routine:

Download the CMS data and convert to dxf file.

Import DXF to a SSI string [DXF2STR].

Consolidate segments.

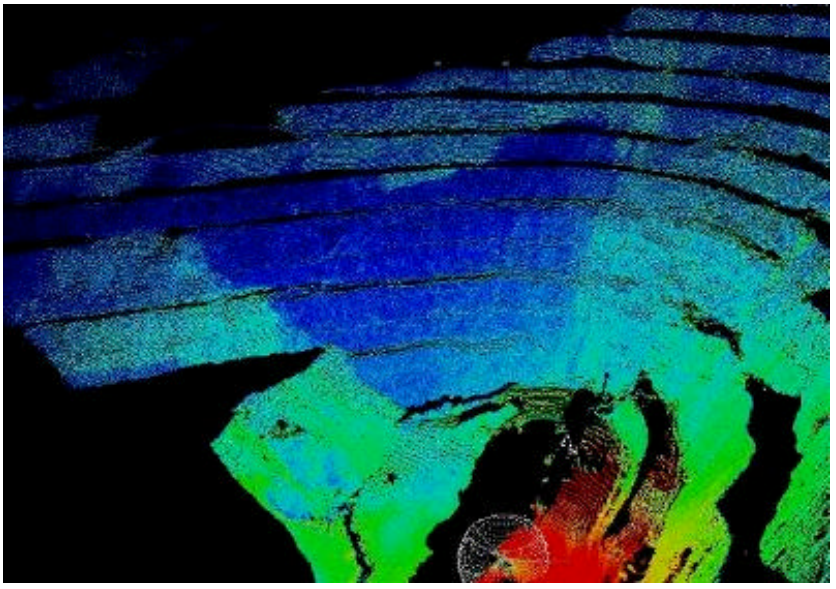
Filter the strings. Usually 0.5m for open pit scan, or 0.3m for stockpiles.

Run disolve.tcl macro … this will take some time.

Save file.

Create boundary files and trim data points outside of these.

Save as a spot height file.



5 REFERENCES

Linda Batemans' Macro	Dissolve.tcl

6 DOCUMENT REVISION HISTORY

Revision Events

Rev.	Author	Changes	Date
A	O Glockner	Initial Concept	May 2002

7 APPENDICES

```
# Macro written by Linda Bateman, August 2001.
# This macro will convert the selected segment into single point segments.
#
# All Description fields are preserved.
# The selected segment will be deleted.
# The new single point segments will have the same string number as the selected segment.
#
# The results generated by this macro should be always be checked by the user.
# Use of this macro in whole or in part by a third party shall be at that users sole risk.
##############################################################################################

SclGetActiveViewport viewport
$viewport SclGetActiveLayer ActiveLayer
$ActiveLayer SclGetStrings strings

# Select the segment you wish to dissolve into single point segments.
set status [SclSelectPoint PointHandle "Select segment to dissolve. Esc to cancel." layer stringid segmentnum pointnum xp yp
zp pointdesc]
if {$status == $SCL_OK} {

# Climb the hierarchy to get a handle on attributes.
  $PointHandle SclGetParent SegmentHandle
  $SegmentHandle SclGetParent StringHandle
  $StringHandle SclGetParent SwaHandle

# Calculate number of points in a segment.
   set point_count [$SegmentHandle SclCountItems]

# Graphics point count starts at 0 so need to add one to correctly report number of points.
  puts "Number of points is [SclExpr $point_count +1]. Please wait while single point segments are created."
  puts "This may take up to several minutes for segments containing more than several thousand points."

   set yindex [$PointHandle SclGetAttributeIndex y]
   set xindex [$PointHandle SclGetAttributeIndex x]
   set zindex [$PointHandle SclGetAttributeIndex z]
   set dindex [$PointHandle SclGetAttributeIndex desc]

  } else {
  puts "User hit Esc key - Macro cancelled."
  return
 }
# Loop through and create a new point at each point in selected segment.
    set i 0
    while  {$i <= $point_count}  {

                                                        #Obtain the coordinates of the first point.
                                                    $SegmentHandle SclGetItem PointHandle $i
                                                set y1 [$PointHandle SclGetValueByIndex $yindex]
                                                set x1 [$PointHandle SclGetValueByIndex $xindex]
                                                set z1 [$PointHandle SclGetValueByIndex $zindex]
                                        set d1 [$PointHandle SclGetValueByIndex $dindex]

# Copy the point.
   $ActiveLayer SclCreateString string [$StringHandle SclGetId]
    $string SclCreateSegment segment [$string SclCountItems]
    $segment SclCreatePoint point [$segment SclCountItems]
    $point SclSetValueByName x $x1
    $point SclSetValueByName y $y1
    $point SclSetValueByName z $z1
    $point SclSetValueByName desc $d1
```

```
   incr i
  }

# Delete the original segment.
set status [ SclFunction "SEGMENT DELETE" {
  select_point=table {winx winy objectx objecty objectz } {
    { "" "" "$xp" "$yp" "$zp" }
  }
}]

# Draw the newly created single point segments.
$string SclDraw
```