Laser-Scan Ltd.

LITES2 - Vignette Generation

User Guide

Science Park, Milton Road, Cambridge, England CB4 4FY tel: (01223) 420414

Document "LITES2 Vignette Generation - User Guide"		Category "User Guide"
Document Issue 1.0	Matt Wenham	20-Nov-1997
Document Issue 1.1	Clarke Brunt	22-Jun-1998
Document Issue 1.2	Clarke Brunt	14-Jul-1998
Document Issue 1.3	Clarke Brunt	17-Jul-1998

1 **Introduction**

This document describes a vignette generation capability using LITES2.

2 **Installation**

The vignette generation software consists of a single LITES2 command file VIGNETTE.LCM which contains LITES2 macro definitions. This file should be in LSL$PUBLIC_ROOT:[LITES2.CMD], and should be executed during LITES2 initialisation. This may be done by adding a command **@VIGNETTE** to either the site-specific or terminal-specific LITES2 initialisation file. Alternatively, the line may be typed at the LITES2 command prompt.

The vignette generation facility is a LITES2 licensed option, and requires licences for the LITES2 region and geometry options in order to function correctly.

3 **Overview**

The LITES2 vignette generation system consists of a set of LITES2 macro commands. Together, they allow the production of a vignette around the outside of a series of coastline features.

The vignette generation algorithm works on sets of associated coastline features within a defined region of interest. A set might consist of a continental coastline and its nearby islands, or a group of islands. Copies are made of all the coastal features, using a dummy feature code. Any open coastline features are closed around the neat line of the sheet.

The features are next buffered to create an outer ring at a specified distance from the features. This outer ring is not part of the finished result. It will normally by further out than the edge of the vignette, producing smoother results the further out it is. The outer ring is then buffered itself, back towards the coastline. This results in a smooth inner ring which is used as the edge of the solid vignette region. The line may be edited if the user wishes, since it is converted into a second type of dummy feature. The solid vignette region is finally buffered itself in order to create the faded region of the vignette.

4 **Preparation**

Some additions need to be made to the relevant FRT file before the vignette generation system may be used. Firstly, two dummy feature codes of graphical type 1 (line) need to be added, one for the copies of the coastal features and one for the outer ring line. These are best given a colour which will make them obvious on the map sheet. These two features may use any spare feature codes.

Secondly, a number of consecutive feature codes of graphical type 12 (fill area) need to be added to produce the vignette itself. The first of these features is used to produce the solid area of the vignette, and should have an appropriate colour. The subsequent features are used to produce the fading part of the vignette area. They should have colours which fade evenly from the colour of the solid vignette to the colour of the background sea regions. This may necessitate extra colours being added to the colour table used for the map.

Finally, in order for the faded region to be drawn correctly, the relevant feature codes must be prioritised using PRIO entries in the FRT file. The feature code for the solid vignette region should have the highest priority, and the subsequent feature codes should be assigned successively lower priorities.

5 **Initialisation**

The vignette generation system is initialised by the macro command **VIGINIT**. This may be followed by up to seven parameters. They should be given in the order listed below, separated by spaces. Each parameter takes the default value shown in brackets when the vignette generation system is first initialised. If a parameter is not specified on the VIGINIT line, its value will not change.

1. **VIG_BUFFER_DIST** (50) - the distance out from the coastal features at which the outer ring is generated.

2. **VIG_SOLID_DIST** (10) - the minimum distance out from the coastal features of the inner ring.

3. **VIG_FADE_DIST** (10) - the distance in which the vignette fades out.

4. **VIG_START_FC** (10002) - the feature code of the solid vignette area.

5. **VIG_NO_FCS** (10) - the total number of real vignette feature codes. This figure does not include the two dummy feature codes, but does include that used for the solid region.

6. **VIG_COAST_FC** (10000) - the feature code used for the closed copies of the coastal features.

7. **VIG_REGION_FC** (10001) - the feature code used for the inner ring region, and for establishing the region of interest.

After initialisation, any of these parameters may be changed by using the LITES2 command LET <PARAMETER_NAME> = <VALUE>.

As well as establishing the parameters, VIGINIT also enables substitution and feature sorting by priority, both of which are needed by the vignette generation system.

The ratio between VIG_BUFFER_DIST and VIG_SOLID_DIST determines the smoothness of the inner ring generated. Typically, VIG_BUFFER_DIST should be 3-10 times VIG_SOLID_DIST.

6 **Neat Line Establishment**

The vignette generation system needs to know the position of the neat line of the map in order to work correctly. The position of the neat line is established using the macro command **GET_NEAT**. This command may be used having first found a map feature which corresponds to the neat line. Alternatively, if no feature is found when the command is invoked, the user may enter the extents of the neat line manually.

7 **Coastline Checking**

The next task the user should perform is that of selecting which features correspond to coastline. This should be done using the LITES2 command SELECT and may be done using one of more of MAPS, LAYERS and FCS. Once only the coastal features are selected, the selection should be saved into a macro using the command **SAVE SELECTIONS _VIG_SEL**. This macro is used by other vignette generation macro commands. If the same feature code is used for inland shore features as for coastline, these features may be left selected. They are filtered out at a later stage.

All of the coastline features should lie inside the neat line, and either be closed (i.e. islands) or begin and end on the neat line as established by the GET_NEAT command. Additionally, the features should not self-intersect. The macro command **VIGCHECK** checks that all these conditions are met for the coastline features selected above. It will report the FSNs of any coastal features which do not meet the above criteria. When the command has checked all the features, it will draw all those features with which problems were found. If no problems were found, it will make no report and draw all of the coastal features.

If VIGCHECK encounters an open coastal feature which has an end less than 0.05 IFF units away from the neat line, it will ask the user whether the end of the feature should be moved onto the neat line. All other coastal features on which VIGCHECK reports will need to be mended manually. If you do not wish to change the existing coastal features, make copies of them using a free feature code or layer, and save those to the _VIG_SEL macro before using VIGCHECK. Do not use either VIG_COAST_FC or VIG_REGION_FC for these copies.

8 **Coastline Closing**

The macro command **VIGCLOSE** should be used next. This makes copies of all the coastal features, closing the open features around the neat line previously established.

When closing the open features, they may be closed either way around the neat line. For features which begin and end on the same edge of the map sheet, VIGCLOSE will ask whether the feature is to be closed the 'short' way around the neat line, i.e. straight between the two end points. You may, of course override this and close the feature the 'long' way around the neat line. A similar facility exists for open features which start and end on adjacent edges of the map sheet, the 'short' closure passing through the corner at which the two edges meet.

For both of these types of feature, you may specify that they should always be closed the 'short' way around by using the command **VIGCLOSE Y**. For open features which begin and end on opposite sides of the map sheet, you will always be asked which way around the feature will be closed. This is because it is much more difficult to establish the 'short' closure direction with this type of feature.

9 Coastline Selection

As mentioned above, each section of coastline and its associated off-shore islands should be treated separately. Groups of large islands and their associated islets should also be treated separately. Each separate region of interest should be defined using the **VIGREGION** macro command.

A feature of FC VIG_REGION_FC should first be digitised to enclose all the features in a particular region of interest. This feature may pass outside the neat line. When VIGREGION is then used, the nearest feature of FC VIG_REGION_FC will be searched for. Its outline will be stored, and the feature itself will be deleted. The region of interest is stored using the LITES2 region facility, in REGION 1. The current region of interest may therefore be redrawn using the command DRAW REGION 1.

This step is optional. If there is no need to select a particular region, then just ensure that REGION 1 is not defined, and the subsequent commands will use all the coastline features defined by _VIG_SEL.

Note that if selection by region is used, the desired coastal features **including** their closing lines must lie completely within the region. If this is not possible, due perhaps to closing lines which enclose most of the sheet, then a better approach is to copy the coastal features in the region of interest to a spare feature code or layer, then select just these features and perform the entire vignetting flowline on them, finally deleting them again before moving on to another region.

10 Coastline Buffering

The inner vignette line corresponding to all the coastal features which lie entirely within the region of interest defined as above should be created next using the macro command **VIGINNER**. This command will first ask the user to place the cursor in the sea on the map in order that it may ascertain which if any coastline features lie inland, i.e. are lakes. If any such features are encountered, it will ask for confirmation that they are lakes. You can override this question being asked, effectively always answering "YES" by using the command **VIGINNER Y**.

Once the features have all been accounted for, the position of the inner ring will be calculated. This may take some time to complete.

First the outer ring at VIG_BUFFER_DIST is created. This is then filtered to remove excessive points. On occasions, the filter operation may cause the features to self-intersect. If this happens, the procedure will stop after writing some messages about the problem. The user may then edit the outer ring to fix the problem (taking care not to corrupt any connecting invisible lines). The VIGINNER command should then be repeated. It will carry on where it left off, again checking the outer ring for self-intersection.

Once the outer ring is satisfactory, the procedure will buffer it back towards the coastline, to produce the solid vignette boundary at VIG_SOLID_DIST. One or more features of FC VIG_REGION_FC are created marking the boundary of the inner ring. These are again filtered. These features may be modified in order to alter the shape if the eventual vignette, but should not be modified anywhere they fall outside of the neat line.

VIGINNER makes use of the LITES2 geometries 1 to 6. Any information contained in these geometries before using VIGINNER will be lost.

11 **Vignette Creation**

The vignette may finally be created using the **VIGCREATE** macro command. This searches for all features of FC VIG_REGION_FC, and uses them as the basis for the solid region of a new vignette. It first checks that these features do not self-intersect. If this happens, the procedure will stop after writing some messages about the problem. The user may then edit the inner ring to fix the problem (taking care not to corrupt any connecting invisible lines). The VIGCREATE command should then be repeated. Additional information concerning the position of the lakes and land is passed from VIGINNER to VIGCREATE in the LITES2 geometries 1 and 2. These should therefore not be altered between using these two macro commands. VIGCREATE itself uses geometries 1 to 6, so any information contained therein will be lost.

Since the vignette features are created in reverse priority order, they should be re-drawn once the command has finished to show the vignette as it will appear.

VIGCREATE also produces features filling in the lakes established by VIGINNER. By default, these are created as FC VIG_START_FC. To use a different feature code, use **VIGCREATE <LAKE_FC>**. To suppress the creation of the filled lake features, make <LAKE_FC> negative.

12 **Removing Dummy Features and Unwanted Vignettes**

Once the vignette for a particular region of interest has been finalised, the inner ring dummy features should be removed before moving on to the next region. This may be done using the macro command **VIGCLEAN**. It removes all features of FC VIG_REGION_FC and then re-draws the screen.

Once all the vignettes have been finalised, the closed dummy coastline features may be removed by using **VIGCLEAN FULL**. This removes all features of FCs VIG_REGION_FC and VIG_COAST_FC.

The macro command **VIGDELETE** will remove all vignette features which lie entirely within the current region of interest. This command may be used to quickly remove a vignette which is not satisfactory. The vignette features may have invisible lines which pass outside the current region of interest. If this is the case they will need to be deleted manually, or the region of interest redefined to enclose the invisible lines.

13 **Vignette Generation Across LITES2 Sessions**

If you wish to stop and re-start LITES2 during a vignette generation session, this is best done between dealing with two regions of interest. However, you may wish to break the LITES2 session while editing an outer or inner region feature, i.e. while using VIGINNER, or in between using VIGINNER and VIGCREATE. If this is done, the information contained in geometries 1 and 2 concerning the position of the land and water areas will be lost. An extra macro command, **VIGLANDWATER**, re-establishes this information, and should be used as follows:

1. Start the LITES2 session as normal, ensuring that @VIGNETTE has been executed at some point.

2. Load the IFF data as normal. The outer or inner ring feature(s) (with feature code VIG_REGION_FC) should be present, since they were saved with the rest of the map data.

3. Re-establish the neat line using GET_NEAT.

4. If required, re-establish the region of interest by digitising a suitable feature and using VIGREGION.

5. Use VIGLANDWATER to re-establish the positions of island and lake features within the region of interest. **VIGLANDWATER Y** will suppress the confirmation of water features as for VIGINNER.

6. Continue editing the ring features, and use VIGINNER (if you were editing an outer ring) or VIGCREATE (if working on inner rings) as usual when finished.

14 **Additional Notes**

1. If you do not wish to choose a region of interest for VIGINNER and VIGCREATE, then omit using VIGREGION completely (and do CANCEL REGION 1 if there is any chance that region 1 already exists).

2. Using VIGREGION to select the region of interest does not work unless a region can be constructed which completely encloses all the features of interest (**including closing lines round the neatline**), whilst not enclosing any which not wanted. You can check that you have the correct features by selecting the closed coastline features (VIG_REGION_FC), and also doing SELECT INREGION 1, then drawing the selected features. If a feature appears to have been missed out of the vignetting operation, then check that it (and its closing lines round the neatline) lie within the region.

   Some charts, for example those with several independent water inlets, cannot be vignetted either in their entirety or by selecting by region, since in either case there are areas which claim to be both land and sea at the same time. In this case, there is no choice but to makes copies of selected coastline features using either a spare feature code or layer, and perform the entire flowline, from VIGCHECK to VIGCREATE, on just these features.

3. If the generation of buffers fails after several instances of the message "Failed to generate buffer. Trying again with vector length...", followed by "Too few points around circle arcs. Alter with TOLERANCE CIRGEN command", then as the message says, change the TOLERANCE CIRGEN parameters to put more points in generated circle arcs. LITES2's HELP on GEOMETRY BUFFER says more about this.

4. If the vignette procedure claims that a feature self-intersects, it will already have displayed a message about the problem. Some of these messages come from LITES2's geometry handling commands, and references to `inner' and `outer' may refer to inner and outer rings of geometries, rather than referring to the vignettes. The coordinates given in these messages may not accurately identify where the self-intersection is, but just identify the particular ring which contains it. You can write out a copy of the data and use ICHECK to find the problem. Note that if you want ICHECK to check layer 0, then it must be specified using the /LAYER qualifier. If there are intersections which ICHECK cannot find, then try using the /TOUCH qualifier with a suitably small tolerance, as this will find more `borderline' cases.

5. The vignetting procedure uses LITES2's FILTER command internally at two stages (both during VIGINNER). First the outer ring is filtered, then the inner ring (the solid vignette boundary). The current settings for TOLERANCE BUNCH are used for these operations, so the user should ensure that these are sensible. Different settings may be tried if the filter operation is frequently introducing self-intersections into area features.

   When making manual edits to remove self-intersections, either in VIGINNER on the outer ring, or in VIGCREATE on the inner ring, then great care should be taken not to spoil the coincident invisible lines which will connect any `holes' to the outer boundary or their area. These must remain precisely coincident if the feature is to be used subsequently as a geometry. It is best not to edit these line segments or their end points at all, confining all edits to points with visible lines on either side.

6. If automating the process of vignetting by embedding the commands in macros, then when doing SAVE SELECTIONS _VIG_SEL, care should be taken that macro _VIG_SEL does not already exist. If it does, it should be cancelled first, using CANCEL MACRO _VIG_SEL. This will cause an error if the macro doesn't exist. You can either ignore the errors by using ENABLE CONTINUE, or else avoid them by testing whether the macro exists first, for example:

   ```
   ASK STATUS MACRO _VIG_SEL
   TEST $ASK_INT1=0
   ELSE CANCEL MACRO _VIG_SEL
   ```

7. The LITES2 command LOOP (which is short for LOOP NORMAL, as opposed to LOOP EXTEND) is the easiest way of precisely closing an existing feature. It moves the first and last points to their mean position, which is the same behaviour as MEND at breaks in features.

15 **PostScript Output of Vignettes**

If you wish to produce Postscript output of a chart including vignettes, you may need to alter the value of **LSL$PS_MAXPATH**. If using a PostScript device of greater than level 1, the value of LSL$PS_MAXPATH may be increased to avoid overly large PostScript output files. On any PostScript device, it will also be necessary to define **LSL$PS_FILL_INVIS** to 1 so that invisible line segments of the vignettes are plotted correctly. See the PLOTTING PostScript guide for further details concerning these two logical names.