Software

Sampling strategies in a forest environment for the elaboration of Isoscapes.

Supplementary File.

Marco Ciolfi,[a] Francesca Chiocchini,[a] Giuseppe Russo[a] Luciano Spacchino,[a] Michele Mattioni,[a] Mauro Lauteri.[a]

```
package webnodes;

import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Toolkit;
import javax.swing.JFrame;
import javax.swing.JTabbedPane;
import org.eclipse.wb.swing.FocusTraversalOnArray;
import java.awt.Component;
import java.awt.datatransfer.Clipboard;
import java.awt.datatransfer.StringSelection;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JPanel;
import javax.swing.JComboBox;
import javax.swing.JTextField;
import javax.swing.JLabel;
import javax.swing.SwingConstants;
```

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JCheckBox;
import javax.swing.JButton;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;


public class WebNodes {

	private static double SQUEEZE = Math.sqrt(.5d);

	private static String DEFAULT_FILE_NAME = "nodes.csv";

	private JFrame frame;
	private JTabbedPane mainPanel;
	private JTextField squareZeroX =  new JTextField();
	private JTextField squareZeroY =  new JTextField();
	private JTextField webZeroX =  new JTextField();
	private JTextField webZeroY =  new JTextField();

	private JComboBox<Integer> squareDivNum;
	private JComboBox<Integer> squareDivSize;
	private JComboBox<Integer> webDivNum;
	private JComboBox<Integer> webDivSize;
	private JCheckBox webDiag;

	private JButton btnSquareGenerate = new JButton("GENERATE");
	private JButton btnWebGenerate = new JButton("GENERATE");
	private JButton btnExport;

	private JTextArea nodesList;

	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
```

```
                        try {
                                WebNodes window = new WebNodes();
                                window.frame.setVisible(true);
                        } catch (Exception e) {e.printStackTrace();}
                }
        });
}


public WebNodes() {
        initialize();
}


private void initialize() {

        frame = new JFrame();
        frame.setResizable(false);
        frame.setBounds(100, 100, 350, 300);
        frame.getContentPane().setBackground(new Color(245, 245, 245));
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);

        mainPanel = new JTabbedPane(JTabbedPane.TOP);
        mainPanel.setBounds(6, 6, 340, 266);
        frame.getContentPane().add(mainPanel);

        JPanel squarePanel = new JPanel();
        initializeSquarePanel(squarePanel);
        mainPanel.addTab("SQUARE", null, squarePanel, "create square nodes lattice");
        mainPanel.setEnabledAt(0, true);
        squarePanel.setLayout(null);

        JPanel webPanel = new JPanel();
        initializeWebPanel(webPanel);
        mainPanel.addTab("SPIDER WEB", null, webPanel, "create web nodes lattice");
        mainPanel.setEnabledAt(1, true);
        webPanel.setLayout(null);

        JPanel nodesPanel = new JPanel();
        initializeNodesPanel(nodesPanel);
        mainPanel.addTab("NODES", null, nodesPanel, "show/export nodes");
        mainPanel.setSelectedIndex(1);
```

```
        mainPanel.setEnabledAt(2, false);
        mainPanel.setForegroundAt(2, Color.GRAY);

        frame.getContentPane().setFocusTraversalPolicy(new FocusTraversalOnArray(new Component[]{mainPanel}));

}

private void initializeWebPanel(JPanel webPanel){
        JLabel lblWebZeroX = new JLabel("ZERO-X");
        lblWebZeroX.setHorizontalAlignment(SwingConstants.RIGHT);
        lblWebZeroX.setBounds(27, 25, 61, 16);
        webPanel.add(lblWebZeroX);

        JLabel lblWebZeroY = new JLabel("ZERO-Y");
        lblWebZeroY.setHorizontalAlignment(SwingConstants.RIGHT);
        lblWebZeroY.setBounds(27, 53, 61, 16);
        webPanel.add(lblWebZeroY);

        webZeroX = new JTextField();
        webZeroX.getDocument().addDocumentListener(new DocumentListener() {
                public void insertUpdate(DocumentEvent de) { interceptDouble(webZeroX, webZeroY, btnWebGenerate); }
                public void removeUpdate(DocumentEvent de) { interceptDouble(webZeroX, webZeroY, btnWebGenerate); }
                public void changedUpdate(DocumentEvent de) { interceptDouble(webZeroX, webZeroY, btnWebGenerate); }
        });
        webZeroX.setToolTipText("x-coordinate of origin (utm m)");
        webZeroX.setHorizontalAlignment(SwingConstants.RIGHT);
        webZeroX.setBounds(100, 19, 134, 28);
        webZeroX.setColumns(10);
        webZeroX.setText("0");
        webZeroX.setColumns(10);
        webPanel.add(webZeroX);

        webZeroY = new JTextField();
        webZeroY.getDocument().addDocumentListener(new DocumentListener() {
                public void insertUpdate(DocumentEvent de) { interceptDouble(webZeroY, webZeroX, btnWebGenerate); }
                public void removeUpdate(DocumentEvent de) { interceptDouble(webZeroY, webZeroX, btnWebGenerate); }
                public void changedUpdate(DocumentEvent de) { interceptDouble(webZeroY, webZeroX, btnWebGenerate); }
        });
        webZeroY.setToolTipText("y-coordinate of origin (utm m)");
        webZeroY.setHorizontalAlignment(SwingConstants.RIGHT);
        webZeroY.setBounds(100, 47, 134, 28);
        webZeroY.setText("0");
        webZeroY.setColumns(10);
```

```
webPanel.add(webZeroY);

JLabel lblWebDiv = new JLabel("SIZE");
lblWebDiv.setToolTipText("nodes lattice cell size (m)");
lblWebDiv.setHorizontalAlignment(SwingConstants.RIGHT);
lblWebDiv.setBounds(27, 88, 61, 16);
webPanel.add(lblWebDiv);

webDivSize = new JComboBox<Integer>();
webDivSize.setToolTipText("nodes lattice cell size (m)");
webDivSize.setModel(new DefaultComboBoxModel<Integer>(new Integer[] {1, 2, 3, 4, 5, 10, 20, 50, 100, 500, 1000}));
webDivSize.setSelectedIndex(0);
webDivSize.setBounds(100, 84, 134, 27);
webPanel.add(webDivSize);

JLabel lblWebNum = new JLabel("NUM");
lblWebNum.setToolTipText("number of lattice nodes in each direction");
lblWebNum.setHorizontalAlignment(SwingConstants.RIGHT);
lblWebNum.setBounds(27, 116, 61, 16);
webPanel.add(lblWebNum);

webDivNum = new JComboBox<Integer>();
webDivNum.setToolTipText("number of lattice nodes in each direction");
webDivNum.setModel(new DefaultComboBoxModel<Integer>(new Integer[] {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 50, 100}));
webDivNum.setBounds(100, 112, 134, 27);
webPanel.add(webDivNum);

JLabel lblWebNESW = new JLabel("N E S W");
lblWebNESW.setToolTipText("horizontal and vertical directions");
lblWebNESW.setHorizontalAlignment(SwingConstants.RIGHT);
lblWebNESW.setBounds(27, 151, 61, 16);
webPanel.add(lblWebNESW);

webDiag = new JCheckBox("NE SE SW NW");
webDiag.setToolTipText("optional diagonal drections");
webDiag.setSelected(true);
webDiag.setBounds(100, 147, 128, 23);
webPanel.add(webDiag);

btnWebGenerate.addMouseListener(new MouseAdapter() {
        public void mouseClicked(MouseEvent e) {
                if(btnWebGenerate.isEnabled()){
                        int ord = 0;
```

```
                        double diagshift;
                        double x = Double.parseDouble(webZeroX.getText());
                        double y = Double.parseDouble(webZeroY.getText());
                        double shift = Double.parseDouble(webDivSize.getSelectedItem().toString());
                        StringBuffer sb = new StringBuffer();
                        sb.append("ORD,ID,UTM_X,UTM_Y\n");
                        sb.append(++ord + ",ZERO," + clean(x) + "," + clean(y) + "\n");
                        for(int i=1; i<=Integer.parseInt(webDivNum.getSelectedItem().toString()); i++){
                                diagshift = i * shift * SQUEEZE;
                                sb.append(++ord + "," + "N" + i + "," + clean(x) + "," + clean(y+i*shift) + "\n");
                                if(webDiag.isSelected()) sb.append(++ord + "," + "NE" + i + "," + clean(x+diagshift) + "," + clean(y+diagshift) + "\n");
                                sb.append(++ord + "," + "E" + i + "," + clean(x+i*shift) + "," + clean(y) + "\n");
                                if(webDiag.isSelected()) sb.append(++ord + "," + "SE" + i + "," + clean(x+diagshift) + "," + clean(y-diagshift) + "\n");
                                sb.append(++ord + "," + "S" + i + "," + clean(x) + "," + clean(y-i*shift) + "\n");
                                if(webDiag.isSelected()) sb.append(++ord + "," + "SW" + i + "," + clean(x-diagshift) + "," + clean(y-diagshift) + "\n");
                                sb.append(++ord + "," + "W" + i + "," + clean(x-i*shift) + "," + clean(y) + "\n");
                                if(webDiag.isSelected()) sb.append(++ord + "," + "NW" + i + "," + clean(x-diagshift) + "," + clean(y+diagshift) + "\n");
                        }
                        nodesList.setText(sb.toString());
                        mainPanel.setEnabledAt(2, true);
                        mainPanel.setForegroundAt(2, Color.BLACK);
                        mainPanel.setSelectedIndex(2);
                }
        }
});
btnWebGenerate.setToolTipText("generate spider web lattice nodes");
btnWebGenerate.setBounds(17, 179, 134, 29);
webPanel.add(btnWebGenerate);

JButton btnWebReset = new JButton("RESET");
btnWebReset.addMouseListener(new MouseAdapter() {
        public void mouseClicked(MouseEvent e) {
                webZeroX.setText("0");
                webZeroY.setText("0");
                webDivNum.setSelectedIndex(0);
                webDivSize.setSelectedIndex(0);
                webDiag.setSelected(true);
                btnWebGenerate.setEnabled(true);
                mainPanel.setEnabledAt(2, false);
                mainPanel.setForegroundAt(2, Color.GRAY);
                nodesList.setText("");
        }
});
```

```
        btnWebReset.setToolTipText("reset all fields");
        btnWebReset.setBounds(179, 179, 134, 29);
        webPanel.add(btnWebReset);
}

private void initializeSquarePanel(JPanel squarePanel){
        JLabel lblSquareZeroX = new JLabel("ZERO-X");
        lblSquareZeroX.setToolTipText("x-coordinate of origin (utm m)");
        lblSquareZeroX.setHorizontalAlignment(SwingConstants.RIGHT);
        lblSquareZeroX.setBounds(27, 25, 61, 16);
        squarePanel.add(lblSquareZeroX);

        JLabel lblSquareZeroY = new JLabel("ZERO-Y");
        lblSquareZeroY.setToolTipText("y-coordinate of origin (utm m)");
        lblSquareZeroY.setHorizontalAlignment(SwingConstants.RIGHT);
        lblSquareZeroY.setVerticalAlignment(SwingConstants.TOP);
        lblSquareZeroY.setBounds(27, 53, 61, 16);
        squarePanel.add(lblSquareZeroY);

        squareZeroX = new JTextField();
        squareZeroX.getDocument().addDocumentListener(new DocumentListener() {
                public void insertUpdate(DocumentEvent de) { interceptDouble(squareZeroX, squareZeroY, btnSquareGenerate); }
                public void removeUpdate(DocumentEvent de) { interceptDouble(squareZeroX, squareZeroY, btnSquareGenerate); }
                public void changedUpdate(DocumentEvent de) { interceptDouble(squareZeroX, squareZeroY, btnSquareGenerate); }
        });
        squareZeroX.setToolTipText("x-coordinate of origin (utm m)");
        squareZeroX.setHorizontalAlignment(SwingConstants.RIGHT);
        squareZeroX.setBounds(100, 19, 134, 28);
        squareZeroX.setColumns(10);
        squareZeroX.setText("0");
        squarePanel.add(squareZeroX);
        squareZeroX.setColumns(10);

        squareZeroY = new JTextField();
        squareZeroY.getDocument().addDocumentListener(new DocumentListener() {
                public void insertUpdate(DocumentEvent de) { interceptDouble(squareZeroY, squareZeroX, btnSquareGenerate); }
                public void removeUpdate(DocumentEvent de) { interceptDouble(squareZeroY, squareZeroX, btnSquareGenerate); }
                public void changedUpdate(DocumentEvent de) { interceptDouble(squareZeroY, squareZeroX, btnSquareGenerate); }
        });
        squareZeroY.setToolTipText("y-coordinate of origin (utm m)");
        squareZeroY.setHorizontalAlignment(SwingConstants.RIGHT);
        squareZeroY.setBounds(100, 47, 134, 28);
        squareZeroY.setText("0");
```

```
squarePanel.add(squareZeroY);
squareZeroY.setColumns(10);

JLabel lblSquareDiv = new JLabel("SIZE");
lblSquareDiv.setToolTipText("nodes lattice cell size (m)");
lblSquareDiv.setHorizontalAlignment(SwingConstants.RIGHT);
lblSquareDiv.setBounds(27, 88, 61, 16);
squarePanel.add(lblSquareDiv);

squareDivSize = new JComboBox<Integer>();
squareDivSize.setToolTipText("nodes lattice cell size (m)");
squareDivSize.setModel(new DefaultComboBoxModel<Integer>(new Integer[] {1, 2, 3, 4, 5, 10, 20, 50, 100, 500, 1000}));
squareDivSize.setSelectedIndex(0);
squareDivSize.setBounds(100, 84, 134, 27);
squarePanel.add(squareDivSize);

JLabel lblSquareNum = new JLabel("NUM");
lblSquareNum.setToolTipText("number of lattice nodes in each direction");
lblSquareNum.setHorizontalAlignment(SwingConstants.RIGHT);
lblSquareNum.setBounds(27, 116, 61, 16);
squarePanel.add(lblSquareNum);

squareDivNum = new JComboBox<Integer>();
squareDivNum.setToolTipText("number of lattice nodes in each direction");
squareDivNum.setModel(new DefaultComboBoxModel<Integer>(new Integer[] {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}));
squareDivNum.setBounds(100, 112, 134, 27);
squarePanel.add(squareDivNum);

btnSquareGenerate.addMouseListener(new MouseAdapter() {
        public void mouseClicked(MouseEvent e) {
                if(btnSquareGenerate.isEnabled()){
                        int ord = 0;
                        double x = Double.parseDouble(squareZeroX.getText());
                        double y = Double.parseDouble(squareZeroY.getText());
                        double shift = Double.parseDouble(squareDivSize.getSelectedItem().toString());
                        StringBuffer sb = new StringBuffer();
                        sb.append("ORD,QUAD,ID,UTM_X,UTM_Y\n");
                        sb.append(++ord + ",AXIS,ZERO," + clean(x) + "," + clean(y) + "\n");
                        int max = Integer.parseInt(squareDivNum.getSelectedItem().toString());
                        for(int i=1; i<=max; i++){
                                sb.append(++ord + ",AXIS,N" + i + "," + clean(x) + "," + clean(y+i*shift) + "\n");
                                sb.append(++ord + ",AXIS,E" + i + "," + clean(x+i*shift) + "," + clean(y) + "\n");
                                sb.append(++ord + ",AXIS,S" + i + "," + clean(x) + "," + clean(y-i*shift) + "\n");
```

```
                                        sb.append(++ord + ",AXIS,W" + i + "," + clean(x-i*shift) + "," + clean(y) + "\n");
                                        for(int j=1; j<=max; j++){
                                                sb.append(++ord + "," + "Q1,E" + i + "N" + j + "," + clean(x+i*shift) + "," + clean(y+j*shift) + "\n");
                                                sb.append(++ord + "," + "Q2,W" + i + "N" + j + "," + clean(x-i*shift) + "," + clean(y+j*shift) + "\n");
                                                sb.append(++ord + "," + "Q3,W" + i + "S" + j + "," + clean(x-i*shift) + "," + clean(y-j*shift) + "\n");
                                                sb.append(++ord + "," + "Q4,E" + i + "S" + j + "," + clean(x+i*shift) + "," + clean(y-j*shift) + "\n");
                                        }
                                }
                                nodesList.setText(sb.toString());
                                mainPanel.setEnabledAt(2, true);
                                mainPanel.setForegroundAt(2, Color.BLACK);
                                mainPanel.setSelectedIndex(2);
                        }
                }
        });
        btnSquareGenerate.setToolTipText("generate square lattice nodes");
        btnSquareGenerate.setBounds(17, 179, 134, 29);
        squarePanel.add(btnSquareGenerate);

        JButton btnSquareReset = new JButton("RESET");
        btnSquareReset.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent e) {
                        squareZeroX.setText("0");
                        squareZeroY.setText("0");
                        squareDivNum.setSelectedIndex(0);
                        squareDivSize.setSelectedIndex(0);
                        btnSquareGenerate.setEnabled(true);
                        mainPanel.setEnabledAt(2, false);
                        mainPanel.setForegroundAt(2, Color.GRAY);
                        nodesList.setText("");
                }
        });
        btnSquareReset.setToolTipText("reset all fields");
        btnSquareReset.setBounds(179, 179, 134, 29);
        squarePanel.add(btnSquareReset);
}

private void initializeNodesPanel(JPanel nodesPanel){

        nodesPanel.setLayout(null);

        nodesList = new JTextArea();
        nodesList.setEditable(false);
```

```
    JScrollPane scroll = new JScrollPane (nodesList);
    scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);
scroll.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
    scroll.setBounds(10, 10, 300, 162);
    nodesPanel.add(scroll);

    btnExport = new JButton("EXPORT");
    btnExport.addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent e) {
                    JFileChooser jfc = new JFileChooser();
                    jfc.setSelectedFile(new File(DEFAULT_FILE_NAME));
                    if (jfc.showSaveDialog(btnExport) == JFileChooser.APPROVE_OPTION) {
                    File file = jfc.getSelectedFile();
                            try{
                                    if (!file.exists()) file.createNewFile();
                                    FileWriter fw = new FileWriter(file.getAbsoluteFile());
                                    BufferedWriter bw = new BufferedWriter(fw);
                                    bw.write(nodesList.getText());
                                    bw.flush();
                                    bw.close();
                                    JOptionPane.showMessageDialog(null, "file " + file.getAbsolutePath() + " saved");
                            }
                            catch(IOException ioex){
                                    JOptionPane.showMessageDialog(null, "failed saving " + file.getAbsolutePath());
                            }
                    }
            }
    });
    btnExport.setToolTipText("export to csv file");
    btnExport.setBounds(24, 184, 117, 29);
    nodesPanel.add(btnExport);

    JButton btnClip = new JButton("CLIP");
    btnClip.addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent e) {
                    StringSelection selection = new StringSelection(nodesList.getText());
                Clipboard clipboard = Toolkit.getDefaultToolkit().getSystemClipboard();
                clipboard.setContents(selection, selection);
                JOptionPane.showMessageDialog(null, "nodes copied to clipboard");
            }
    });
    btnClip.setToolTipText("copy to clipboard");
    btnClip.setBounds(179, 184, 117, 29);
```

```
            nodesPanel.add(btnClip);
        }


        private static double clean(double inval){
            return Math.round(inval * 1000.d) / 1000.d;
        }

        private boolean checkDouble(JTextField tf){
            try{
                    Double.parseDouble(tf.getText());
                    return true;
            }
            catch(NumberFormatException nex){
                    return false;
            }
        }

        private void interceptDouble(JTextField tf, JTextField othertf, JButton button){
            if(checkDouble(tf)){
                tf.setForeground(Color.BLACK);
                if(checkDouble(tf) && checkDouble(othertf)){
                    button.setEnabled(true);
                }else{
                    button.setEnabled(false);
                }
            }else{
                tf.setForeground(Color.RED);
                button.setEnabled(false);
            }
        }

}
```