

Ferramenta para classificação de nuvem de pontos cotados

Luis Fernando Chimelo Ruiz ¹

Laurindo Antonio Guasselli ¹

Alexandre ten Caten ²

Guilherme Garcia de Oliveira ¹

¹ Universidade Federal do Rio Grande do Sul

Centro Estadual de Pesquisas em Sensoriamento Remoto e Meteorologia

Programa de Pós-Graduação em Sensoriamento Remoto

Av. Bento Gonçalves, 9500, Porto Alegre - RS, Brasil

ruiz.ch@gmail.com, laurindo.guasselli@ufrgs.br, g.g.oliveira10@gmail.com

² Universidade Federal de Santa Catarina

Campus Curitibanos

Rod. Ulysses Gaboardi, Km 3, Curitibanos - SC, Brasil.

tencaten@gmail.com

Abstract. The understanding of the behavior of the relief serves as basis for studies in hydrology, pedology and geomorphology. Methods of surveying for this purpose, as LIDAR or UAV with photographic cameras, do not collect terrain data, aye of the surface. The problem is that studies aiming at modeling the relief in need of a Digital Terrain Model (DTM) information terrain. One possibility to extract only the terrain points is applying a filter in the point cloud, order to remove points on undesirable local. The methods employed in filtering based only in information geometric and due to this have difficulties to remove points on small trees. Thus, this study aimed to develop and evaluate a tool for classifying cloud of elevation points, using the classification model by decision tree. To develop this work was used the programming language Python 2.7, library machine learning Scikit-Learn (sklearn) and the framework of spatial data processing available in QGIS program. The tool developed by using the decision tree model obtained a classification of points on undesirable sites with accuracy of approximately 90% for the building class and 93% for the class tree. The highest hit rates were found when was used the default value for the growth of the classification model, achieving to classify small trees.

Palavras-chave: filtragem nuvem de pontos, Veículo Aéreo Não Tripulado, modelo digital do terreno, filter points cloud, Unmanned Aerial Vehicle, digital terrain model.

1. Introdução

O entendimento do comportamento do relevo serve de embasamento para estudos em hidrologia, pedologia e geomorfologia. Para a coleta desses dados podem ser utilizados levantamentos com sistema de navegação global por satélites (*Global Navigation Satellite System* - GNSS), perfilamento a laser (*Light Detection And Ranging* - LIDAR), interferometria por radar de abertura sintética (*Interferometric Synthetic Aperture Radar* - InSAR) e ultimamente, câmeras não métricas acopladas em Veículos Aéreos Não Tripulados (VANT).

Os VANT com câmeras não métricas podem ser uma alternativa quando são considerados os gastos envolvidos na coleta dos dados, esses sistemas possuem baixos custos de operação e manutenção (Everaerts, 2008). O processamento das imagens capturadas com essa tecnologia utilizam a abordagem de Estrutura a partir do Movimento (*Structure from Motion* - SfM) para a obtenção de ortoimagens e nuvem de pontos cotados (Fonstad et al., 2013).

A nuvem de pontos cotados é comumente utilizada em ambientes de Sistema de Informação Geográfica (SIG) como grades regulares. Para isso técnicas interpolações são aplicadas na nuvem de pontos, obtendo como resultado um Modelo Digital de Superfície (MDS) (Mancini et al., 2013). O problema é que estudos com objetivo de modelar o relevo necessitam de um Modelo Digital do Terreno (MDT), ou seja, apenas informações do terreno. Os métodos de levantamento que utilizam LIDAR e câmeras fotográficas, entretanto, coletam dados da superfície, incluindo feições como prédios e árvores (Bandara et al., 2011).

Uma possibilidade de extrair pontos apenas do terreno é aplicando uma filtragem na nuvem de pontos com o objetivo de remover pontos sobre os locais indesejáveis (árvores e prédios). Os métodos de filtragem empregados para essa finalidade podem ser considerados, em alguns casos, complexos para a solução do problema (Naci et al., 2003) e também, como consideram apenas informações geométricas (declividade entre pontos), apresentam dificuldades na remoção de pontos sobre vegetação de pequeno porte (Bandara et al., 2011).

A nuvem de pontos proveniente de câmeras não métricas possui uma tabela de dados com valores altimétricos e também, valores das bandas do sensor (Rosnell e Honkavaara, 2012). Esses dados das bandas podem ser explorados para a remoção dos pontos nos locais indesejáveis.

Nesse sentido, este trabalho teve como objetivo desenvolver e avaliar uma ferramenta para classificação de nuvem de pontos cotados, utilizando o modelo de classificação por árvore de decisão.

2. Metodologia de Trabalho

2.1 Desenvolvimento da ferramenta

Para o desenvolvimento deste trabalho foi utilizada a linguagem de programação Python 2.7 (Rossum, 2013), a biblioteca de aprendizagem de máquina *Scikit-Learn* (sklearn) (Pedregosa et al., 2011) e o *framework* de processamento de dados espaciais disponível no programa QGIS 2.4 (Olaya, 2014). Python é uma linguagem de alto nível, interpretada e com tipagem dinâmica. Foi criada em 1991, por Guido van Rossum, seguindo a Licença Pública Geral (*General Public License* - GPL). No *framework* disponível no QGIS é possível integrar programas livres como o GRASS-GIS, SAGA-GIS, Orfeo e o TauDEM. Para construir novos modelos de processamentos pode ser utilizada uma forma gráfica, ou desenvolver códigos utilizando a linguagem R ou Python. Outra facilidade desse *framework* é a capacidade de criar interfaces gráficas, utilizando linhas de comando. Para a integração desses programas e da linguagem R são necessárias configurações das opções do *framework*. Para acessar as ferramentas do *framework* e do QGIS foram utilizadas as bibliotecas *processing* e *PyQGIS* (Sherman, 2014), respectivamente.

A elaboração da ferramenta iniciou pela construção da interface gráfica, como ilustrado no Quadro 1. O caractere especial “##” é utilizado no começo de cada linha para informar que os comandos a seguir serão referentes a interface gráfica. Cada linha cria uma parte da interface: (1) na primeira linha é informado o grupo de *scripts* que será inserido o código, nesse caso será criado um novo grupo com o nome de *Classification by decision tree*; (2) a segunda linha cria uma caixa de escolha, no qual o usuário pode informar qual arquivo vetorial que será utilizado para treinamento do modelo; (3) na terceira linha é gerada uma caixa de seleção em que aparecem os nomes das colunas da tabela do arquivo vetorial, os nomes das colunas são referentes ao arquivo vetorial selecionado anteriormente. Para realizar essa ligação é necessário apenas repetir o nome da variável que recebe o arquivo vetorial no final da linha do código; (4) na linha quatro também é gerada uma caixa para o usuário selecionar o arquivo vetorial referente aos dados que serão classificados; (5) na linha cinco é utilizada uma caixa de texto que aceita apenas números, o valor inserido nessa caixa será utilizado como parâmetro no modelo de classificação. Esse parâmetro controla a profundidade da árvore, ou seja, o quanto o modelo pode se ajustar aos dados de treinamento, quanto maior o valor mais complexa será a árvore; (6) a sexta linha do código gera uma caixa de texto, um botão e uma opção booleana. Essa última interface serve para o usuário informar o local e o nome do arquivo que será gerado.

Quadro 1: Código que gera a interface do usuário

<p> Linha 1 - ##Classification by decision tree=group Linha 2 - ##Vector_samples=vector Linha 3 - ##Classes_field=field Vector_samples Linha 4 - ##Vector_datas=vector Linha 5 - ##Maximum_depth=number 0 Linha 6 - ##Output_classification=output vector </p>

Após a criação da interface são escritos os códigos que irão manipular os arquivos vetoriais. Para facilitar a codificação foram criadas quatro funções de processamentos, as duas primeiras fazem a leitura da tabela dos arquivos vetoriais e as transformam em uma estrutura de dados conveniente para a leitura desses dados na biblioteca *sklearn*.

A terceira função gera o modelo de classificação por árvore de decisão (Quadro 2). Para treinar o modelo são aceitos dados numéricos e nominais, os dados nominais apenas devem ser inseridos na tabela como *string*. A biblioteca utiliza o modelo CART (*Classification and Regression Trees*) para gerar a classificação, utilizando como critério de mensuração da impureza o coeficiente *Gini* ou a entropia (Quinlan, 1993). A complexidade da árvore pode ser controlada pelo parâmetro *Maximum_depth*. Posteriormente, a isso é possível ajustar o modelo com os dados de treinamento e classificar a nuvem de pontos (*modelTree.predict()*).

Na última função é gerado o novo arquivo vetorial, contendo a tabela do vetor de dados e uma nova coluna com as classes, esses valores das classes estão na lista (*self.classificationDatas*) obtidas na terceira função.

Essa ferramenta está disponível na íntegra no endereço eletrônico https://github.com/RuizGeo/classification_decision_tree e pode ser inserida no QGIS seguindo os seguintes passos: Processamento -> *Scripts* -> *Tools* -> *Add script from file*.

Quadro 2 - Classificação por árvore de decisão utilizando sklearn.

```

trans_train = vec.fit_transform(self.training).toarray()
trans_dadas = vec.fit_transform(self.dadas).toarray()
clf = tree.DecisionTreeClassifier( max_depth = Maximum_depth )
modelTree = clf.fit(trans_train, self.classes)
self.classificationDadas = modelTree.predict(trans_dadas)

```

2.2 Classificando a nuvem de pontos

A nuvem de pontos foi obtida a partir de imagens capturadas com um Veículo Aéreo Não Tripulado (VANT) hexacoptero da Universidade Federal de Santa Catarina (UFSC), Campus Curitibanos, mais detalhes do voo estão informados na Tabela 1. O voo foi realizado sobre a área da UFSC e foram distribuídos na nesse local dez quadros com as dimensões de 50 por 50 cm, esses quadros possuem um xadrez de cores, preto e branco. Sobre esses quadros foram coletadas coordenadas tridimensionais com posicionamento relativo cinemático em tempo real (RTK – *Real Time Kinematic*), no sistema de coordenadas Universal Transversa de Mercator (UTM), fuso 22 e no sistema de referência SIRGAS2000. As coordenadas obtidas apresentaram uma precisão de 2 cm na horizontal e 3 cm na vertical. Também foi utilizado o programa Agisoft Lens para calcular as distorções provenientes da lente do sensor. A georreferencia do bloco de imagens apresentou uma precisão menor que um pixel, sendo que a resolução espacial da ortomagem foi de 15 cm.

As imagens e os dados auxiliares foram inseridos no programa PhotoScan, obtendo-se como resultado uma densa nuvem de pontos. No total foram gerados 380.244 de pontos para uma área de 6 hectares. Para diminuir o tempo de processamento foram selecionados 133.532 pontos.

Tabela 1. Especificações do aerolevanteamento.

Itens	Descrições
Plataforma	Hexacoptero
Tempo de voo	12 minutos
Velocidade	15 km/h
Altura do voo	90 metros
Sobreposição entre faixas	70%
Sobreposição entre imagens	60%
Marca da câmera	Canon
Resolução espacial da câmera	12 megapixel
Resolução espectral da câmera	banda do vermelho, verde e azul (R, G e B)

A nuvem de pontos resultante possui uma tabela de dados com valores de altimetria e das bandas do sensor, neste caso, valores para R, G e B. Também foi gerado um agrupamento de valores de altimetria, utilizando a ferramenta *Cluster Points Cloud* do programa SAGA-GIS, selecionando a opção de distância mínima e o valor 15 para o número total de agrupamentos. A coluna com valores do agrupamento foi convertida para nominal por meio da calculadora de campo do QGIS. Essa conversão faz com que a biblioteca sklearn entenda que as classes de agrupamentos são nominais e não numéricas. Outro atributo calculado foi a razão entre as bandas do verde e do vermelho, gerando assim uma nova coluna com esses valores. No total

foram utilizadas seis variáveis preditoras para gerar os modelos de classificações por árvore de decisão: os valores de altimetria; as três bandas da imagem; o *cluster* da altimetria; e a razão entre bandas.

Para treinar o modelo foram selecionados amostras de pontos para as classes: árvores, calçadas, solo exposto, asfalto, prédio e campo. Foram realizadas seis classificações com valores 0, 2, 5, 10, 15 e 30, para o parâmetro *Maximum depth*, criando as classificações AD_default, AD_2, AD_5, AD_10, AD_15 e AD_30, respectivamente.

As avaliações das classificações foram efetuadas pela seleção de pontos em áreas consideradas críticas, foram selecionados 1251 pontos sobre árvores e 355 pontos sobre o prédio. Após cada classificação foi analisado a quantidade de pontos classificados corretamente. Também foi avaliado o tamanho da árvore considerando o número total de nós gerados.

3. Resultados e Discussão

A interface gráfica criada está ilustrada na Figura 1, a partir da qual foi possível executar o trabalho com mais facilidade, um problema encontrado foi a falta de uma interface que possibilitasse a seleção de várias colunas da tabela de uma vez, para suprir essa deficiência foi pré-estabelecido que as colunas das tabelas devem ser as mesmas para ambos os arquivos vetoriais. Apenas as amostras de treinamento possuem a coluna classe a mais. O *framework* de processamento do QGIS facilitou o desenvolvimento da interface e a forma como foi disponibilizada a ferramenta.

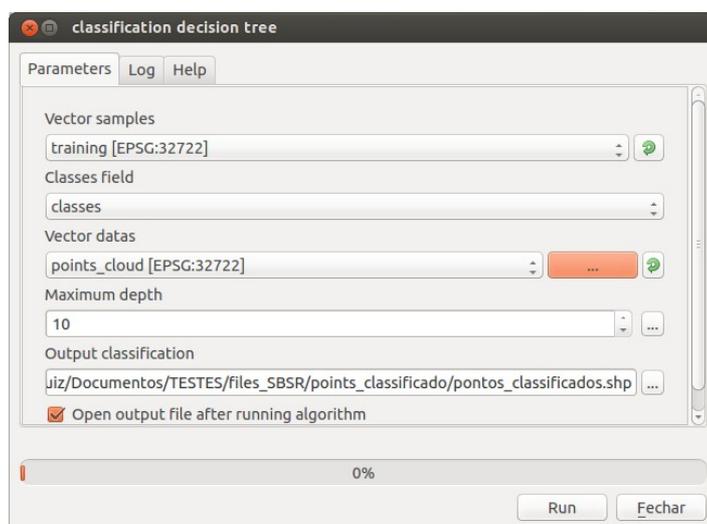


Figura 1. Interface gráfica da ferramenta *Classification by decision tree*.

Os modelos por árvore de decisão possibilitaram as classificações dos pontos com aproximadamente 90% de acerto para a classe prédio e 93% para a classe árvore (Tabela 2). As confusões mais significativas foram entre as classes prédio com asfalto e calçada e entre as classes árvore e solo exposto. Essas confusões estão relacionadas com a semelhança espectral das feições, constatando que os modelos gerados atribuíram uma maior relevância aos valores das bandas. Isso pode ter ocorrido porque foram inseridos mais atributos preditores referentes as bandas.

As maiores taxas de erros ocorreram quando foi utilizado o valor dois (AD_2) para o parâmetro de crescimento da árvore, esses equívocos foram entre as classes prédio e calçada, também entre as classes árvore e solo exposto. Isso pode ter acontecido devido ao número de nós dessa árvore, no qual resultou em apenas sete nós. O número reduzido de nós gerou uma árvore mais generalizada, consequentemente degradando o poder preditivo do modelo. Em

um estudo feito por Ruiz et al. (2014) foi constatado um decréscimo da predição do modelo por árvore com a diminuição dos valores dos nós folhas.

Analisando o número de nós de decisões gerados e os acertos de cada modelo, entende-se que pode ser utilizado *default* (AD_default) para o parâmetro de crescimento. Ao selecionar o *default* o crescimento da árvore é controlado pelo número máximo de amostras em cada nó e em cada folha. O valor padrão para esses dois parâmetros é o mínimo possível, permitindo assim o crescimento máximo da árvore, que neste caso resultou nos melhores valores de acertos. Outra interpretação possível é que em algum momento a árvore possui um crescimento máximo, como pode ser visto na AD_30, onde os resultados obtidos nessa classificação foram semelhantes a AD_default. O tempo de processamento para cada classificação foi de aproximadamente 45 segundos.

Tabela 2. Resultado das classificações em relação aos dados de validação.

Classes	AD_default		AD_2		AD_5		AD_10		AD_15		AD_30	
	Prédio	Árvore	Prédio	Árvore	Prédio	Árvore	Prédio	Árvore	Prédio	Árvore	Prédio	Árvore
Prédio	90,99	0,08	37,75	0,08	80,85	0,08	89,86	0,08	90,14	0,08	90,14	0,08
Calçada	1,69	1,28	50,99	0,08	7,04	0,32	1,97	0,48	1,41	0,24	1,41	0,24
Árvore	3,66	94,23	3,66	89,58	3,66	93,67	3,38	94,15	3,38	93,99	3,38	93,67
Solo exposto	0,56	3,69	7,61	10,26	1,13	4,89	1,13	4,25	0,85	4,57	0,85	4,97
Asfalto	3,10	0,48	0,00	0,00	7,32	1,04	3,66	0,56	3,94	0,80	3,94	0,80
Campo	0,00	0,24	0,00	0,00	0,00	0,00	0,00	0,48	0,28	0,32	0,28	0,24
Número de nós	1471		7		61		563		1225		1471	

Na Figura 2a está ilustrada a ortoimagem da área de estudo, com as respectivas coberturas da terra. A nuvem de pontos gerada para a área está representada na Figura 2b. Pode-se observar que há pontos sobre o prédio com valores de altimetria muito baixos. Isso pode ter acontecido no momento de gerar a nuvem de pontos, porque em cima do prédio há superfícies refletoras (vidros) que dificultam a determinação de características e descritores nas imagens. Esses descritores são utilizados para unir as imagens e calcular o 3D dos pixels. Problemas semelhantes a esses foram encontrados por Mancini et al. (2013). Ao averiguar a classificação AD_default na Figura 2c, podemos constatar que esses pontos errôneos foram classificados como solo exposto ou árvores, esse acontecimento explica a taxa de erro (0,56%) entre as classes solo exposto e prédio visto na Tabela 2. A confusão produzida pela AD_default foi em relação a classe prédio com asfalto e calçada.

Analisando os círculos em vermelho na Figura 2a e 2c, percebe-se que a metodologia proposta conseguiu identificar árvores de pequeno porte, sendo um ponto positivo no momento da elaboração do MDT.

Para utilizar a nuvem de pontos classificada na obtenção do MDT o ideal é que todos os pontos sobre os locais indesejáveis sejam removidos, neste estudo ocorreram confusões entre classes e conseqüentemente, o MDT obtido por meio desses dados apresentariam inconsistências.

A metodologia empregada permitiu o uso de atributos preditores tanto numéricos quanto nominais. Entretanto, futuros trabalhos podem avaliar diferentes atributos, como também,

augmentar o número de amostras para as classes de interesse, que nesse caso, seriam as classes consideradas sobre o terreno (solo exposto, campo, calçada e asfalto).

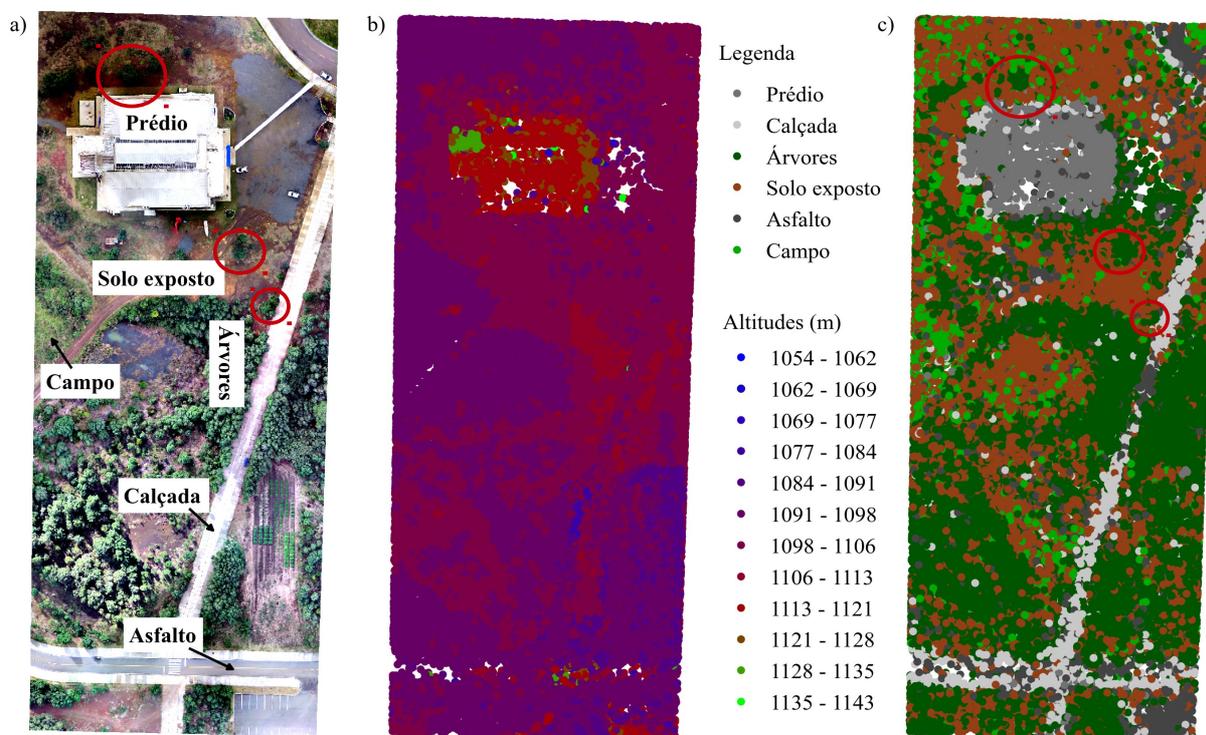


Figura 2. a) ortoimagem, b) nuvem de pontos original e c) nuvem de pontos classificadas de acordo com as classes de cobertura da terra.

4. Conclusões

A linguagem Python juntamente com as bibliotecas utilizadas se mostraram eficientes no desenvolvimento da ferramenta, o *framework* de processamento do QGIS facilitou a criação da interface gráfica e a manipulação dos arquivos vetoriais.

A ferramenta desenvolvida utilizando o classificador por árvore de decisão proporcionou a classificação dos pontos sobre locais indesejáveis com acerto de aproximadamente 90% para a classe prédio e 93% para a classe árvore. As maiores taxas de acertos foram encontradas quando utilizou-se o valor *default* para o crescimento do modelo de classificação, conseguindo classificar árvores de pequeno porte.

Referências

Fonstad, M. A.; Dietrich, J. T.; Courville, B. C.; Jensen, J. L. & Carbonneau, P. E. Topographic structure from motion: a new development in photogrammetric measurement. **Earth Surface Processes and Landforms**, v. 38, n.4, p. 421-430, 2013.

Quinlan, J.R. **C4. 5: programs for machine learning**. San Francisco: Morgan Kaufmann Publishers Inc., 1993, 302 p.

Mancini, F.; Dubbini, M.; Gattelli, M.; Stecchi, F.; Fabbri, S.; Gabbianelli & G. Using Unmanned Aerial Vehicles (UAV) for High-Resolution Reconstruction of Topography: The Structure from Motion Approach on Coastal Environments. **Remote Sensing**, v. 12, n.5, p.

6880-6898, 2013.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v.12, p. 2825-2830, 2011.

Rosnell, T. & Honkavaara, E. Point Cloud Generation from Aerial Image Data Acquired by a Quadcopter Type Micro Unmanned Aerial Vehicle and a Digital Still Camera. **Sensors**, v.12, n.1, p.453-480, 2012.

Rossum, G. V. **The Python Language Reference, Python Software Foundation**. 2013. Disponível em : <<http://docs.python.org/py3k/reference/index.html>>. Acessado em: 10 out. 2014.

Ruiz, L. F. C.; ten Caten, A.; Dalmolin, R.S.D. Árvore de decisão e a densidade mínima de amostras no mapeamento da cobertura da terra. **Ciência Rural**, v.44, n.6, p.1001–1007, 2014.

Sherman, G. E. **PyQGIS Programmer's Guide: Extending QGIS 2.4 with Python**. 2014. Disponível em: <http://docs.qgis.org/testing/en/docs/pyqgis_developer_cookbook/>. Acessado em: 8 out. 2014.

Olaya, V. **Plugin: Processing, Spatial data processing framework for QGIS**. 2014. Disponível em: <<https://plugins.qgis.org/plugins/processing/>>. Acessado em: 7 out. 2014.