Sensibility analysis of the Trace Transform on Land Coverage Images

Ricardo Román Brenes
Centro Nacional de Alta Tecnología and
Instituto Tecnológico de Costa Rica
Costa Rica
Email: rroman@ic-itcr.ac.cr, rroman@cenat.ac.cr

Francisco Torres-Rojas
Instituto Tecnológico de Costa Rica
Costa Rica
Email: torresrojas@gmail.com

Álvaro de la Ossa
Centro Nacional de Alta Tecnología and
Universidad de Costa Rica
Costa Rica
Email: alvaro.delaossa@ucr.ac.cr

***Abstract*—Aerial or satellital images may be used to produce terrain coverage maps, which in time are a very useful and important tool for decision-making in several fields including biodiversity, telecommunications and natural disaster management. The Trace Transform method can be used to process these images. This method extracts features from the images by applying a series of functionals to produce a numeric representation that will be used for classification later on. This model depends on several factors in order to have an efficient operation, among them, the frequency parameters of the traces, classifier type and land coverage. Experimentation on feature extraction time and precision rate revealed that the frequency parameters, specially $\Delta\tau$, and the classifier type can affect both of them.**

I. Introduction

At PRIAS[1], CeNAT's[2] environmental sensing laboratory, land coverage maps are produced using aerial or satellital images.

These maps are an important tool for decision makers, policy makers and administrative staff, among others, in areas like risk control, natural disaster management and biodiversity [1]. These products show a graphical distribution of an attribute overlayed on a map [2].

As of today at PRIAS, the process for creating a land coverage map has little computer assistance: the geographer takes the image from which the map is to be created, and on top of it, he or she draws polygons according to the kind of land it is: forest, water bodies or urban zones. This is a time-consuming and monotonous task that can be automated to a certain degree. PRIAS maintains a collection of over 13000 unclassified images, coming from different sources [3] [4].

The *Trace Transform (TT)* [5] has characterized and classified successfully and automatically different types of images [6] [7] [8]. This method extracts features associated with the image using several functionals, one after another, to reduce the complexity of the image. This reduced version of the image is known as *Triple Feature (TF)* and is the one used for classification.

At CNCA[3], CeNAT's computing research lab, a prototype implementation of TT was developed and used to classify experimental data with good results [9]. The next step will be to test the improved method using real data from PRIAS.

TT can have a considerable run-time, depending on several factors that will be explained further on. In [9], tests on 5000 $\times$ 5000 pixel images lasted an average of 3.7 hours. In order to reach a good precision level, several functionals have to be applied. The aerial images from PRIAS have sizes of at least 4000 $\times$ 4000 pixels [4]. Therefore it is of special importance that TT uses the right functional parameters in order to make it work in an efficient way.

This paper will review the TT algorithm, describe the experiments performed so far, to assess the influence of the parameters and the obtained results and outline future work. All the images, code and scripts are available at http://cluster.cenat.ac.cr/wordpress/?p=90

II. The Trace Transform

TT is a feature extraction method. Its basic idea is to draw lines (called emphtraces) over an image, from where pixels are sampled in order to transform it into a much simpler version but with an equally powerful meaning.

A detailed description of the algorithm can be found in [10], yet here the most relevant details are mentioned.

TT sweeps through an image I with lines, that can be defined by two parameters: ρ and ϕ. Let o be the center of I. From o, ρ is a distance and ϕ is an angle from a coordinate axis centered at o. A trace τ (line) is made such that it is tangent to the circle of radius ρ at angle ϕ, as it is shown in figure 1.

The model also has three frequency parameters:

- $\Delta\rho$ (dRho): defines the length in pixels from o. If $\Delta\rho$=1 there will be traces each pixel, if $\Delta\rho$=2, traces will be made every 2 pixels, and so on.
- $\Delta\phi$ (dPhi): determines the interval in degrees in which the trace will be made. If $\Delta\phi$=1°, traces will be made every degree, for a total of 360 lines. If it has a value of 90°, traces will be made every right angle, for a total of 4 lines.
- $\Delta\tau$ (dTau): over τ, this parameter defines which pixels are taken into account for feature extraction. If $\Delta\tau$=1,

[1]Programa de Investigaciones Aerotransportadas y Sensores Remotos; in English Airborne Research and Remote Environmental Sensing Program

[2]Centro Nacional de Alta Tecnología; in English National Center for Advanced Technology

[3]Colaboratorio Nacional de Computación Avanzada; in English National Collaboratory for Advanced Computing

978-1-4673-7846-8/15/$31.00 ©2015 IEEE

Fig. 1. Traces from a Trace Transform parameters.

all the pixels of the line will be used. If its value is 2, then every other pixel will be used, and so on.

Both $\Delta\phi$ and $\Delta\rho$ function like polar coordinates. Following the pixel selection three functionals will be applied in succession; the pixels that are touched by these τ lines and that comply with $\Delta\tau$ in I will be the ones used as input for the first functional T, or trace functional. The result, a first reduction of I, is a two-dimensional array (that can be interpreted as a 2D image), I_T, defined in terms of parameters of ϕ and ρ. The next step is to take each column of I_T and use it as input for functional P, or diametric functional. This will generate an array I_P defined in terms of ϕ. Finally, the last functional, Φ or circus functional, is applied to this array to create a single decimal number called I_Φ or *Triple Feature (TF)*. This process can be seen in figure 2.

It is possible to use more than one functional T. We can use several that can be indexed in a table of functionals T. These produce multiple I_T, which can be called I_T^τ, where τ is the index number of that functional in the table of functionals T. Using multiple functionals in the three parts of the TT has shown to be a good practice, since it will produce an array of TF that can be more representative of I than a single TF.

Once all the desired TF are extracted, these arrays have to be classified. For this task any kind of computational classifer may be used. Methods like artificial neural networks, support-vector machines, clustering and regressions are all valid.

III. Experimentation

Two experiments were designed to test the sensitivity of TT when the frequency parameters are changed from the default set ($\Delta\tau$=1, $\Delta\rho$=1 $\Delta\phi$=1°), following the multifactorial experiment design from [11].

A. Experiment A: Feature Extraction Run-time.

The first experiment measured the influence of the factors (independent variables) $\Delta\tau$, $\Delta\rho$, $\Delta\phi$ and land coverage type on the response variable (dependant variable) time. The levels of each factor are:

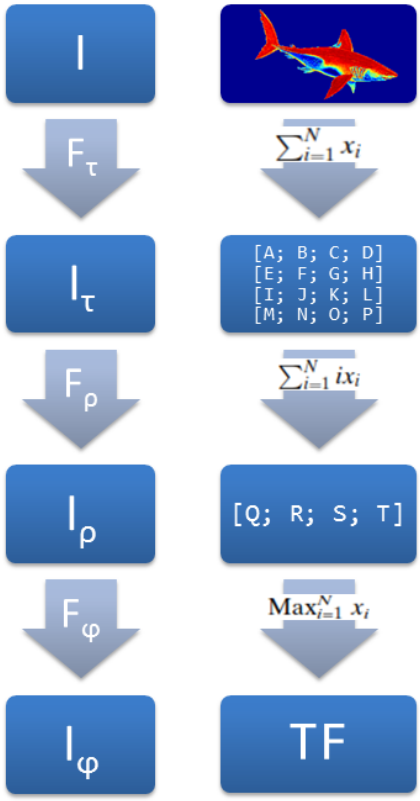


Fig. 2. Graphical flow of the Trace Transform.

- $\Delta\tau$ (pixels): 1, 2 and 3.
- $\Delta\rho$ (pixels): 1, 2 and 3.
- $\Delta\phi$ (degrees): 0.5°, 1° and 2°.
- Land coverage (types): Forest, water, rural, urban and plantation.

Given these levels and that there are 32 images per land coverage type, there is a total of 4320 ($3 * 3 * 3 * 32 * 5$) runs per replica (3), which yields a total of 12960 executions of the TT.

Using a script all the executions were automated to run all the combinations of the factor's levels. Each execution produced two results: the feature extraction run-time execution and a 5580 TF array for each of the images. The first was used in experiment A and the second was used in experiment B.

1) Dataset: The dataset used to generete the TFs contains 160 images previously classified by an expert geographer. There are 32 images for each land coverage type taken in Costa Rica during the Carta 2005 Mission [3]. Some examples can be seen in figure 3. Each image is 500×500 pixels.

2) Implementation of the TT: The program used in [9] was modified so that the frequency parameters could be loaded into execution instead of using the default set. The operating system is in charge of taking the time for the feature extraction. Each instance of the program operated over one image, so multiple programs could be launched in different machines to run over the dataset.

Fig. 3. Examples of land coverage types. Each column has images of a different land coverage, respectively: water, forest, rural, plantation and urban.

The program ran on CNCA's computing cluster. Five nodes were used, each with 32GB of RAM, Xeon E5530 processors and running CentOS 6.5.

3) Functionals: Functional selection is critical for a good extraction process, and is a research topic in itself. The functionals used here were those that showed the best results in classifying textures in [8] and [10].

B. Experiment B: Classification Precision

The second experiment measured the influence of $\Delta\tau$, $\Delta\rho$, $\Delta\phi$ and the classifier on the precision rate. The levels for factor classifier are: **KMeans**, **SMO**, **LogitBoost** and **BayesNet**. The levels for the other factors are the same as in experiment A.

A new field was added to each of the TF array (one for each image and parameter configuration) produced in experiment A, the land-coverage type of it's original image, so the classifiers can evaluate the array's ability to represent the image.

The data mining tool WEKA [12][4] was used to test the classifiers. Although there are other statistical tools available, like R [13][5], WEKA was chosen due to its vast list of publications[6], familiarity of use and output standards.

A total of 108 classifications were produced. Their output, among others, included a confusion matrix, recall and precision ratings and the method's particular output. The precision rates were taken and tabulated and as well as in the first experiment, were analyzed with ANOVA.

The precision rate is calculated using the confusion matrix and the following function:

$$\text{Precision} = \sum \frac{\text{Correctly classified instances}}{\text{Instance total}} \qquad (1)$$

[4]http://www.cs.waikato.ac.nz/ ml/weka/
[5]http://www.r-project.org/about.html
[6]http://www.cs.waikato.ac.nz/ml/publications.html

1) Classifiers: Four classifiers were selected to weigh the effects on precision. Each one has a different nature: clustering, function-based, regression and Bayesian.

The first one is the basic clustering algorithm, **k-means** [14], named "simpleKMeans" in WEKA. It was set up to use the whole dataset for training, 500 iterations and Euclidean distance.

The second one is a **support-vector machine** learning algorithm, using Platt's Sequential Minimal Optimization [15], known as "SMO" in WEKA. The setup used a polynomial kernel function, an acceptable error of 1.0E-12 and a normalization filter on the data. The dataset partition followed a cross-validation with 10 folds. This is a much more robust algorithm than the clustering and so it was expected to perform better.

The third one is a **regression** based algorithm, LogitBoost [16] with the same name in WEKA. It was used with the default parameters.

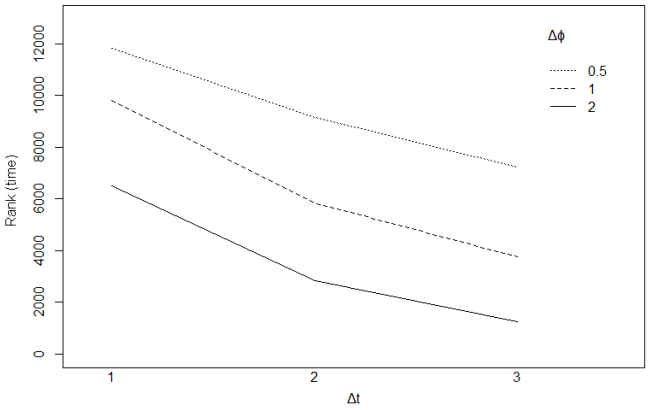
The fourth and last one is a **Bayes network** [17], named "BayesNet" in WEKA which also ran with the default parameters.

IV. Results and Analysis

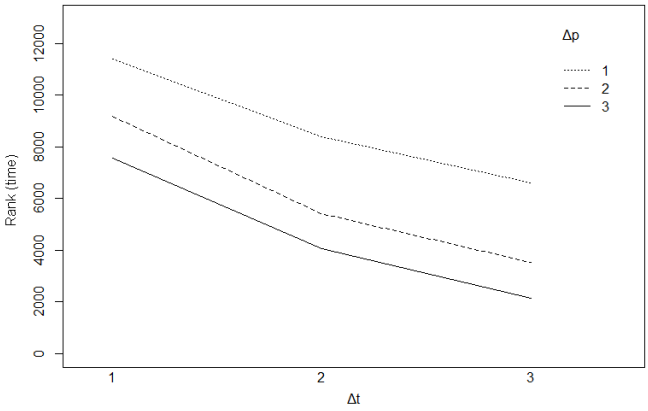
As was said before, the results of both experiments were tabulated and ran through ANOVA. The statistical procedure Analysis of Variance checks k samples obtained from k populations, and determines if there is statistical evidence that some of them have different means [11]. Unfortunately the data on either of the experiments did not follow ANOVA's assumptions (random error normally and independently distributed with mean 0.0 and constant variance) and thus a ranking transformation was applied to bypass these problems. The longer the time, the higher the rank; the fastest extraction time was of 5.177 seconds, which is rank 1. The slowest extraction time was of 216.606 seconds, which is rank 12960. The statistic package R was used for this analysis.

TABLE I. ANOVA TABLE FOR FEATURE EXTRACTION TIME.

Factor	Df	Sum Sq	Mean Sq	F value	Pr(>F)
dTau	1	6.056e+10	6.056e+10	9.681e+04	**<2e-16**
dPhi	1	7.188e+10	7.188e+10	1.149e+05	**<2e-16**
dRho	1	3.796e+10	3.796e+10	6.068e+04	**<2e-16**
coverage	4	2.135e+07	5.338e+06	8.533e+00	**7.15e-07**
dTau:dPhi	1	5.291e+07	5.291e+07	8.458e+01	**<2e-16**
dTau:dRho	1	1.383e+08	1.383e+08	2.211e+02	**<2e-16**
dPhi:dRho	1	1.108e+03	1.108e+03	2.000e-03	0.966
dTau:coverage	4	2.380e+06	5.951e+05	9.510e-01	0.433
dPhi:coverage	4	8.064e+04	2.016e+04	3.200e-02	0.998
dRho:coverage	4	1.340e+05	3.351e+04	5.400e-02	0.995
dTau:dPhi:dRho	1	2.709e+09	2.709e+09	4.330e+03	**<2e-16**
dTau:dPhi:coverage	4	1.172e+06	2.929e+05	4.680e-01	0.759
dTau:dRho:coverage	4	6.292e+04	1.573e+04	2.500e-02	0.999
dPhi:dRho:coverage	4	8.207e+05	2.052e+05	3.280e-01	0.859
dTau:dPhi:dRho:coverage	4	4.000e+05	1.000e+05	1.600e-01	0.959
Residuals	12920	8.082e+09	6.256e+05		



(a) $\Delta\tau$ and $\Delta\phi$



(b) $\Delta\tau$ and $\Delta\rho$

Fig. 5. Means of second level interactions in experiment A

A. Experiment A

The ANOVA table for the experiment A is shown in table I. The relevant factors and interactions are: $\Delta\tau$, $\Delta\tau$, $\Delta\rho$, land coverage type, $\Delta\tau$ and $\Delta\rho$, $\Delta\tau$ and $\Delta\phi$, $\Delta\tau$ and $\Delta\rho$ and $\Delta\phi$.

The four main factors all show influence on the response variable, plots of these effects can be found in figure 4. As the frequency parameters get larger, it takes less time to generate the TFs, which is something that was expected. The finer the sweep (smaller values on frequency) the longer the time the TT takes to extract features. The effect on the land-coverage factor is uncertain, although statistically significative, the difference between times vary from 31.323 seconds at rank 6400 to 31.354 seconds at rank 6500.5; rural coverages take slightly less time than the others.

The second level interactions between $\Delta\tau$ and the other two frequency parameters are in figure 5. Both interactions behave similarly, as $\Delta\tau$ and the other parameter increase together, it takes less time to extract the features. This does not occur with $\Delta\rho$, $\Delta\phi$ or the land-coverage type. These results make $\Delta\tau$ stand out from the others.

Finally the third level interaction between the three frequency parameters is shown in figure 6. Again the changes on $\Delta\tau$ affect time: the larger the parameter, the longer it takes. It can be seen in the plots that with $\Delta\tau = 1$ the lines are clustered at the top; with $\Delta\tau = 3$, they are clustered at the bottom and with $\Delta\tau = 2$, they are in the middle. Also $\Delta\rho$, $\Delta\phi$ both decrease toghter. This, again, stresses the importance of the right selection on $\Delta\tau$.

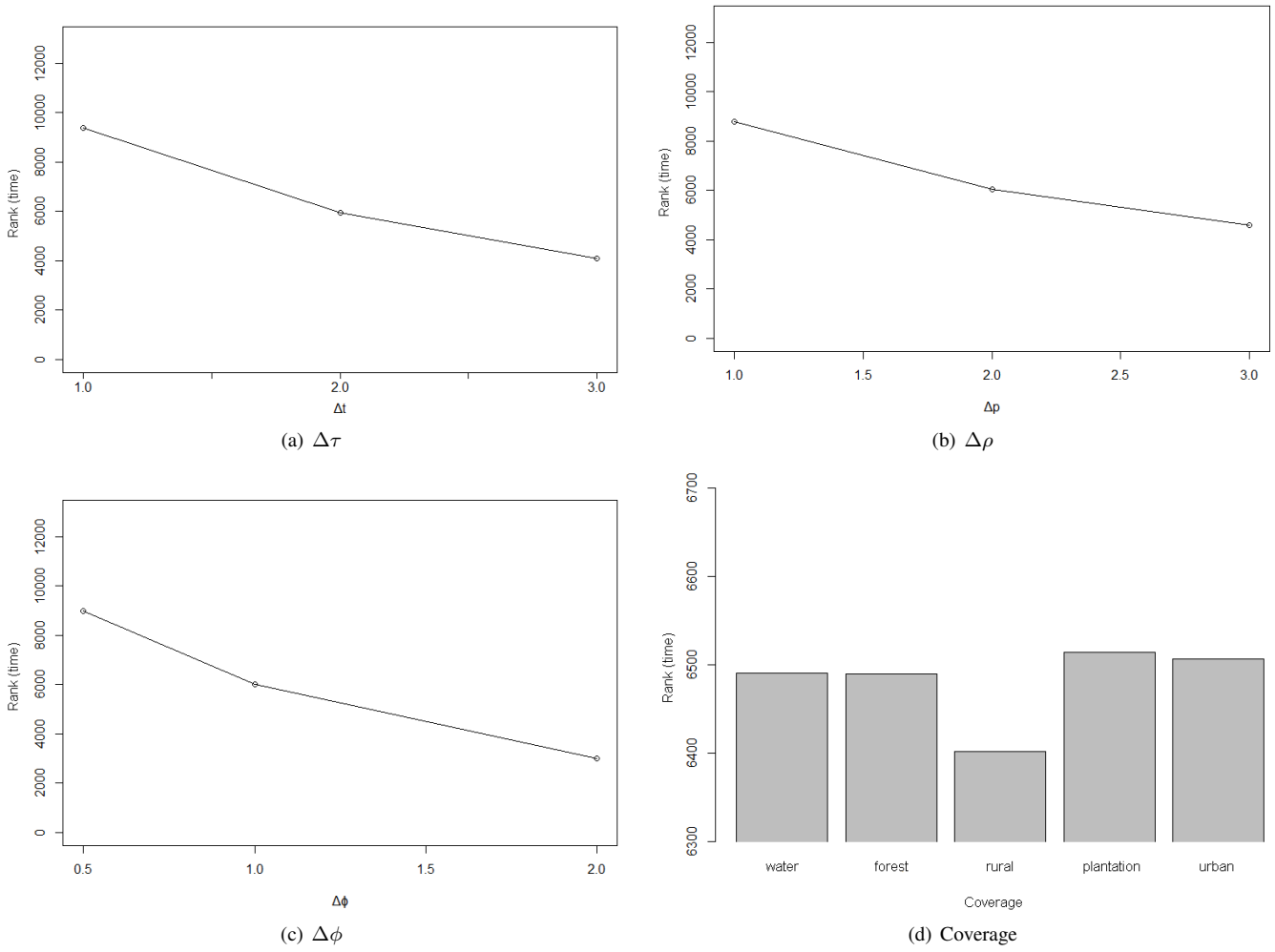
B. Experiment B

The ANOVA table for the experiment B is shown in table II. It is worth noticing that in this analysis a low rank means a high precision; so the bottom part of the plot has a high precision rate (low ranking) and low precision rate is at the top (high ranking). The relevant factors and interactions are: $\Delta\tau$, classifier; $\Delta\tau$ and classifier.

The factor $\Delta\tau$ has little effect on the precision, although statistically significative, this can be seen on plot (a) of figure

TABLE II. ANOVA TABLE FOR CLASSIFIER PRECISION.

Factor	Df	Sum Sq	Mean Sq	F-value	Pr(>F)
dTau	1	1168	1168	4.622	**0.034752**
dPhi	1	89	89	0.353	0.553947
dRho	1	133	133	0.528	0.469773
classifier	3	72454	24151	95.560	**<2e-16**
dTau:dPhi	1	58	58	0.230	0.632989
dTau:dRho	1	248	248	0.979	0.325495
dPhi:dRho	1	0	0	0.000	0.989759
dTau:classifier	3	5793	1931	7.641	**0.000157**
dPhi:classifier	3	1384	461	1.825	0.149661
dRho:classifier	3	1970	657	2.598	0.058362
dTau:dPhi:dRho	1	462	462	1.827	0.180523
dTau:dPhi:classifier	3	664	221	0.876	0.457247
dTau:dRho:classifier	3	62	21	0.082	0.969620
dPhi:dRho:classifier	3	208	69	0.275	0.843459
dTau:dPhi:dRho:classifier	3	474	158	0.625	0.601146
Residuals	76	19208	253		



(a) $\Delta\tau$ (b) $\Delta\rho$

(c) $\Delta\phi$ (d) Coverage

Fig. 4. Means of main factors of experiment A

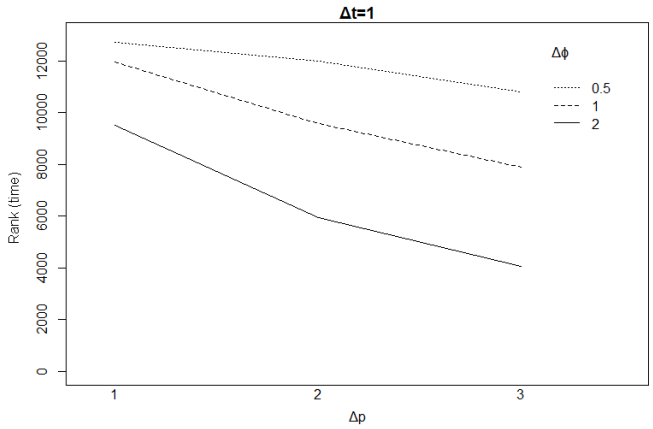
7. Plot (b) shows the effect of the classifier on precision rate, and here, heavy differences can be seen: K-means being by far the worst and LogitBoost being the best. K-means is a simple algorithm that depends heavily on the starting position of the centroids and the iterations. On the other hand LogitBoost relies on additive regression, that builds up a solution iteratively until it reaches certain level, so this makes it a far more precise process. On average the precision rate for K-means was 59.52%, BayesNet 88.75%, SMO 90.23% and LogitBoost 91.34%.

The second level interaction between $\Delta\tau$ and the classifier is in figure 8. Here it can be seen once again the heavy effect of the classifier; with $\Delta\tau$=1, the finer sweep, LogitBoost has a much better rank precision than any of the other classifier. When $\Delta\tau$=2, the places change, with SMO having a better performance than any other classifier, but having LogitBoost in close pursuit. At the coarse sweep, LogitBoost remains second to SMO in ranking precision. Given the ranking transformation, the raw data shows that LogitBoost is the better classfier, with SMO in close pursuit.

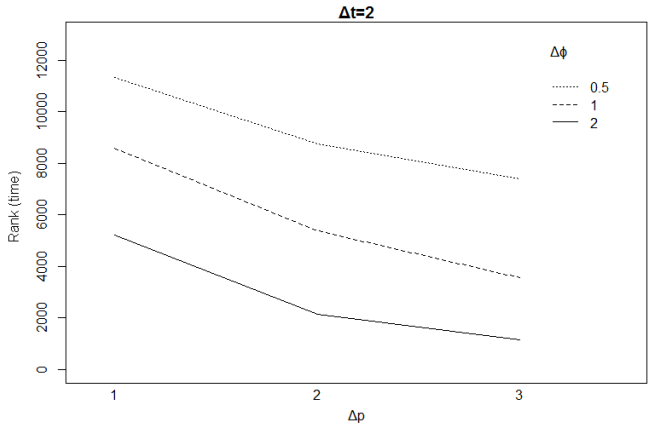
The main outcome that can be deduced from experiment A is that the larger the frequency parameters, the faster the extraction time is, with $\Delta\tau$ being the factor that has the biggest effect on time, since this is the parameter that selects which pixels will be used in extracting the features. On the other hand from experiment B, the precision rate appears to be in direct correspondance with the classifier and in a smaller degree of $\Delta\tau$. Both experiments are bonded with one relevant factor, again, $\Delta\tau$. Also there's an emerging property of the TT, it seems that depending on the classifier, it may not be worth making finer sweeps. The gain in precision does not justify the time taken, in particular when it is known that the real-life images are bigger than the ones used in these tests. On average the finer sweep with $\Delta\tau = 1$, $\Delta\rho = 1$ and $\Delta\phi = 1$ took 206.838 seconds while the coarser sweep with $\Delta\tau = 3$, $\Delta\rho = 3$ and $\Delta\phi = 2$ took 5.634 seconds. With the finer configuration and the best classifier LogitBoost, the precision rate hit a 95.00% an with the coarser configuration, a 89.38%. Gaining 5% in aerial image classification might not be of much help; but if the image is a CAT scan of the brain seaching for cancer, it might save lives.

V. Conclusions

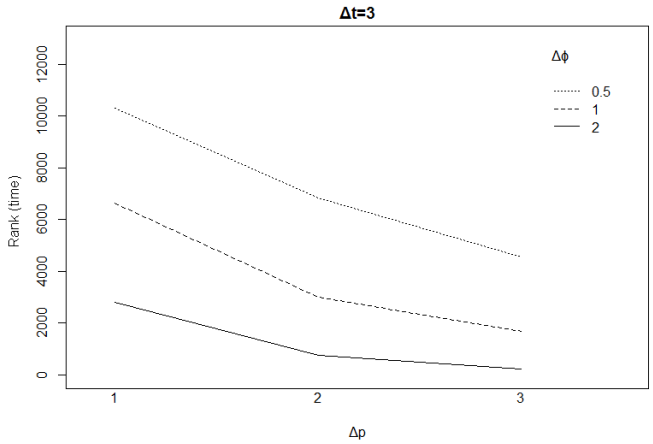
This study proposed two experiments to evaluate the influence of the frequency parameters, land coverage and classifier to the feature extraction time and precision rate of the Trace Transform on land coverage images. The first experiment



(a) $\Delta\tau$=1 and $\Delta\rho$ and $\Delta\phi$

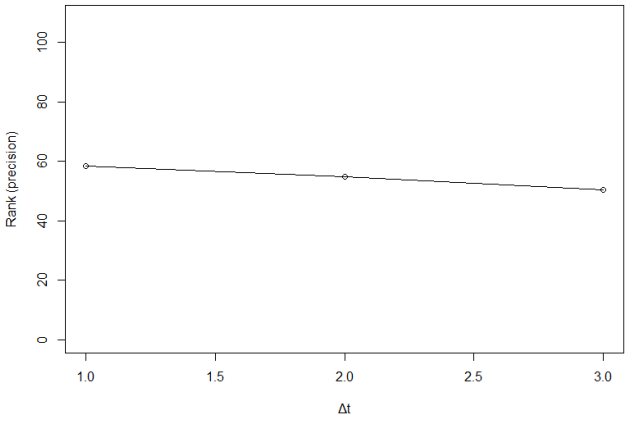


(b) $\Delta\tau$=2 and $\Delta\rho$ and $\Delta\phi$

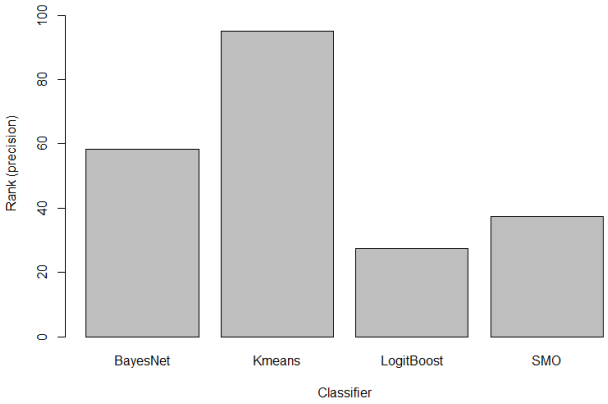


(c) $\Delta\tau$=3 and $\Delta\rho$ and $\Delta\phi$

Fig. 6. Means of third level interaction of experiment A



(a) $\Delta\tau$



(b) Classifier

Fig. 7. Means of $\Delta\tau$ and classifier factors for experiment B.

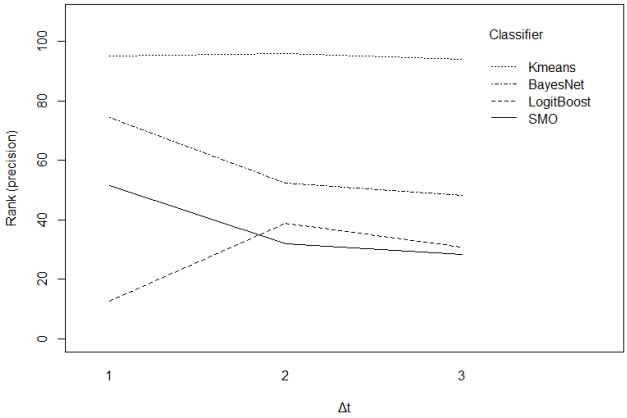


Fig. 8. Means of second level interaction in experiment B

showed how extraction time decreases as do the parameters and that the land coverage type influence was not very important. The second experiment revealed that the classifier is the main factor to influence precision rate. The study showed that TT can be optimized both at the parameter selection and the classifier choice.

VI. Future work

The next step is to build a land-coverage map generator using the classification given by the TT. Benchmarking would be recommened as well on different execution platforms. The prototype can be modified in order for it to work with batches of images. Another approach is to use an analysis window on larger images, rather than to segment that large image and pass those smaller ones to the TT. A study should be made to see if the window size can influence the map's precision. The application of TT to other types of images is encouraged, specially in the medical sciences, where there would be large datasets available that have already been studied.

Acknowledgment

A special thanks to CeNAT for the internship fund granted to make this research possible.

References

[1] BIOMARCC-SINAC-GIZ, "Clasificación sistemas marino costeros costa pacífica de costa rica," 2012. [Online]. Available: http://www.biomarcc.org/download_PDF/SerieTecnica2_SistemasMarinos.pdf

[2] P. Zatelli, "Thematic maps," 2014. [Online]. Available: http://www.ing.unitn.it/~zatelli/environmental_data_management/Thematic_maps.pdf

[3] SINAC-MINAE, "Redd+: Correción atmosférica de imagenes rapid eye," 2012. [Online]. Available: http://www.cenat.ac.cr/gestion-ambiental/programas/prias/misiones

[4] Programa de Investigaciones Aerotransportadas y Sensores Remotos, "Fotografías aéreas proyecto carta," 2005. [Online]. Available: http://finanzascarbono.org/nuevos-mecanismos-de-mitigacion/redd/

[5] A. Kadyrov and M. Petrou, "The trace transform and its applications," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 23, no. 8, pp. 811–828, Agosto 2001.

[6] B.-S. Shin, E.-Y. Cha, K.-B. Kim, K.-W. Cho, R. Klette, and Y. W. Woo, "Effective feature extraction by trace transform for insect footprint recognition," in *3rd International Conference on Bio-Inspired Computing: Theories and Applications*. IEEE, sep 2008, pp. 97 – 102.

[7] S. Srisuký, K. Ratanarangsanký, W. Kurutach, and S. Waraklang, "Face recognition using a new texture representation of face images," *Proceedings of Electrical Engineering Conference*, pp. 1097–1102, November 2003.

[8] M. Petrou, A. Talebpour, and A. Kadyrov, "Reverse engineering the way humans rank textures," *Pattern Analysis and Applications*, vol. 10, no. 2, pp. 101–114, 2007.

[9] R. Garita and A. De la Ossa, "Paralelización del método de la transfomada de trazo para el análisis de texturas en imágenes aéreas," in *6a Conferencia Latinoamericana de Computación de Alto Rendimiento*, 2013, awaiting publication.

[10] M. Petrou and F. Wang, "A tutorial on the practical implementation of the trace transform," in *Handbook of Texture Analysis*, X. X. S. J. Mirmehdi, M., Ed. Oxford: Oxford University Press, 2008.

[11] D. Montgomery, *Design and Analysis of Experiments*. John Wiley & Sons, 2001.

[12] M. Hall, I. Witten, and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[13] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014. [Online]. Available: http://www.R-project.org

[14] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 1:281–297, 1967.

[15] J. C. Platt, "Advances in kernel methods," B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208. [Online]. Available: http://dl.acm.org/citation.cfm?id=299094.299105

[16] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *The Annals of Statistics*, no. 2, pp. 337–407, 2000. [Online]. Available: http://web.stanford.edu/~hastie/Papers/AdditiveLogisticRegression/alr.pdf

[17] R. R. Bouckaert, "Bayesian network classifiers in weka for version 3-5-7," 2008. [Online]. Available: http://www.cs.waikato.ac.nz/~remco/weka.bn.pdf