

MONO-TEMPORAL GIS UPDATE ASSISTANCE SYSTEM BASED ON UNSUPERVISED COHERENCE ANALYSIS AND EVOLUTIONARY OPTIMISATION

Christian Becker, Joern Ostermann, Martin Pahl

Leibniz Universität Hannover
Institut für Informationsverarbeitung
Appelstr. 9a, 30167 Hannover, Germany
(becker, ostermann, pahl)@tnt.uni-hannover.de
<http://www.tnt.uni-hannover.de>

Commission IV/2

KEY WORDS: Land Cover, Updating, Change Detection, Semi-automation

ABSTRACT:

Data in Geo Information Systems (GIS) is used for map services and various applications. Thus, quality assessment on a regular basis is required to keep the data up-to-date. In this paper we focus on one key reason for updates: incorrect object borders. State of the art systems semi-automatically analyse up-to-date satellite image data to narrow down areas that have to be considered for GIS updates. Often resources are limited and only data from one point in time is available that is compared to the data. Rule based systems are required to bridge the gap between GIS specifications and results from image analysis. We present a system that can find areas of change without any manual configuration. Our approach automatically learns about important aspects of GIS specifications by analysing correct GIS objects. In potentially out-dated GIS data still a majority of objects is unchanged. Thus, we derive an model for normality (= correctness) by evaluating the coherence of relations between GIS objects and image analysis results. We synthesise changes at GIS object borders and analyse the impact on normality. In an evolutionary optimisation we determine areas of change that are rated with a significance value. We show that we can find 83% of all relevant update areas with a precision of 0.18, not considering the significance of changes. Including significance we can push the precision to 0.26 while still finding 77% of all relevant update areas.

1 INTRODUCTION

Up-to-date spatially referenced data is crucial for various applications like urban planning, hazard management and agriculture (Lu et al., 2004). Such data usually is stored as data of a Geo Information System (GIS). Each data entry is stored as an geo referenced object, features of the entry are stored as attributes. A mandatory attribute is the GIS *feature* to group object (eg. settlement or forest). Depending on the data, the shape may be specified as lines, polygons or points. While gathering and updating spatial reference data can be a very complex task (eg. performing field surveys and in-depth studies), finding potentially out-dated areas can be simplified by evaluating land cover changes in remote sensing data like satellite images. Still, manual checks of satellite images is very time consuming. Hence, support through (semi-) automatic systems is requested. Approaches that use various sensors and/or data for several have been developed (Lu et al., 2004). In many application only satellite images for a current point in time is available, though. In this mono-temporal setting images are analysed and results are compared to the GIS data to highlight potentially out-dated areas.

Existing systems can be divided into two groups: First, there are approaches that focus on image analysis methods for each considered GIS feature. Afterwards the comparison with the GIS data is considered a trivial step (Lacroix et al., 2006),(Leignel et al., 2010). However, currently only for special GIS features a reliable image analysis algorithm exists. Therefore, the correctness is often considered a problem in a multi-feature environment. The lack of correctness inevitably leads to irrelevant areas for update. Hence, a second group of systems introduce an evaluation step to perform the comparison. Existing systems (Busch et al., 2004), (Buck et al., 2011) introduce for each GIS feature manually configured rules in order to control the comparison. These systems can flexibly handle image analysis results, allowing to limit the

impact of only partial reliable results and to consider GIS feature specifications for determining update areas. On the other hand, rule sets can get too complex and unmanageable easily. Moreover, to be able to compensate for complex image analysis results a rule designer needs to have knowledge of the characteristics of the result as well as of GIS specifications (see Figure 1 for examples of GIS, image data and analysis results).

We show that important aspects of GIS specifications can be automatically and implicitly learnt from correct GIS objects. Since even it out-dated GIS data the vast majority of objects are still correct, learning from correct objects is the same as learning from normal objects. A normality measure of GIS objects is determined by evaluating the coherence of relations between potentially out-dated GIS data and image analysis results. We already presented a GIS object based system in (Becker et al., 2012) . In this paper we concentrate on one of the main aspects of this scenario: incorrect object borders of existing GIS objects. This additional constraint allows us to enhance spatial resolutions of update areas to sub-object level. Using image analysis results we are able to sub-divide GIS objects into segments. By reassigning (or as we name it: *switching*) segments from one GIS object to one of its neighbours we are able to synthesise expanded and shrunken GIS objects. Evaluating the changes in normality caused by switches we are able to find update areas. In this paper we show that it is possible to use the squared Mahalanobis distance to implicitly express to what extent an GIS object complies to GIS specifications, using only the coherence of relations between GIS objects and image analysis results. We show that this can be used to replace rule based systems. Finally we show that using segments we are able to give update hints at sub-object level.

In section 2 we first have to introduce some essential terms. In Section 3 we provide theoretical background how to measure nor-



Figure 1: Satellite image and image analysis result. Each colour represents a different kind of texture. GIS object borders describe the GIS features.

mality. This is required for Section 4 where we present how we model normality for the application to find incorrect object borders. In Section 5 we describe the evolutionary algorithm that is used to find segments that describe areas of change. Results are presented and discussed in Section 6. Finally we summarise our contributions in Section 7.

2 TERMINOLOGY

By analysing georeferenced image data with existing computer vision algorithms, we obtain a pixel-wise classification. Each pixel is classified into one of several specific *classes*. Interconnected pixels of the same class form georeferenced and classified *segments*. They cover the same area in the scene as the GIS objects. Therefore, we say that every GIS object is *composed of* (classified) segments and a segment is *connected to* a GIS object. To ensure that every segment is only connected to one specific GIS object, segments are split at GIS object borders. In our scene representation the location and extent of a (GIS) *object* is only defined by the segments connected to the object. Since we are going to change the composition of GIS objects we will name the state of compositions of all objects a *configuration* of the scene. When there is more than one scene, each scene is called *scene version*.

GIS object are composed of segments that are still represented as connections of pixels. While this representation is useful for visualisation, an automatic system needs information to be present as numerical *attributes*. To describe the composition of a GIS object we developed suitable object attributes. To compare several objects they have to provide the same attributes. Segments don't require any attributes (they still are classified, though).

3 MEASURING NORMALITY

In our system relations between a GIS object and image analysis results are described by a continuous multi-dimensional attribute vector, thus all objects form a continuous attribute space.

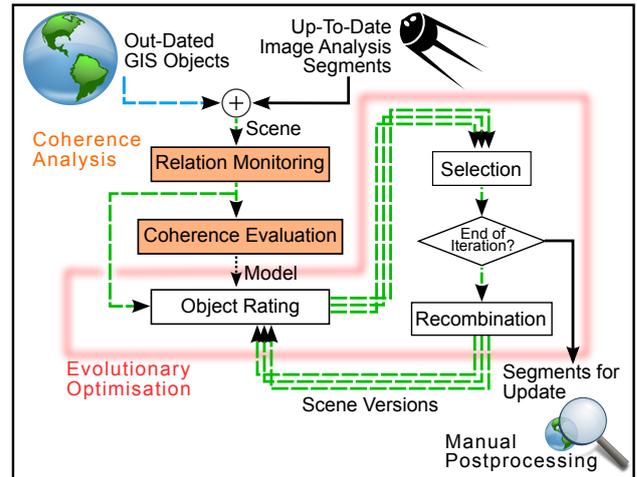


Figure 2: Object Coherence Analysis and Update Detection by Evolutionary Optimisation.

The topic normality and abnormality in attribute space is known in the field of *data mining* (Tan et al., 2005). To estimate a normality model, a large data set is required. We want to restrict the data basis for forming a statistical model of normality to GIS objects of the potentially out-dated input data set. Since the data may be partly incorrect the estimation of the model (some probability distribution) needs to be robust.

Robustness can be enhanced by considering a priori knowledge. Since we don't want to use configurations or rule sets, a-priority knowledge is rare, though. The only knowledge we assume is that GIS objects that belong to the same GIS feature also look similar. Depending on the GIS specifications this might require considering additional GIS attributes for sub-classification of GIS features. Choosing an appropriate probabilistic distribution is also fundamental to ensure robustness. We decided to use a multivariate Gaussian (normal) distribution to model the coherence of object attributes (= relations). The distribution's parameters *mean* and covariance matrix can be robustly estimated from large data sets. From our knowledge that a GIS object belongs to one of several GIS features, Gaussian distributions are estimated for each GIS feature independently. To determine distances in space with multivariate Gaussian distributions we use the squared Mahalanobis distance:

$$d_{mah}^2(x, y) := (x - y)S^{-1}(x - y)^T \quad (1)$$

where S is the covariance matrix of the data. The Mahalanobis distance is a standard method for this application (Tan et al., 2005). The covariance matrix S in the formula is used to compensate pair-wise correlations between attributes. To interpret the squared Mahalanobis distance as normality measure, the distance of an object towards the Gaussian's mean μ must be determined. Objects near the centre are more normal than more distant objects. Thus, the *abnormality value* of an GIS object o is calculated by

$$\text{abnormality}(o) := d_{mah}^2(o, \mu) \quad (2)$$

$0 \leq \text{abnormality} < \infty$, using the mean and covariance matrix determined for o 's GIS feature. When we use in this paper the term high *normality* this corresponds to a low abnormality value and vice versa. The absolute values of abnormality have no special meaning.

4 COHERENCE ANALYSIS

The algorithms performs several procedures as can be seen in Figure 2 that will be described in the following subsections.

4.1 Relation Monitoring

To monitor relations between image segments and GIS objects we develop attributes. Since we want to determine normality of GIS objects by using the squared Mahalanobis distance (see Section 3), we want the number of attributes to be small. A large number of attributes would require a very large number of samples to robustly approximate mean and covariances. The attributes should also be quite general so that all objects can be sensibly described by them. Finally, they should only reflect relevant information.

For our application we defined *Segment Histogram Attributes*. We will use the notation for sets as segments are connected pixels. Operation $|s|$ measures the size of a segment s in pixels, the size of the an object is the sum of sizes of the segments connected to the object.

For each GIS object o and all segment classes $i = 1, 2, \dots, n$ attributes a_1, a_2, \dots, a_n are calculated so that

$$a_i(o) := \frac{\sum_{s \in \text{Segments}} \delta_i(s) |g \cap s|}{|o|} \quad (3)$$

with

$$\delta_i(s) := \begin{cases} 1 & \text{if } s \text{ has segment class } i \\ 0 & \text{else} \end{cases} \quad (4)$$

Segment Histogram Attributes are invariant towards size so that a GIS object will not be unusual due to its size only. If the size were a major factor, it could be checked in a pre-processing step easily, using GIS data only.

4.2 Coherence Evaluation

The step *Coherence Analysis* evaluates the attribute space spanned by the *Segment Histogram Attributes* of all objects. Therefore, for each GIS feature the mean and covariance matrix are estimated. This allows determining the squared Mahalanobis distance (section 3) for each GIS object with respect to its own GIS feature. Means and covariance matrices are the *normality model* for the algorithm. It will remain fixed for the rest of the algorithm.

5 EVOLUTIONARY OPTIMISATION

Now that we have gained an understanding of how GIS objects are supposed to be composed of segment classes it is possible to rate an object's normality by calculating its squared Mahalanobis distance. However, an object rating neither provides update areas at sub-object level, nor does it specifically detect errors caused by incorrect object borders. Both can be achieved in our system by evaluating the effects when a segment is switched from one GIS object to a neighbouring object. If the abnormality decreases, the switched segment is considered an update area. The difference of normality before and after the switch is called the *significance* of the switch.

How many configurations have to be tested to check for all possibilities so that all objects may have best normality? Given the application there is no reason to limit switches to just segments that initially touch a border of a GIS object. If major changes took place, an area described by a group of connected segments could have changed.

We formulate the following conditions that a segment must comply to in order to be considered as a candidate for a switch:

- Only segments that touch the border of a GIS object are able to switch GIS objects. Otherwise GIS objects would not be growing from their borders.
- Every segment may only switch once in the whole optimisation. This is done to ensure that a GIS object cannot "move" in the scene. Furthermore within several iteration steps a GIS object might interchange segments with GIS objects that are not immediate neighbours.

Every segment switch results in changes in object borders so that every switch the segments that fulfil the restrictions above change. This makes it very difficult to determine an order of optimisation. Therefore we decided to implement an evolutionary algorithm.

Evolutionary algorithms (De Jong, 2002) are global optimisation algorithms. A evolutionary algorithm starts with an initial (not optimal) solution that is used to generate an initial set of possible solutions (called a *generation*) by simple duplication. Each solution in a generation is then modified independently. The modifications are evaluated to be able to select candidates that are going to be duplicated to form the next generation. Advantages are easy implementation and formulation of optimisation tasks especially for large data sets where brute force algorithms are too complex (like in our case). The downside of a evolutionary algorithms is that the quality of the result cannot be guaranteed. Depending on the application the selection process can be designed to limit this vulnerability to get stuck in an local optimum. Finally, like in all iterative algorithms some ending condition is required.

In the systems flow chart in Figure 2 our evolutionary algorithm is highlighted by a red border. An excerpt of the flow chart showing the evolutionary circle only is shown in Figure 3.

In our system a generation consists of *scene versions*. We are working with a flexible generation size. While at the start of every iteration the number of scene versions is reduced to just a single scene version, within an iteration the number of scene version is increased dynamically.

5.1 Initialisation

The iterating process starts with an evaluation of the current state of the initial scene's configuration. For each object the *Segment Histogram Attributes* (Equation 3) are determined and its normality is evaluated by determining the abnormality value (Equation 2) using the appropriate model (section 4). In the first iteration the *End of Iteration* step is skipped. Finally, the evolutionary circle is ready to start.

5.2 Selection

In this step it is decided which of the available scene versions will be used for further changes. The evaluation of a scene version's configuration is done by summing up the abnormality value (Equation 2) of all of its objects. The scene version with the smallest sum of abnormalities is selected. The scene versions with a higher sum are discarded.

5.3 End of Iteration Check

Deciding when to end an iterative algorithm is always a challenge. We track the sum of the distances of the selected scene over iterations and the iteration is ended when there is no major change over some appropriate number of iterations. We had good results when ending the iteration after as many iterations had happened without any effective change as there are objects in a scene.

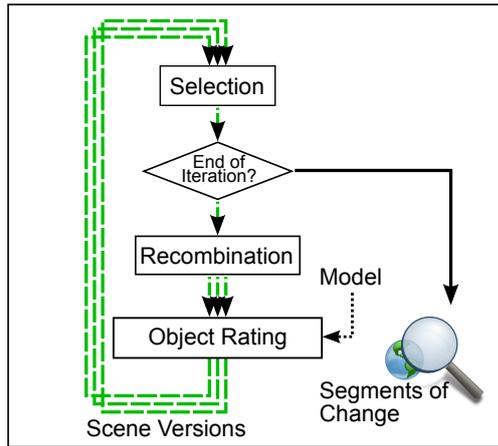


Figure 3: Flow chart showing the system's evolutionary part after initialisation.

5.4 Recombination

In the recombination step a segment is randomly chosen from the incoming scene version. First, the segment is checked against the conditions of candidates for a change (see Section 5). If the segment fails the check, another segment is chosen randomly.

To find the options for a change, all neighbouring segments are taken into account. Only neighbours with another GIS feature than the chosen segment are considered, though. (We want to detect changes in land usage, so there is no point in switching segments between objects of the same features). This also includes segments connected to the same GIS object. Now for each neighbouring segment left, the original scene is duplicated and a switch is performed: First, the GIS object connected to the neighbouring segment is determined. Afterwards the segment's connection to its current GIS object is removed, then a connection to the neighbour segment's object is created.

The original scene version and all new versions are the new generation of scene versions.

5.5 Object Rating

For each scene version, the changed object's *Segment Histogram Attributes* (equation 3) are updated and its normality is evaluated (see Section 5.1). Additionally, for each switched segment, changes in normality are stored since we use it to rank the significance of the switch.

5.6 System Output

The output of the systems are segments that have changed when compared to the initial scene version.

6 RESULTS AND DISCUSSION

In this section we present experimental results for our system. The test area is located in central Germany. IKONOS imagery with 1 m resolution on four channels (*red, green, blue and near infrared*) is available for image analysis to determine segments. The image analysis is not part of this contribution. To highlight the performance of our new approach, we only use basic image analysis results calculated by a *Support Vector Machine* (SVM), (Vapnik, 2000) with RBF kernel. Features are *mean, covariance and Haralick features* (Haralick et al., 1973) in a 25×25 pixel region. For performance reasons, every eighth pixel is classified,

only. Original resolution is gained though nearest neighbouring up scaling. We have trained the SVM with samples industry halls, forest, small houses and grass/cropland. The test area is located in central Germany. GIS data is taken from the German GIS data set ATKIS (ATKIS, 2011). We selected the most prominent GIS features 2111 (settlement), 2112 (industry), 4101 (cropland), 4102 (grassland), 4107 (forest).

Reference Results Unfortunately, there are no benchmark systems available for general GIS quality assessment systems. Thus we had to create a new reference data set to evaluate our system. We decided to label input segments as correct and incorrect by an independent person. Since the whole scene consists of roughly 25000 segments, 100 GIS objects with 1008 segments have been randomly selected for a check. The reference found 42 relevant updates.

To evaluate the performance several error measures are determined. First, we define following terms:

Update Segment Segment that is regarded as area where this GIS object needs to be updated.

True Update Segment Update segment according to the reference.

Potential Update Segment Update segment that has been determined as update segment by the system. In this section several aspects of the system are evaluated. It depends on the evaluation if an segment is regarded an update segment.

Basic measures show to which amount system and the independent reference agree:

True Positives (tp) Number of potential update segments that are true update segments.

False Positives (fp) Number of potential update segments that are not true update segments.

True Negatives (tn) Number of segments that are no potential update segment and to true update segment.

False Negatives (fn) Number of true update segments that are no potential update segments.

6.1 Evaluation ignoring Significance of Switches

Only considering switched segments as potential update segments we get following results:

System \ Reference	Update Segments	No Update Segments
Update Segments	tp: 35	fp: 162
No Update Segments	fn: 7	tn: 804

The basic measures can be used to calculate measures that express the trade off between tp, fp, tn, fn:

$$\text{Precision} = \frac{tp}{tp + fp}, \quad 0 \leq \text{Precision} \leq 1 \quad (5)$$

$$\text{Recall} = \frac{tp}{tp + fn}, \quad 0 \leq \text{Recall} \leq 1 \quad (6)$$

On the one hand, high precision means that only relevant information has to be checked, in our case that many potential update segments are true update segments. It does not tell how many true update segments are not potential update segments. On the other hand, high recall indicates that nearly all true update segments are potential update segments. However, there is no indication about the number of potential update segments without being an true update segment.

Results are: Precision = 0.18, Recall = 0.83.

6.2 Evaluation of Significance of Switches

Switched segments are rated by their significance value (Section 5.5) that can be used to improve precision or recall. Measures of last section can still be determined but they depend on specific significance values so they are expressed as graphs. True/false positives are displayed in Figures 4 (a), (c) and (e). Since true and false positives/negatives are strongly connected, each pair is expressed in a common graph. Each point on the curve results from determining true and false positives/negatives for a specific significance threshold. The value of the threshold is colour coded, the legend can be seen as colour bar at the right axis. To include also segments that have been missed by the system, we assigned them a significance of -0.0001 so they are all equal and lower in significance than any switched segment. For better visualisation, we provide graphs only for switched segments in Figures 4 (b), (d) and (f). It can be easily seen that switched segments with the lowest significance are mostly non-true update segments. In other words: when ignoring segments with a slightly positive significance as potential updates, the number of false positives drops by roughly from 162 to 51 while only three additional true positive objects have been missed. This effect can also be seen in Figures 4 (e) and (f). Precision can be strongly increased without much loss in recall. A similar effect can be seen considering the true and false negative figures which is remarkable considering that only 42 of 1008 segments are true update segments.

In a real application, of course, no graph is available to select an optimal significance value. In this case we propose to check potential update segments starting with the highest significance. As can be seen the frequency of finding true update segments is very high compared to the frequency with low significance.

7 CONCLUSIONS

Updating GIS data is of major interest for many applications, but is a very complex and time consuming task. This paper deals with one major source of changes: incorrect GIS object borders. Existing quality assessment systems follow rather uniform approaches that introduce complex rule sets. We propose to change the procedure from rule based into an automatic evaluation. We demonstrate how a normality model for relations between image analysis results and GIS objects, based on squared Mahalanobis distances, is used to replace knowledge that in existing systems is introduced by rules. GIS objects are sub-divided into segments of specific image analysis classes. Relations between GIS data and segments are described by object wise determining distributions of analysis classes. Finally, evolving the scene by switching segments between GIS objects and monitoring the changes in distributions and normality we identify segments that are probable changes at sub-object level. In addition, proposed updated segments are rated with a significance value.

This paper shows that rule based systems can be replaced by an automatically evaluation. Lowering requirements for human operators and providing areas for proposed updates at sub-object

level the scope of application exceeds the scope of existing systems. We show that using only basic image analysis, 83 % of all update areas could be found with more than every sixth proposed update is a relevant update. Considering the significance value, only checking less than every fourth proposed segment still around 77 % of relevant segments could be found. Our system is not meant to be an alternative to cutting-edge image analysis, though. Without any question, results would further benefit from using more elaborate image analysis methods. However, this also applies in the opposite direction: using our system, image analysis methods can be developed that focus on general image analysis tasks instead of dealing with specific GIS feature definitions.

REFERENCES

- ATKIS, 2011. Amtliches Topographisch-Kartographisches Informationssystem.
- Becker, C., Pahl, M. and Ostermann, J., 2012. Automatic quality assessment of gis data based on object coherence. GEOBIA 2012.
- Buck, O., Peter, B. and Büker, C., 2011. Zwei-skaliger ansatz zur aktualisierung landwirtschaftlicher referenzkuliszen (Ipis). Photogrammetrie - Fernerkundung - Geoinformation PFG, E. Schweizerbart'sche Verlagsbuchhandlung 5(5), pp. 339–348.
- Busch, A., Gerke, M., Grünreich, D., Heipke, C., Liedtke, C.-E. and Müller, S., 2004. Automated verification of a topographic reference dataset: System design and practical results. International Archives of Photogrammetry and Remote Sensing XXXV, Part B2, pp. 735–740.
- De Jong, K. A., 2002. Evolutionary Computation: A Unified Approach. 1st edn, The MIT Press.
- Haralick, R. M., Shanmugam, K. and Dinstein, I., 1973. Textural features for image classification. Systems, Man and Cybernetics, IEEE Transactions on 3(6), pp. 610–621.
- Lacroix, V., Idrissa, M., Hincq, A., Bruynseels, H. and Swartenbroeckx, O., 2006. Detecting urbanization changes using spot5. Pattern Recognition Letters 27(4), pp. 226 – 233. Pattern Recognition in Remote Sensing (PRRS 2004).
- Leignel, C., Caelen, O., Deibeir, O., Hanson, E., Leloup, T., Simler, C., Beumier, C., Bontempi, G., Warzée, N. and Wolff, E., 2010. Detecting man-made structure changes to assist geographic data producers in planning their update strategy. ISPRS Archive Vol. XXXVIII, Part 4-8-2-W9.
- Lu, D., Mausel, P., Brondízio, E. and Moran, E., 2004. Change detection techniques. International Journal of Remote Sensing 25(12), pp. 2365–2401.
- Tan, P.-N., Steinbach, M. and Kumar, V., 2005. Introduction to Data Mining, (First Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Vapnik, V., 2000. The Nature of Statistical Learning Theory. Springer, Berlin.

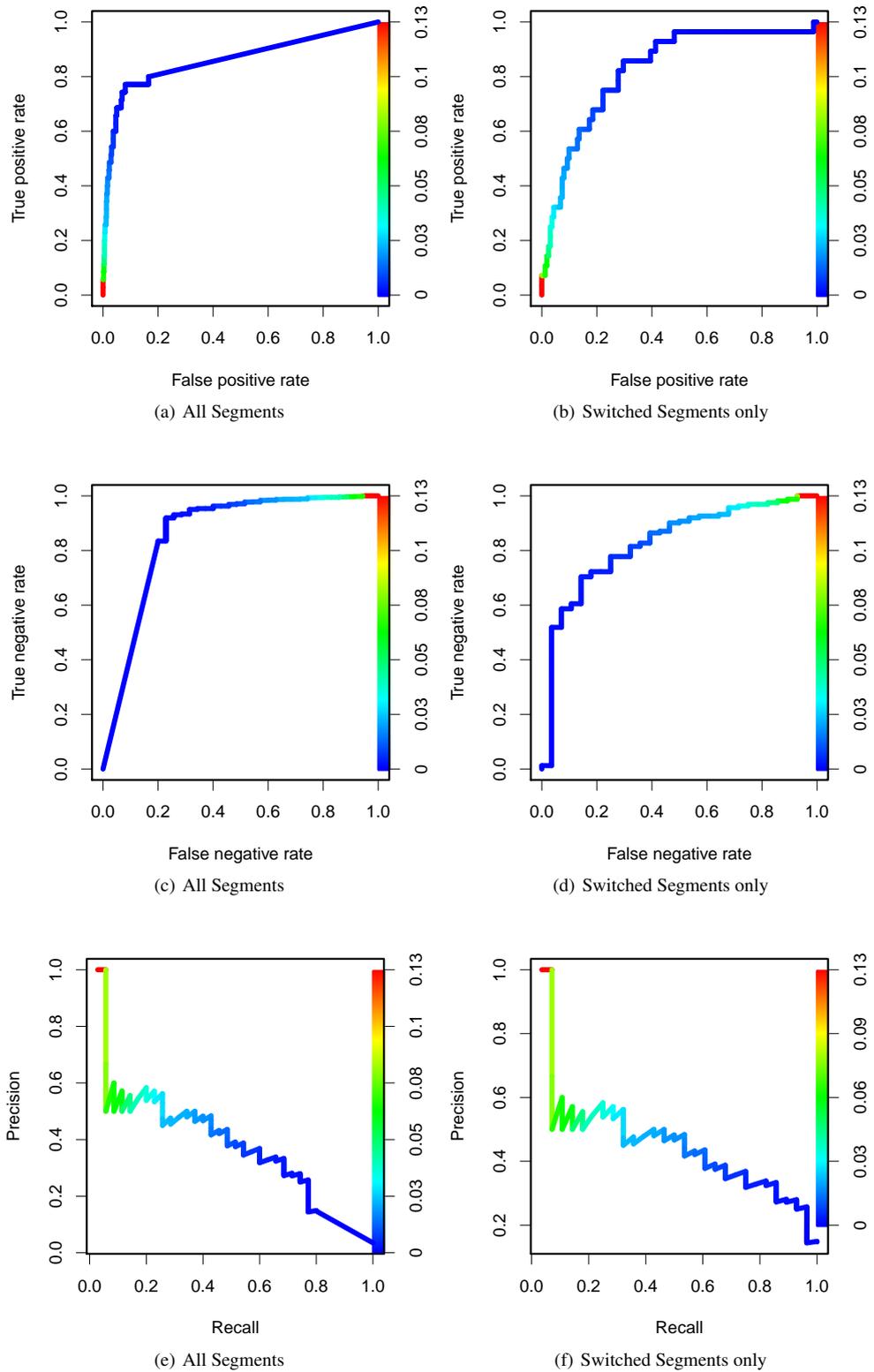


Figure 4: Evaluation results true/false positives/negatives and precision/recall with update segment significance as threshold, not detected changes are given a rating of -0.0001 for visualization purposes.