Sequential Classification in Point Clouds of Urban Scenes

To appear at the 3DPVT 2010 Symposium in Paris France [*]

Olympia Hadjiliadis
Brooklyn College & Graduate Center of CUNY
2900 Bedford Avenue, Brooklyn, NY 11210
ohadjiliadis@brooklyn.cuny.edu

Ioannis Stamos
Hunter College & Graduate Center of CUNY
695 Park Avenue, New York, NY 10065
http://www.cs.hunter.cuny.edu/~ioannis

Abstract

Laser range scanners have now the ability to acquire millions of 3D points of highly detailed and geometrically complex urban sites, opening new avenues of exploration in modeling urban environments. In the traditional modeling pipeline, range scans are processed off-line after acquisition. The slow sequential acquisition though is a bottleneck. The goal of our work is to alleviate this bottleneck, by exploiting the sequential nature of the data acquisition process. We have developed novel online algorithms, never before used in laser range scanning, that perform data classification on-the-fly as data is being acquired. These algorithms are extremely efficient, and can be potentially integrated with the scanner's hardware, rendering a sensor that not only acquires but also intelligently processes and classifies the scene points. This sensor, armed with the proposed algorithms, can classify 3D points in real-time as being in vegetation vs. non-vegetation regions, or in horizontal vs. vertical regions. The former classification is possible by the implementation of sequential algorithms through a hidden Markov model (HMM) formulation, and the latter through the use of a combination of cleverly designed sequential detection algorithms. We envision an arsenal of algorithms of this type to be developed in the future.

1. Introduction

The photorealistic modeling of large-scale scenes, such as urban structures, has received significant attention in recent years [1, 15, 21]. In this paper we describe algorithms that use mathematical techniques from the areas of applied probability and sequential statistics for the classification of the highly complex 3D data that laser scanners acquire, and for the detection of abrupt changes. Decisions are made on-the-fly as data is being acquired by the sensor, by extremely efficient algorithms that yield very robust results. Our goal is the development of an intelligent sensor that not only acquires but also provides meaningful data classification and detection in real time.

The collection of high-resolution point-clouds from laser scanners is currently a slow process. For instance collecting 10 million 3D points in an urban area can take a significant amount of time (in the order of hours using the Leica ScanStation2 scanner [8]). Point-cloud segmentation algorithms use all the acquired data in order to split the scene into major surfaces. In a real-time application though decisions have to be made instantly. In these cases we need to classify as fast as possible. Thus, in this setting sequential classification techniques become relevant [13, 17]. In the same framework we wish to detect the spatial locations of abrupt changes in the environment (i.e. changes between street to facade level, etc.).

Our approach thus leads to more intelligent sensing and is beneficial for higher-level recognition and decision processes. As one example, the user may want to acquire high-resolution data from facades, and low-resolution data on ground-level, or in vegetation areas. In order to achieve this a classifier that directs the scanning process is needed. This classifier can classify and detect major surfaces from an initial low-resolution scan of the scene. Therefore, the system can then acquire high-resolution scans in the areas of interest (for instance facades), thus leading to lower cost of acquisition in terms of time and power consumed. Other examples include real-time detection and recognition of objects in a 3D-scene, as well as 3D modeling.

The current technology in laser range-scanning acquires 3D data in an online sequential fashion. Our techniques are able to take advantage of this fact, and thus make more accurate decisions sooner while data is being collected, rendering a more efficient data acquisition system.

Our contribution is the development of novel real-time algorithms for automatic classification during range data acquisition. In particular we present sequential on-line algorithms for: (a) the reliable classification between vegetation and non-vegetation data, via a novel hidden Markov model formulation. Note that range data in the areas of vegetation (trees, etc.) pose significant challenges to current segmentation algorithms, and (b) the reliable classification between

[*] This work has been supported in part by the following NSF grants: DMS-IGMS-0929317, IIS-0915971, CCF-0916452 and MRI CNS-0821384.

vertical, and horizontal surfaces in the non-vegetation data regions. A simple region-growing algorithm that uses the results of the above classifiers and reliably separates the data into connected regions of vertical and horizontal surfaces. We envision an arsenal of sequential algorithms that provide classifications of different types in the future.

2. Previous work

There is a variety of range image segmentation techniques including edge detection [3, 18], region growing [4, 5], polynomial surface fitting [4, 7], and graph-cut approaches [20]. These methods do not associate any classes with the extracted segments, and process the data offline. Also, most of them require the computation of surface normals from the data (surface normals would be inconsistent in non-smooth areas). There is also literature on the topic of classification of 3D point-clouds using Markov network models [2, 10, 11]. These techniques assume that the data becomes available all at once, as opposed to sequentially. They also require training. On-line detection techniques have been used in 3D computer vision mainly in the context of a moving sensor [16, 19], as opposed to a steady high-resolution laser-scanner. The goal is to separate between two states: drivable vs. non-drivable terrain. In [19] a hidden Markov model is used to achieve this goal. Our work, on the other hand, is able to classify data on-the-fly into three regions, one of them being vegetation. In order to achieve this, we use a cleverly selected measurement model, which reduces the dimensionality of the data. To this data we incorporate hidden Markov models and are able to apply sequential algorithms.

3. Measurement Model

In order to describe the sequential classification method, let us assume that we are sequentially observing streams of data in a discrete-time fashion, denoted by

$$\begin{bmatrix} \text{Stream 1:} & \mathbf{Z_{1,1}} & \mathbf{Z_{1,2}} & \dots \\ \text{Stream 2:} & \mathbf{Z_{2,1}} & \mathbf{Z_{2,2}} & \dots \\ \dots & \dots & \dots & \dots \\ \text{Stream M:} & \mathbf{Z_{M,1}} & \mathbf{Z_{M,2}} & \dots \end{bmatrix}$$

Each of the $\mathbf{Z_{i,j}}$ are vectors whose elements include the 3D coordinates and the reflectance value of an acquired 3D point. I.e. $\mathbf{Z_{i,j}} = [x_{ij}, y_{ij}, z_{ij}, r_{ij}]$. Let us suppose that we are given a reference plane Π that corresponds to the horizontal direction. That means that we also know the vertical axis $\mathbf{z}$. Knowledge of the vertical direction (axis $\mathbf{z}$) is provided by many laser scanners, or can be easily acquired via hardware. Most robotics application (for instance [19]) make this assumption as well.

Let us denote by $\mathbf{X_{i,j}} = [x_{ij}, y_{ij}, z_{ij}]$ the vector of 3D coordinates. These coordinates are directly extracted by the distance and the angles α and β of the emitted laser beam with respect to the local coordinate system of the laser scanner, whose location is $(0, 0, 0)$ (see Fig. 1 for explanation).

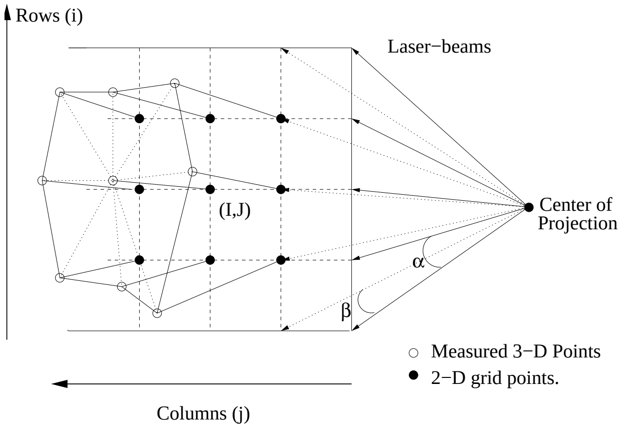

Figure 1. Description of laser-scanning mechanism. A set of laser-beams are emitted toward the scene in a sequential fashion. The angle between the beams in the vertical direction is α and in the horizontal β. These angles are fully controllable by the user. The device measures the distance between the center of projection and the surface point along the beam direction. Each point is acquired sequentially in a raster-scanning order (i.e. first column then second column, etc. - within each column first row, second row, etc.).

The scanner is placed on a steady platform and takes measurements of distance to the closest surface sequentially by emitting a laser beam. The movement of the beam is fully known and controlled by the user. Note that it is also possible that no distance is going to be measured when the laser hits transparent or highly specular surfaces, or when the measured point is at a distance bigger than a threshold (300 meters in our setting).

We now define $D_{i,k} = X_{i,k+1} - X_{i,k}$ (difference of two successive measurements in a given scanline i), and

V_{ik}**:** the angle of the vector $D_{i,k}$ with the pre-determined $\mathbf{z}$ axis (0 to 180 degrees).

$sV_{ik} = s_{ik} * V_{ik}$**:** the sign of the dot product between the vectors $D_{i,k}$ and $D_{i,k-1}$, multiplied by V_{ik}.

From the above definitions all the above quantities are scalars. As we obtain sequential observations along any given fixed stream i, we can form the sequence of scalar random variables $\{sV_{ik}\}$, $k = 1, 2, \dots$. Univariate detection and classification schemes can then be applied to it as described in the next section.

To be more specific, if the measured points $\mathbf{X_{i,k}}$ $k = 1, 2, \dots$ are on the horizontal plane then the measurements sV_{ik} are expected to assume values around $90°$. In the case that the measured points $\mathbf{X_{i,k}}$ $k = 1, 2, \dots$ are in an almost vertical surface then the sequence of measurements sV_{ik} will produce values that are expected to be approximately $90°$ less than the ones on the horizontal plane. If the scanning plane (plane defined by the column i and the center of projection in Fig. 1) and the vertical surface are perpendicular to each other then each of the above measurements is expected to be around 0. In other cases, they would be higher than 0. The values of these measurements can provide the orientation of the vertical surface on the horizontal plane. An example of these values along scanlines can be seen in Fig. 2. The other important characteristic of the sequence of signed angles sV_{ik}, is that this sequence generates a distinguishable pattern in vegetation areas. In these areas, the signed angles alternate rapidly between positive and negative values. Note that a negative value in sV_{ik} indicates a change in direction between vectors $D_{i,k}$ and $D_{i,k-1}$. In areas of solid surfaces the sV_{ik} are always positive. Fig. 3 displays the sequence of $D_{i,k}$ that connect successive 3D points. In the vegetation areas the relative orientation between these vectors changes, rapidly. This pattern is captured via a hidden Markov model described in the following sections.

4. Modeling

Each scanline is considered as a stream of observations that can be treated in a sequential manner. For each scanline i each measurement $X_{i,k}$ is classified as either horizontal, vertical, or vegetation data. The vegetation classification is achieved via an online algorithm which detects a change from one three-state HMM to another.

4.1. Vertical/Horizontal Modeling

In order to distinguish between the horizontal and the vertical direction we begin by recognizing that the difference between a "horizontal" and a "vertical" surface is that the difference in the angles of the former to the z-axis and the angles of the latter to the z-axis is expected to be around $90°$. In particular, it is natural to expect that the former angles are approximately $90°$ (with some deviations due to an inclined ground) while the latter ones are approximately $0°$ (with some possible deviations too). We thus think of the sequence of sV_{ik} $k = 1, 2, \ldots$, within each fixed scanline i, as realizations of random variables of the same distribution (in this case unit variance Gaussians) with either a mean of $\mu_0 = 90°$ or a mean of $\mu_1 = 0°$. Therefore a change from a "horizontal" to a "vertical" surface is characterized by a shift in the mean of these observations.

4.2. Hidden Markov Modeling

Scene areas that include vegetation and trees produce a unique behavior in the sequence of measurements along each scanline. In these areas the signed angle $sV_{i,k}$ (see Sec. 3) measurements alternate rapidly between negative (around $-90°$ degrees) and positive angles, or they are stable on non-negative values, either around $10°$ or $90°$ degrees. Figs. 2 and 3 provide an example.

The above behavior can be captured by hidden Markov models. In particular, we can introduce a sequence of state variables $\{u_k\}$ $k = 1, 2, \ldots$ which represent the state of the hidden Markov model at each observation k. We can then distinguish three states, namely 1, 2 and 3, as follows. When in state $u_k = 1$, the signed angles $sV_{i,k}$ can be seen as random variables distributed around a mean of approximately $\mu_1 = 90°$. Similarly, when in state $u_k = 2$, the signed angles $sV_{i,k}$ can be seen as random variables distributed around a mean of approximately $\mu_2 = 10°$. Finally, when in state $u_k = 3$, the signed angles $sV_{i,k}$ can be seen as random variables distributed around a mean of approximately $\mu_3 = -90°$. The selection of the above states will become clear in the sequel.

To formally describe a discrete HMM we need the following parameter triple $\lambda = (A, B, \pi)$, where

$$A = [a_{lr}] = [P(u_{k+1} = r | u_k = l)],\ l, r = 1, 2, 3 \qquad (1)$$

is the state transition matrix of the underlying Markov chain (i.e. a_{lr} is the probability of moving from state l to state r) and where

$$\pi = [\pi_l = P(u_1 = l)],\ l = 1, 2, 3 \qquad (2)$$

is the initial distribution of the underlying Markov states and

$$B = [b_{lk}] = [f_{sV_{ik}}(x | u_k = l)],\ l = 1, 2, 3 \qquad (3)$$

is the conditional distribution of the observation sV_{ik} (in scanline i) given the state u_k.

For reasons that will become apparent in the sequel we introduce the forward variable of an HMM as follows:

$$\alpha_k(l) = p(sV_{i1}, \ldots, sV_{ik}, u_k = l | \lambda),\ l = 1, 2, 3,$$

which represents the joint distribution of the data up to observation k within scanline i and the state at that time given the parameter triple λ.

It is easily checked that the following recursion holds for the forward variable

$$\alpha_{k+1}(r) = \left[\sum_{l=1}^{3} \alpha_k(l) a_{lr} \right] b_{r(k+1)},\ r = 1, 2, 3 \qquad (4)$$

with initial condition

$$\alpha_1(r) = \pi(r) b_{r1},\ r = 1, 2, 3. \qquad (5)$$

In our set-up we model the conditional distribution of the $\{sV_{ik}\}$, $k = 1, 2, \ldots$, given the state variable u_k by a Gaussian distribution with means μ_l, where $l = 1, 2, 3$ as above. That is, the B in Eq. 3 is a matrix of Gaussian probabilities.

Fig. 2 provides an intuition to our thinking. The part of the data before observation 189 could be captured by a two state HMM with states 1 and 2 or by a three state HMM with states in which the transition probability to and from the third state is simply 0. The transition from state 1 to 2 and vice versa appears to be relatively slow or unlikely. On the other hand after observation 189 (i.e. after the start of the vegetation region), the transition between all three states appears relatively fast or likelier.

We can thus detect the beginning of vegetation by detecting a change from an HMM model with “unlikely” transitions to one with “likely” transitions. In particular, let $\lambda_0 = (A_0, B, \pi)$ and $\lambda_1 = (A_1, B, \pi)$ be the parameter of the HMM before and after the change. We set

$$A_0 = \begin{bmatrix} p_1 & p_2 & 1-p_1-p_2 \\ q_1 & q_2 & 1-q_1-q_2 \\ r_1 & r_2 & 1-r_1-r_2 \end{bmatrix}$$

where $p_1 = 0.9, p_2 = 0.1, q_1 = 0.1, q_2 = 0.9, r_1 = r_2 = 0$, and A_0 is the matrix of Eq. 1 corresponding to the initial HMM model with parameter triple λ_0. The selected values for the entries of A_0 thus ensure that the HMM is relatively stable at state 1 or state 2.

We also set

$$A_1 = \begin{bmatrix} p_1' & p_2' & 1-p_1'-p_2' \\ q_1' & q_2' & 1-q_1'-q_2' \\ r_1' & r_2' & 1-r_1'-r_2' \end{bmatrix}$$

where $p_1' = p_2' = q_1' = q_2' = r_1' = r_2' = 1/3$, and A_1 is the matrix of Eq. 1 corresponding to the alternative HMM model with parameter triple λ_1. This transition probability matrix A_1 captures the fact that transitions from state to state are “likelier” after the change and the fact that negative values in the signed angles are also possible after the change. The two HMM models with parameters λ_0 and λ_1 give rise to two distinct sequences of forward variables (see Eqs. 4 and 5) which we denote by $\{\alpha_k^0(r)\}$ and $\{\alpha_k^1(r)\}$, $k = 1, 2, \ldots$ respectively.

5. Sequential Classification

We will begin by describing two main algorithms used in sequential classification and quickest detection. The first is known as the sequential probability ratio (SPRT) test and the second one is known as the cumulative sum (CUSUM) test. For an overview of these tests please refer to [13]. For simplicity in exposition we will drop the subscript i, since we consider sequential observations $\{sV_k\}$ $k = 1, 2, \ldots$ within a specific scanline. The classical problem of sequential identification addresses the issue of determining which of two simple hypotheses (H_0 or H_1) concerning the distribution of our data is true, subject to pre-specified levels of type I (probability of rejecting H_0 when H_0 is true) and type II errors (probability of rejecting H_1 when H_1 is true). In particular, let

$$\begin{array}{ll} & H_0: \quad sV_k \sim f_0,\ k = 1, 2, \ldots \\ \text{versus} & \\ & H_1: \quad sV_k \sim f_1,\ k = 1, 2, \ldots \end{array} \qquad (6)$$

where f_0 describes the distribution of the data under H_0, and f_1 under H_1. The algorithm we construct determines the first instance (the number of observations necessary) at which to stop and declare which of the two hypotheses to select subject to a pre-specified level α and β of type I and type II errors respectively. In [17] it is seen that the fastest algorithm that addresses this problem is known as the sequential probability ratio test (SPRT) and it consists of stopping at observation number

$$N = \inf\left\{ n; \frac{f_1(sV_1, \ldots, sV_n)}{f_0(sV_1, \ldots, sV_n)} \notin (A, B) \right\},$$

(where (A, B) is the open interval from A to B) and declaring that hypothesis H_1 (or H_0) is true if exit occurs at B (or at A). The functions f_1 and f_0 are the joint distributions of $(sV_1, \ldots, sV_n)$ under H_1 and H_0 respectively. The constants A and B are related to α and β through the equations

$$1 \leq B \simeq \frac{1-\beta}{\alpha},$$
$$1 \geq A \simeq \frac{\beta}{1-\alpha}.$$

In the special case in which the $\{sV_k\}$, $k = 1, 2, \ldots$ are i.i.d. (independent and identically distributed) unit variance Gaussian random variables with means μ_0 and μ_1 under each of the hypotheses H_0 and H_1 respectively, N simplifies to

$$N = \inf\left\{ n; (\mu_1 - \mu_0) \sum_{k=1}^{n} \left(sV_k - \tfrac{1}{2}(\mu_1 + \mu_0) \right) \notin (a, b) \right\},$$

where $a = \ln A < 0$ and $b = \ln B > 0$ and a decision in favor of H_1 (or H_0) is taken if exit occurs through b (a).

The problem of quickest detection is concerned with determining the first instance τ at which the mechanism by which observations are generated changes. That is, the first time that the distribution of the random variables $\{sV_k\}$ changes from f_0 to f_1. One of the most well-known algorithms which addresses this problem is known as the CUSUM algorithm and was introduced in [12]. It is also known to enjoy very strong optimality properties as later proved in [9]. For an overview of these results please refer to [13]. To mathematically describe the problem, we assume that

$$\begin{array}{rl} sV_1,\ sV_2,\ \ldots,\ sV_{\tau-1} & \sim\ f_0, \\ sV_\tau,\ sV_{\tau+1},\ \ldots & \sim\ f_1. \end{array}$$

where f_0 and f_1 are defined as above. The CUSUM algorithm stops and declares a change at the first n for which

$$T = \inf\left\{ n; \sup_{1 \leq \tau \leq n} \ln \frac{f_1(sV_\tau, \ldots, sV_n)}{f_0(sV_\tau, \ldots, sV_n \mid sV_{\tau-1}, \ldots, sV_1)} \geq h \right\}.$$

In the special case in which the sequence of random variables are i.i.d. before and after the change point τ the form of the CUSUM simplifies to $\inf\{n; S_n \geq h\}$ with

$$S_n = \max\{0, S_{n-1} + g(sV_n)\},\ S_0 = 0, \qquad (7)$$

with $g(sV_n) = \ln \frac{f_1(sV_n)}{f_0(sV_n)}$.

In particular, under the assumption that $\{sV_k\},\ k = 1, 2, \ldots$ are i.i.d. unit variance Gaussian random variables with mean μ_0 and μ_1 under each of the hypotheses H_0 and H_1 respectively, the sequence $\{S_n\}$ of Eq. 7 simplifies to

$$S_n = \max\left\{0, S_{n-1} + sV_n - \frac{\mu_1 + \mu_0}{2}\right\},\ S_0 = 0, \qquad (8)$$

when $\mu_1 > \mu_0$ and

$$S_n = \max\left\{0, S_{n-1} - \left[sV_n - \frac{\mu_1 + \mu_0}{2}\right]\right\},\ S_0 = 0, \qquad (9)$$

when $\mu_1 < \mu_0$. Now recall that the CUSUM algorithm is a result of repeated SPRT's with left boundary $a = \ln A = 0$. In particular the CUSUM is the first time that repeated SPRT's exit the interval (a, b) on the right [14] with $b = h$.

It is important to notice that in the case in which the sequence of $\{sV_k\}\ k = 1, 2, \ldots$ is modeled through a hidden Markov model, the simplification of Eq. 7 fails to hold. In particular,

$$\ln \frac{f_1(sV_\tau, \ldots, sV_n)}{f_0(sV_\tau, \ldots, sV_n|\ sV_{\tau-1}, \ldots, sV_1)} = \sum_{i=\tau+1}^{n} \ln \frac{f_1(sV_i|\ sV_{i-1}, \ldots, sV_\tau)}{f_0(sV_i|\ sV_{i-1}, \ldots, sV_1)} + \ln \frac{f_1(sV_\tau)}{f_0(sV_\tau|\ sV_{\tau-1}, \ldots, sV_1)}$$

Since the CUSUM algorithm is a result of repeated applications of the SPRT, this gives rise to the following CUSUM-like modification of Eq. 7 as suggested in [6]:

$$S_n = \max\{0, S_{n-1} + g(sV_n, \ldots, sV_k)\},\ S_0 = 0, \qquad (10)$$

where

$$g(sV_n, \ldots, sV_k) = \ln \frac{f_1(sV_n|\ sV_{n-1}, \ldots, sV_k)}{f_0(sV_n|\ sV_{n-1}, \ldots, sV_k)}$$

and sV_k is the last sample after the last reset, i.e. $S_{k-1} = 0$.

Similarly, under the hidden Markov model the SPRT may be written in the form

$$N = \inf\left\{n; \sum_{i=1}^{n} \ln \frac{f_1(sV_i|sV_{i-1}, \ldots, sV_1)}{f_0(sV_i|sV_{i-1}, \ldots, sV_1)} \notin (a, b)\right\}, \qquad (11)$$

with $a = \ln A$ and $b = \ln B$.

To conclude this section, we notice that it is possible to express $f_j(sV_k|sV_{k-1}, \ldots, sV_1),\ j = 0, 1$ in terms of the forward variable of an HMM by noticing that

$$f_j(sV_k|sV_{k-1}, \ldots, sV_1) = \frac{\sum_{l=1}^{3} \alpha_k^j(l)}{\sum_{l=1}^{3} \alpha_{k-1}^j(l)}, \qquad (12)$$

for each $j = 0$ and $j = 1$.

6. Region classification algorithms

In this section we will apply the algorithms developed in the previous section to classify horizontal vs. vertical surfaces and vegetation vs. non-vegetation regions. Both algorithms are extremely efficient and easy to implement. They require O(1) operations per observation, and O(1) memory space. They can thus be implemented at the sensing level without slowing down the acquisition process.

6.1. Horizontal vs. Vertical classification

Using the Gaussian modeling of Sec. 4.1, it is now possible to apply the CUSUM algorithm of Eq. 9 to detect a change from a horizontal to a vertical surface by using $\mu_0 = 90^\circ$ and $\mu_1 = 0^\circ$, which leads to $\frac{\mu_0+\mu_1}{2} = 45^\circ$. The CUSUM algorithm is simply the update of S_n according to Eq. 9, and the declaration of a change when S_n exceeds threshold h.

There are a number of important points to be made here. To begin, we notice that when we have an inclined surface we may have a μ_0 greater than or less than 90°. The mean that will then characterize a vertical surface will deviate from μ_0 by approximately 90°. In other words, we will have $\mu_1 = \mu_0 - 90^\circ$. Now, the proposed algorithm will still work as long as $E_{f_0}[V_k] - 45^\circ > 0$, while $E_{f_1}[V_k] - 45^\circ < 0$.

In practice, we have used a threshold $h = 20$, which makes the algorithm sensitive enough to capture deviations from the horizontal surface.

Once a detection from a horizontal to a vertical surface is made, it is also important to detect the end of the vertical surface. This is achieved by using the opposite CUSUM algorithm as in Eq. 8. In order to make this algorithm more sensitive we have used a value of $\frac{\mu_0+\mu_1}{2} = 30^\circ$ and a threshold of $h = 15$. The first instance at which this CUSUM algorithm stops marks the end of the vertical surface and the detection of the horizontal surface.

By using a combination of these algorithms it thus becomes possible to distinguish horizontal and vertical regions.

6.2. Vegetation detection and classification

As discussed in section 4.2 we can capture the beginning of vegetation as the first instance in a sequence of our data at which there is a change from an HMM model with "unlikely" transitions to one with "likely" transitions. We are now in a position to describe the CUSUM-like algorithm of Eq. 10, which detects a change from an HMM with $\lambda_0 = (A_0, B, \pi)$ to an HMM with $\lambda_1 = (A_1, B, \pi)$ as suggested in [6].

1. Set $n = 1$. Set $S_0 = 0$, and $\alpha_0^0(l) = \alpha_0^1(l) = 1$ for $l = 1, 2, 3$ (this is necessary for the first iteration of the update of step 3).

2. Initialize the forward variables using $\alpha_n^j(l) = \frac{1}{3}\phi(\frac{sV_1 - \mu_l}{\sigma_l})$, $l = 1, 2, 3$ for each $j = 0$ and $j = 1$, where ϕ denotes the Gaussian kernel.

3. Update using

$$S_n = S_{n-1} + \ln \frac{\sum_{l=1}^{3} \alpha_n^1(l)/\sum_{l=1}^{3} \alpha_{n-1}^1(l)}{\sum_{l=1}^{3} \alpha_n^0(l)/\sum_{l=1}^{3} \alpha_{n-1}^0(l)}.$$

4. If $S_n > b$, with $b = h$, stop and exit; If $S_n < a$, with ($a = 0$), set $S_n = 0$, $n = n + 1$ and go to 2. If $a < S_n < b$ ($a = 0,\ b = h$) continue.

5. Update $\alpha_{n+1}^j(r)$ for each $j = 0$ and $j = 1$ according to the recurrence of Eq. 4. Set $n = n + 1$. Goto step 3.

We set the threshold h of the above CUSUM-like algorithm to $h = 10$ and call the observation number on which it exits N. The observation N signifies the possible beginning of the region of vegetation. We need to verify that this is indeed the beginning of vegetation, and also detect the end of that region, once its beginning has been identified. Now the problem is turned into one of testing two hypotheses, namely H_0 under which the HMM with parameter λ_0 is true (corresponding to non-vegetation) versus H_1 under which the HMM with parameter λ_1 is true (corresponding to vegetation).

This is achieved by modifying the CUSUM-like algorithm into an SPRT algorithm, with a being negative (set to $a = -5$) and $b = h$ (set to $h = 10$ in our experiments). We start from observation $N + 1$ (since at N the CUSUM went off), and perform steps 1,2,3 and 5 as above, while modifying step 4 to an SPRT algorithm, as follows:

4. If $S_n > b$ or $S_n < a$ we stop and exit. Otherwise, we continue on Step 5.

We call the observation number on which the SPRT algorithm exits $N+M$. If $S_{N+M} < a$ we repeat the sequence of the above two algorithms (CUSUM-like followed by SPRT) starting at $N + M + 1$. Otherwise if $S_{N+M} > b$ we classify the M points between observations N and $N + M$ as vegetation, and continue running a sequence of SPRT algorithms until an exit at a occurs, at which point we declare the end of the vegetation region. Please note that if the CUSUM-like algorithm never goes off, no vegetation is declared at that scanline. Similarly, if the SPRT algorithm never goes off, again no vegetation is declared.

The parameters σ_1, σ_2 and σ_3 are first set to nominal values and then σ_1 and σ_2 estimated using a trimmed variance estimator over the regions of horizontal and vertical surfaces. The parameter σ_3 is then set equal to σ_1. It is interesting to point out that the algorithm is fairly robust to changes in these parameters.

6.3. Region Growing

The classification results for each 3D point allows for a fast sequential region growing algorithm. This sequential algorithm places clusters of connected 3D points of the same class. The algorithms runs in parallel with the aforementioned classification, i.e. the cluster of each point is decided just after the point is acquired by the sensor. The algorithm works briefly as follows. Let us provide labels to clusters sequentially, starting from 0. At each point, we call l_{num} the label of the current cluster of points. Let us consider the k-th 3D point in scanline i (i.e. point $P_{i,k}$). Let us suppose that the class of this point has been decided to be $C_{i,k}$, where $C_{i,k}$ is either H (horizontal), V (vertical), or T (vegetation). If its class is T then we proceed to the next point (i.e. vegetation points are not placed on connected components due to their irregular structure). Due to the sequential nature of data-acquisition the class $C_{i,k-1}$ of the previous point $P_{i,k-1}$ in the scanline, and the class $C_{i-1,k}$ of the corresponding point in the previous scanline (in case $i > 1$) have already been determined. Note, also that it is possible that one or both of the points $P_{i,k-1}$, and $P_{i-1,k}$ can be missing, meaning that the sensor did not provide any measurement due to to high-specularity, or distance greater than a limit. Consider the following boolean values $A_{i,k-1} = (C_{i,k} == C_{i,k-1}) \wedge (|P_{i,k} - P_{i,k-1}| < L_1)$, and $A_{i-1,k} = (C_{i,k} == C_{i-1,k}) \wedge (|P_{i,k} - P_{i-1,k}| < L_2)$. Each one of the booleans is true iff the current class agrees with the class of the neighboring point, and the distance between the two points is below a user-specified limit. If, now, both $A_{i,k-1}$ and $A_{i-1,k}$ are false (i.e. the class $C_{i,k}$ of the currently sensed (and not missing) point is different than both $C_{i,k-1}$ and $C_{i-1,k}$, or the distance of the point between its neighbors is great), then a new cluster with label $l_{num} = l_{num} + 1$ is being created. The point $P_{i,k}$ is placed on this new cluster. Otherwise, if just one of $A_{i,k-1}$ and $A_{i-1,k}$ is true, then the point $P_{i,k}$ is added to the cluster of the neighbor that makes the boolean true. Finally if both $A_{i,k-1}$ and $A_{i-1,k}$ are true, then clusters of the neighbors are merged (if they are not already the same). The point $P_{i,k}$ is added to the merged cluster. Note that no actual merging needs to take place (that would be an expensive operation), but a fast union-find data structure can be used for keeping track of the merged sets. Results of region growing can be seen in Fig. 5. In summary the region growing algorithm, can in real-time provide vertical vs. horizontal clusters of 3D points. This information can be used by higher-level recognition processes.

7. Results and Conclusions

We have tested our algorithms on a number of scans in busy urban settings, that include acquisition of ground levels, vegetation, moving objects, and building structures. Our acquisition device is the Leica ScanStation2 [8]. This is a time-of-flight scanner with a spherical field of view, that generates a sequence of 3D points at a distance of up to 300m and accuracy of 5mm per point. Visualizations of our results can be seen in Figs. 2, 4, and 5. The results are very accurate. Very small mis-classification can appear within

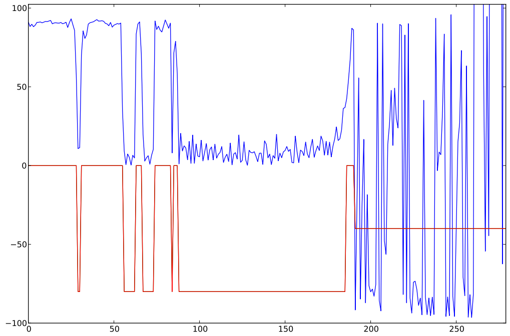

Figure 2. A plot of the signed angles sV_{ik} for a scanline (see Sec. 3). The sequence of sV_{ik} is shown in blue. Overlaid is a red plot of three different values that correspond to the class of each measured point as detected by our algorithms (0 representing horizontal surface, -75 representing vertical surface, and -45 representing vegetation). The regions where the signed angles are close to 90° are horizontal, the ones that are close to 0° are vertical. The regions in which there is a high variation between positive and negative angles are vegetation areas.

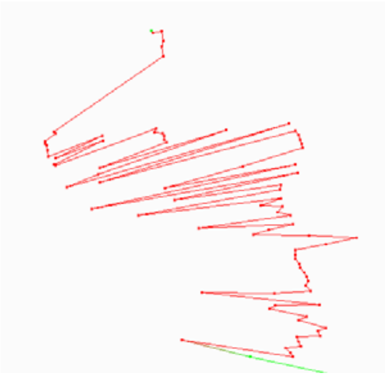

Figure 3. Sequence of 3D points along a specific scanline i in a vegetation region. The vectors connecting successive 3D points are also shown. The signed angle sV_{ik} between successive vectors changes rapidly between positive and negative values. A negative value signifies a change in orientation between successive vectors. See Fig. 2 for the actual values.

vegetation regions, though. Also there is a small delay in the detection of the end of the vegetation regions. Note that our algorithms have not yet been integrated on the sensor's hardware. For more results please visit the homepage of the second author. In summary, in this paper we presented extremely efficient algorithms (O(1) operations per point, O(1) space) for the real-time sequential classification of range points of urban scenes. These algorithms can lead to new intelligent sensors, that not only acquire, but also process and classify the data on-the-fly. We envision the development of more classification algorithms on the same framework.

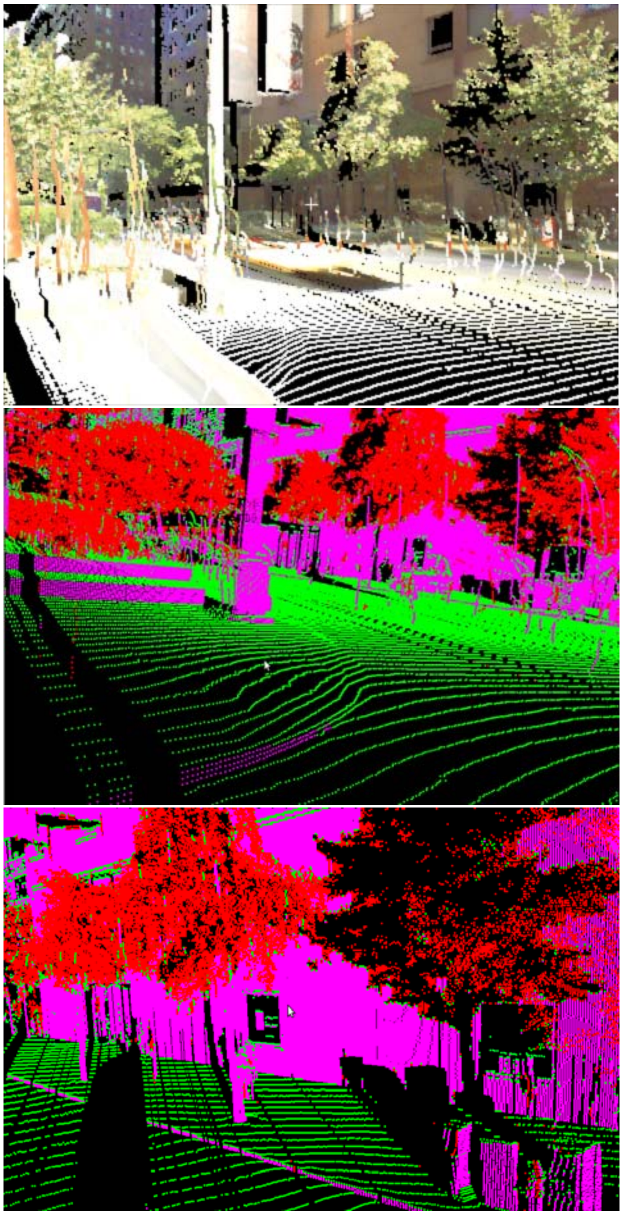

Figure 4. (Top) Texture-mapped 3D point cloud of urban scene. Note presence of vegetation, and of vertical spikes. These spikes correspond to moving objects (people in most cases). (Middle-Bottom): Two views of the scan, where the three class separation into horizontal (green color), vertical (purple color) and vegetation (red color) is shown. Note that each point is being assigned to one of the three classes on-line (i.e. just after it has been acquired by the sensor). No off-line post-processing has been performed to the dataset. The results are very robust, and even small vertical surfaces (steps) are extracted reliably. Vertical spikes are correctly classified. Vegetation regions are very robustly classified as well. Note that some parts of solid objects that appear after or between vegetation regions will be mis-classified as vegetation regions, due to the decision delay of the on-line algorithms.

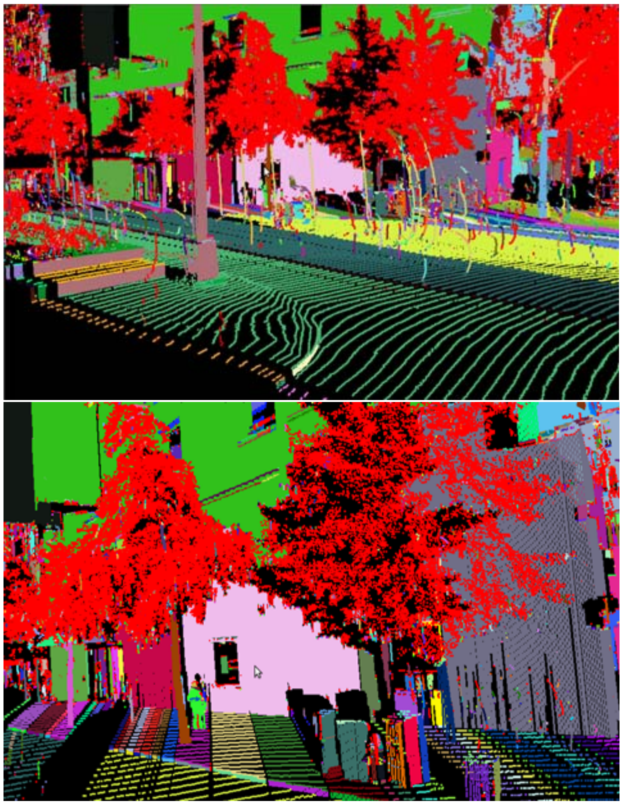

Figure 5. Region growing results (see Sec. 6.3). The images show the sequentially generated connected clusters of vertical and horizontal regions. Vegetation points are not placed on connected components (colored red in the images). The fast sequential algorithm uses the three established classes for each range point. No off-line post-processing has been performed to the dataset. The robust initial classification allows for fast and correct separation of regions. Note that missing data between scanlines may separate connected regions. The correct classification of vegetation, allows for robust distinction between vertical and horizontal regions. Without the vegetation detection, parts of the non-solid regions would have been misclassified. Note the classification of small vertical steps, and of the trunks of trees. Higher-level processes may use this information (for instance: a solid trunk with vegetation on top of it can be recognized as a tree).

References

[1] P. K. Allen, I. Stamos, A. Troccoli, B. Smith, M. Leordeanu, and S. Murray. New methods for digital modeling of historic sites. *IEEE Computer Graphics and Applications*, 23(6):32–41, 2003. 1

[2] D. Anguelov, B. Taskarf, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of markov random fields for segmentation of 3D scan data. In *Computer Vision and Pattern Recognition*, volume 2, pages 169–176, June 2005. 2

[3] O. R. P. Bellon and L. Silva. New improvements to range image segmentation by edge detection. *IEEE Signal Processing Letters*, 9(2):43–45, February 2002. 2

[4] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on PAMI*, 10(2):167–192, March 1988. 2

[5] C. Chao and I. Stamos. Range image segmentation for modeling and object detection in urban scenes. In *the Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, Montreal, Canada, August 2007. 2

[6] B. Chen and P. Willett. Detection of hidden markov model transient signals. *IEEE Transactions on Aerospace and Electronic Systems*, 36(4):1253–1268, October 2000. 5

[7] G. L. D. Marshall and R. Martin. Robust segmentation of primitives from range data in the presence of geometric degeneracy. *IEEE Transactions on PAMI*, 23(3):304–314, March 2001. 2

[8] Leica Geosystems. http://hds.leica-geosystems.com/. 1, 6

[9] G. V. Moustakides. Optimal stopping times for detecting changes in distributions. *Annals of Statistics*, 14(4):1379–1387, 1986. 4

[10] D. Munoz, J. Bagnell, N. Vandapel, and M. Hebert. Contextual classification with functional max-margin markov networks. *Computer Vision and Pattern Recognition*, pages 975–982, 2009. 2

[11] D. Munoz, N. Vandapel, and M. Hebert. Directional associative Markov Network for 3-D point cloud classification. In *the Proceedings of the 4th International Symposium on 3D Data Processing, Visualization and Transmission*, Georgia Institute of Technology, Atlanta, GA, June 2008. 2

[12] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1–2):100–115, 1954. 4

[13] H. V. Poor and O. Hadjiliadis. *Quickest Detection*. Cambridge University Press, 2008. 1, 4

[14] D. Siegmund. *Sequential Analysis*. Springer-Verlag, New York, 1985. 5

[15] I. Stamos, L. Liu, C. Chao, G. Wolberg, G. Yu, and S. Zokai. Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. *International Journal of Computer Vision*, 78(2–3):237–260, July 2008. 1

[16] S. Thrun, M. Montemerlo, and *et al*. Stanley: the robot that won the DARPA grand challenge. *Journal of Field Robotics*, 23(9):661–692, 2006. 2

[17] A. Wald. *Sequential Analysis*. Wiley, New York, 1947. 1, 4

[18] M. A. Wami and B. G. Batchelor. Edge-region-based segmentation of range images. *IEEE Transactions on PAMI*, 16(3):314–319, March 1994. 2

[19] D. Wolf, G. Sukhatme, D. Fox, and W. Burgard. Autonomous terrain mapping and classification using hidden markov models. In *the Proceedings of the International Conference on Robotics and Automation*, pages 2038–2043, Barcelona, Spain, April 2005. 2

[20] Y. Yu, A. Ferencz, and J. Malik. Extracting objects from range and radiance images. *IEEE Transactions on Visualization and Computer Graphics*, Oct-Dec 2001. 2

[21] H. Zhao and R. Shibasaki. Reconstructing a textured CAD model of an urban environment using vehicle-borne laser range scanners and line cameras. *Machine Vision and Applications*, 14(1):35–41, 2003. 1