Object Detection via Time-Of-Flight Technology

Minjie Chen Günter Bärwolff Hartmut Schwandt

***Abstract*—In the current work we propose a new method based on the modulation-based Time-Of-Flight (TOF) ranging technology for the automatic trajectory extraction of moving objects. We introduce a new data structure called TOF-tree for the segmentation of data in a sequence of frames. A key feature of this data structure is the monotonicity in height values acquired from distance measurement. This tree-like data structure enables an efficient calculation of the segmented objects (which we call TOF-objects) on arbitrary height levels. We apply a simple matching on the segmented TOF-objects to reconstruct the original objects and compute, accordingly, the position evolution of the detected objects in the scene. Automatic trajectory extraction of this kind can be easily adapted for industrial use, for example, automatic passenger counting in public transportation.**

***Index Terms*—Time-Of-Flight, object detection, trajectory extraction**

I. Introduction

In the current text we propose a new method based on the Time-Of-Flight (TOF) measuring technology for the trajectory extraction of moving objects.

Many traditional laser-based methods of distance measurement require precise measurement of light travel duration. For a short distance, this task is very hard to perform. The so-called pulse-based TOF technology measures the direct light travel duration on a scale of picoseconds by means of single-photon avalanche diode (SPAD) detectors [**? ?**]. The other category of TOF is modulation-based [**? ? ?**]. Instead of the direct light travel duration, in modulation-based TOF the phase shift $\Delta\phi$ between an amplitude-modulated light wave and its reflection from the target object will be measured. Within a certain range, this phase shift $\Delta\phi$ of an amplitude-modulated light wave with frequency f is proportional to the actual flight duration, and can thus be used to determine the distance to measure (with c denoting the speed of light):

$$\Delta\phi = \frac{4\pi f}{c} \cdot d. \tag{1}$$

Obviously, the more challenging task of measuring very short flight duration is no longer necessary. The maximum distance range to guarantee the unambiguity of (1) is $d_{\max} = \frac{c}{2f}$. The calculation of $\Delta\phi$ via sampled intensity values of the light signal is explained in [**?**]. For a detailed error analysis of the measurement [**?**] can be consulted.

II. System Construction

In our system, a TOF sensor will be installed perpendicular to the ground of the scene. The recording of the scene through the sensor provides us with ranging data stored in a sequence of n frames $(F_0, \ldots, F_{n-1})$, and in each of these frames, a cloud of sample points. The positions of these points, as in the so-called sensor coordinate system, can be easily converted into the local coordinate system of the scene. For simplicity's sake, we choose the x-y-plane $(z = 0)$ of the local coordinate system to be the floor of the scene. The z-axis is set to be the reverse of the optical axis of the sensor. The radial distances of the sample points to the sensor will be projected onto the z-axis to retrieve the height information of these points. Depending on the hardware resolution, for every sample point, its (x, y)-position in the local system can be rescaled into the index of a two-dimensional array representing the original region of observation Ω, see Fig. 1. The algorithmic description of our method is therefore independent of hardware specifications.

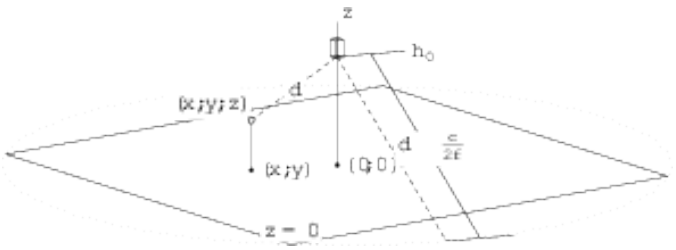

Fig. 1. Installation of the TOF sensor at a height of h_0 in the local coordinate system. The radial distance d to the object will be converted into the z-component of the coordinates.

Our previous work [**?**] proposed a quadtree-like data structure called "TOF-node". A TOF-node stores the z-value associated with a sampled position (x, y) and holds pointers to up to four different child TOF-nodes (written as W, N, E and S). A child node is required to have a smaller z-value (height), thus a TOF-node exhibits a local maximum in height among all its descendant nodes. A simplified representation is given in Fig. 2. The child nodes, written as W (west,

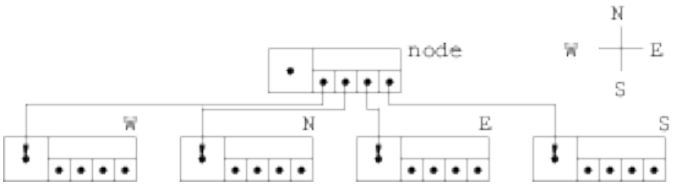

Fig. 2. Schematic representation of the TOF-node. The TOF-node "node" may or may not be de-referenced by a parent node. The compass shows the relative positions respecting the four possible child nodes.

with $(\Delta x, \Delta y) = (-1, 0)$), N (north, with $(\Delta x, \Delta y) = (0, 1)$), E (east, with $(\Delta x, \Delta y) = (1, 0)$) and S (south, with $(\Delta x, \Delta y) = (0, -1)$), refer to the immediate neighbouring positions where the z-values are lower. A TOF-node, which is itself not de-referenced by any other TOF-node, will be referred to as a "TOF-tree". Since the monotonicity respecting the z-values is imposed, the root node of a TOF-tree is always associated with a local maximum of height in the scene. Consequently, a convex object can be represented by a TOF-tree. We notice that the original geometric information is completely preserved in this data structure.

III. Algorithm

A. Construction of TOF-trees

We review the process of constructing all the TOF-trees in a frame:

procedure BUILD**:**
parameter: global information

repeat
find point p with maximum height;
create TOF-node node with p;

Manuscript created March 31, 2014; revised April 25, 2014.

The authors are with the Institut für Mathematik, Technische Universität Berlin, Germany. Emails: {minjie.chen, baerwolf, schwandt}@math.tu-berlin.de

call BUILD_TREE with node;
until all points processed

return

This procedure searches for the local maxima in the global scene and constructs a TOF-tree with each of these:

procedure BUILD_TREE**:**
parameter: node

allocate memory for W, N, E and S;
mark node as processed;
call BUILD_TREE_W with relevant candidate position and W;
call BUILD_TREE_N with relevant candidate position and N;
call BUILD_TREE_E with relevant candidate position and E;
call BUILD_TREE_S with relevant candidate position and S;

return

The construction of a single TOF-tree is a recursive procedure. The expansion of this TOF-tree may follow in all four directions of W, N, E and S. For example, in the direction of W (that is, with a relative position $(-1, 0)$) the recursion looks like:

procedure BUILD_TREE_W**:**
parameter: candidate position, node

if candidate position not defined in Ω
save state "boundary_west"; // in other directions $(*)$
// "boundary_north", "boundary_east" and
// "boundary_south" respectively
elseif candidate position not already processed
and associated height value not increasing
set W as TOF-node pointer to candidate;
call BUILD_TREE with node;
fi

return

The other three components BUILD_TREE_N, BUILD_TREE_E and BUILD_TREE_S are analogous.

Since BUILD processes all the sample points in the scene, the result of this procedure is a segmentation of the sample points in the observation area Ω. Sample points with extra noises (due to measurement inaccuracy or error etc.) would lead to isolated TOF-trees. These TOF-trees have generally very small sizes and can be easily filtered out. Although this results later in holes of the reconstructed objects, the side effects of this phenomenon (for example, in retrieving geometric information of the objects) are in most cases neglectable.

B. Processing of TOF-trees

Once we have the segmentation in the frames, the TOF-trees can be processed. The basic geometric information about the TOF-trees can be retrieved in a similar way as in the above recursive procedure BUILD_TREE. A special task of the processing of a TOF-tree is to retrieve its boundary information like $(*)$ in BUILD_TREE_W on any arbitrary height level, but this time it includes the information about the neighbouring TOF-trees. In other words, in every frame F_i $(i = 0, \ldots, n-1)$, we have a segmentation of the sample points, composed of a collection of TOF-trees $t_{i,0}, \ldots, t_{i,f_i-1}$ (with f_j denoting the number of the TOF-trees constructed in the frame F_j, $j = 0, \ldots, n-1$).

We observe the fact that a geometrically convex object can be represented by a unique TOF-tree. Consequently, the objects detected in the scene can, in general, be represented by sets of connected TOF-trees. Let $T_i = \{0, \ldots, f_i - 1\}$ be the set of the indices of the TOF-trees. Let further 2^{T_i} denote the power set of T_i. For every nonempty element B of 2^{T_i} ($B \subseteq T_i$, $B \neq \emptyset$), we examine the boundary information of the TOF-trees associated with this index set B: when these are connected,[1] we will construct a so-called "TOF-object" with these indices.[2] A TOF-object can be thus considered as a union of the TOF-trees associated with the indices from the set B; it serves as a candidate for a real object to be detected in the scene. We notice the almost exponential complexity of this step through introducing the power set 2^{T_i}, so the pre-processing (smoothing etc.) of the original sampled data and the filtering of the irrelevant TOF-trees are very necessary. Empirically speaking, however, above a certain scale, the growth of the complexity in this step is evidently much lower than being exponential.[3] The geometric information of the TOF-objects can be easily computed once we know the corresponding TOF-trees.

C. Matching of TOF-objects

The last two steps are performed within the individual frames. We now consider another data structure which we call "TOF-trajectory". A TOF-trajectory is a record of TOF-objects in a continuous sequence of frames. Given such a record, the physical trajectory (or position evolution) of the corresponding real object detected in the scene can be reconstructed.

In the start frame F_0, we may initialize a series of incomplete TOF-trajectories $l_{0,0}, \ldots, l_{0,f_0-1}$ by the present TOF-trees in F_0. In the next frame F_1, these trajectories will be marked as active, if not otherwise closed (completed).

In a following frame F_i $(i = 1, \ldots, n-1)$, a TOF-trajectory $l_{i-1,k}$ will be marked as active, if $l_{i-1,k}$ contains a TOF-object (addressed by k) from the frame F_{i-1}. In other words, $l_{i-1,k}$ is active, if it has not been closed in the previous frame F_{i-1} already. Assume, in frame F_i, we have constructed a collection of TOF-objects $O_i = \{o_{i,0}, \ldots, o_{i,|O_i|-1}\}$ (with obviously $|O_i| \leq 2^{|T_i|} - 1$); at the same time, we have a collection of incomplete TOF-trajectories $L_{i-1} = \{l_{i-1,0}, \ldots, l_{i-1,|L_{i-1}|-1}\}$ derived from the previous frame F_{i-1}. It is then possible to construct a distance matrix D_i with $|O_i| \cdot |L_{i-1}|$ items:

$$D_i = \left(d_{i,u,v}\right), \text{for } u = 0, \ldots, |O_i| - 1, v = 0, \ldots, |L_{i-1}| - 1, \quad (2)$$

where $d_{i,u,v}$ is the distance from the TOF-object $o_{i,u} \in O_i$ to the active TOF-trajectory $l_{i-1,v} \in L_{i-1}$. The distance function to compute $d_{i,u,v}$ in (2) can be defined independently.

A very straightforward suggestion for $d_{i,u,v}$ can be:

$$d_{i,u,v} = |o_{i,u} - o_{i-1,v}|_2, \quad (3)$$

where $o_{i-1,v}$ denotes the TOF-object stored in the TOF-trajectory $l_{i-1,v}$ in the previous frame F_{i-1} (in other words, $o_{i-1,v}$ is the last record in the active TOF-trajectory $l_{i-1,v}$ in frame F_i). The geometric centre of a TOF-object $o_{i,u}$, which is again the weighted average of

[1] Our formal definition for "being connected": either B is composed of exactly one element; or in case B is composed of two elements, there exists a common boundary shared by the TOF-trees associated with these two indices; or otherwise for every pair (p, q) with $p, q \in B$, $p \neq q$, there exists a sequence of elements $\alpha, \ldots, \omega$ from the rest of B, that is, $\{\alpha, \ldots, \omega\} \subseteq B \setminus \{p, q\}$, so that all the pairs $(p, \alpha), \ldots, (\omega, q)$ are connected as in the case of a subset B with two elements.

[2] A note on the plural form: obviously this does not exclude that B is composed of exactly one element in which case this index alone will be used to construct a TOF-object.

[3] Although the examination of all the elements B in 2^{T_i} remains of a task of exponential complexity, the overall cost of this step can be considered roughly as linear with large constants, since most elements of 2^{T_i}—which are indices of the TOF-trees—are not qualified for the construction of TOF-objects (with the corresponding TOF-trees being not connected).

the centres of the associated TOF-trees generated by the index set B, can be used to describe its position:

$$\frac{\sum_{b\in B} t_{i,b}}{\sum_{b\in B} |t_{i,b}|}\,.$$

The next step is to match the current TOF-objects with the active TOF-trajectories and update the TOF-trajectories for the next frame F_{i+1}:

procedure MATCH**:**
parameter: O_i, L_{i-1}

mark incomplete TOF-trajectories from L_{i-1} as active;
compute D_i and set all items in D_i as active;

repeat
select minimum $d_{i,u,v}$ from all active items in D_i;
if $d_{i,u,v}$ below threshold
add $o_{i,u}$ to $l_{i-1,v}$;
deactivate $l_{i-1,v}$;
else
construct an incomplete TOF-trajectory by $o_{i,u}$
for the next frame F_{i+1};
fi
deactivate all items $d_{i,u',v'}$ in D_i
for all $u' = 0, \ldots, |O_i| - 1$, $u' \neq u$ that $o_{i,u} \cap o_{i,u'} \neq \emptyset$
and $v' = 0, \ldots, |L_{i-1}| - 1$; (∗∗)
until no active item in D_i left

if there are active TOF-trajectories left
// but there should be no TOF-object
mark all active TOF-trajectories as closed;
else
// there are active TOF-objects left
// but there should be no TOF-trajectory
repeat
construct an incomplete TOF-trajectory by an active
TOF-object for the next frame F_{i+1};
deactivate all other overlapping active TOF-objects;
until no active TOF-objects left
fi

return

Procedure MATCH can be decomposed into three parts. The first part is the initialization and computation of a distance matrix D_i concerning all the current TOF-objects and TOF-trajectories in the frame. In the second part, the procedure searches for the minimum in the distance matrix after every update of the TOF-objects and TOF-trajectories (the distance matrix D_i itself does not need to be re-calculated). Given the successful search result of a TOF-object and a TOF-trajectory, we either, if the current minimum distance is below an empirical threshold value, append the TOF-object to the TOF-trajectory, or we start a new TOF-trajectory with the TOF-object. The empirical threshold value applied here needs to be calibrated with consideration of the external distance function (3) with which the distance matrix D_i is computed and the actual frame frequency of the sensor data. Line (∗∗) says that once a TOF-object has been used to construct or update a TOF-trajectory, all the other TOF-objects with which there is an overlapping of the TOF-trees must be removed from the current frame F_i. The third part of this procedure deals with the rest TOF-objects or TOF-trajectories, but not both (otherwise if there were both TOF-objects and TOF-trajectories left, there would be also active $d_{i,u,v}$ items present in the distance matrix D_i). The TOF-trajectories we have here can be considered as complete, since no TOF-object will be attached to them, they will not be activated in the next frame. Similarly, the rest TOF-objects will be used to construct new TOF-trajectories, since they are not matched with any active TOF-trajectory in the current frame. Naturally, we need to run the procedure MATCH through the frames $F_1, \ldots, F_{n-1}$.

Frames from an example simulation are given in Fig. 3.

IV. Possible Extension

The above approach can be regarded as a very simple contour-based object tracking method. The background idea of our method is to register objects in the scene with their possible contour lines and then apply a simple matching of position transition on the registered contours. The generation of contour lines is very different from other traditional methods, owing to the speciality of the TOF data delivered by the ranging system. Since ours is a very rudimentary approach in the processing of TOF data, we have only applied a very simple matching method to establish the ownership relationship of an object's actual position to its possible trace over the time (that is, trajectory). The distance function in the procedure MATCH is designed deliberately to be an external function for future development. Numerous established matching methods can be embedded here as function modules. The distance function in (3) can be extended to include a prediction of position under the assumption of a constant velocity:

$$d_{i,u,v} = |o_{i,u} - o'_{i,v}|_2, \tag{4}$$

with $o'_{i,v}$ denoting the position prediction of the TOF-object stored in the TOF-trajectory $l_{i-1,v}$ in the current frame F_i. With a constant frame frequency (that is, the interval lengths of the recording are constant), we have:

$$o'_{i,v} = 2o_{i-1,v} - o_{i-2,v}.$$

With (4), procedure MATCH requests the information of another frame F_{i-2} for the processing of the current frame F_i in the data sequence.

V. Further Discussion

In the current paper we introduced a new method of moving object detection and automatic trajectory extraction via the Time-Of-Flight technology. Since modulation-based TOF has a range limit of $d_{\max} = \frac{c}{2f}$ (with $f = 20\text{MHz}$ the maximum range would be 7.5m approximately), this method is applicable primarily in smaller observation regions. The method can be applied in various contexts, for example in the automatic passenger counting in public transportation. Our method itself is hardware-independent. The specific TOF hardware we used is a SwissRanger4000 sensor[4] with a resolution of 176×144. In practice, even a much lower resolution would be feasible, if the range information is collected with a sufficient precision (so that the geometric information, especially shape, of the objects can be reconstructed). However, the quality of data collection depends on further factors like light conditions which are generally discussed in the context of optical ranging systems.

VI. Acknowledgement

The authors gratefully acknowledge the support of Federal Ministry for Economic Affairs and Energy of Germany for the project VP2653402RR1 and federal state of Berlin/Investitionsbank Berlin for the project 10153525.

[4]Produced by Mesa Imaging AG (Switzerland), see homepage http://www.mesa-imaging.ch.

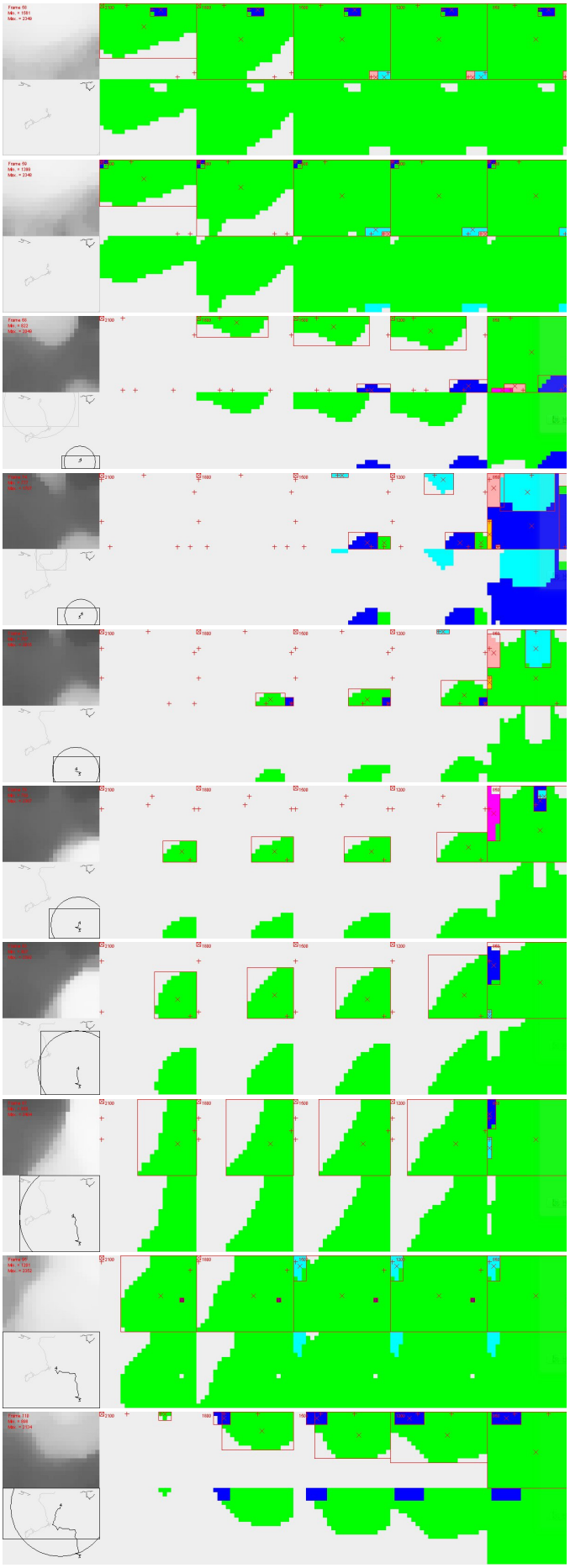

Fig. 3. Frames from a simulation example of two passengers walking through a door. The opening of the door was captured in two small trajectories in the corners. Trajectories of the passengers were shown with bounding-boxes. The segmentation of the sample points were shown on the right side of the figures.