2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Edge-based Visual Odometry with Stereo Cameras using Multiple Oriented Quadtrees

Changhyeon Kim, Junha Kim and H. Jin Kim

Abstract— **We propose an efficient edge-based stereo visual odometry (VO) using multiple quadtrees created according to image gradient orientations. To characterize edges, we classify them into eight orientation groups according to their image gradient directions. Using the edge groups, we construct eight quadtrees and set overlapping areas belonging to adjacent quadtrees for robust and efficient matching. For further acceleration, previously visited tree nodes are stored and reused at the next iteration to warm-start. We propose an edge culling method to extract prominent edgelets and prune redundant edges. The camera motion is estimated by minimizing point-to-edge distances within a re-weighted iterative closest points (ICP) framework, and simultaneously, 3-D structures are recovered by static and temporal stereo settings. To analyze the effects of the proposed methods, we conduct extensive simulations with various settings. Quantitative results on public datasets confirm that our approach has competitive performance with state-of-the-art stereo methods. In addition, we demonstrate the practical values of our system in author-collected modern building scenes with curved edges only.**

I. Introduction

Recent advances in the accuracy and real-time performance of visual odometry (VO) have been striking thanks to standard pipelines such as sparse point-based approaches [1]–[3], direct methods which find an optimal camera motion minimizing intensity residual between two images [4]–[6], and iterative closest points (ICP)-based algorithms [7]–[10] that align a pair of large point sets, e.g. 3-D point clouds, to obtain relative camera motions.

Despite such maturity, robustness to feature-less scenes and fluctuating illuminations is not yet sufficient. To alleviate this, irregular brightness changes have been considered as an affine model and compensated for semi-dense regions [11], and straight line features are invited to maintain VO to keep track of motions even in low-textured scenes [12], [13].

As another attempt to improve robustness, edge-based VO systems have been introduced recently [14], [15]. The image edges can be stably detected by a traditional method [16] even in monotonic surfaces often encountered in the man-made world. Moreover, a continuum of edge pixels gives more 3-D structural information of surroundings than sparse points, which fits interactive applications better.

For utilizing image edges for VO, as reported in [17], several hurdles still remain. Especially, 1) there are no apparent and efficient matching criteria for edges contrary to points and straight lines [18], [19], and 2) too many edge pixels cause computational burden too heavy for real-time employment. In this paper, our main objective is to deal with these two issues and efficiently incorporate free-formed edges into a robust stereo visual odometry system.

All authors are with Department of Mechanical and Aerospace Engineering, Seoul National University and Automation and Systems Research Institute (ASRI), Seoul, South Korea. {rlackd93,wnsgk02,hjinkim}@snu.ac.kr

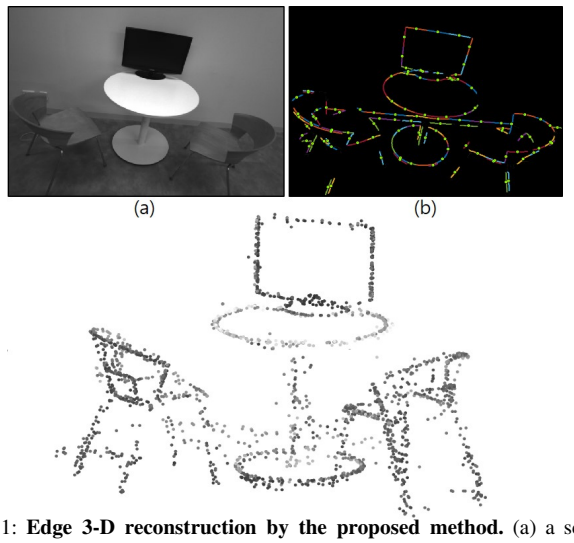

Fig. 1: **Edge 3-D reconstruction by the proposed method.** (a) a scene with monotonous objects in the modern office, (b) a result of the proposed edge culling method. Our method can stably operate on challenging scenes with few points and straight lines by only using free-formed edges. Thanks to the static and temporal stereo, the edge depth map can be fully recovered.

A. Related works

Points & direct intensity: Early VO approaches have been mostly developed using point features, and point-based methods have shown high localization accuracy and robustness to large motions between frames with real-time operations [1]–[3]. Recently, VO methods utilizing intact brightness values for localization, so called direct methods, are actively researched [4], [5]. Compared to the former, the latter is less susceptible to motion blurs and provides denser representation, which is more attractive in practical aspects. Despite the successful research history, both methods still have limitations in real-world situations: point-based methods heavily rely on point features hardly existing in modern man-made scenes, and direct methods can be influenced by varying illuminations.

Lines: Straight lines are intermediate features between points and free curves, observed even in low-textured scenes. For enhancing robustness against those scenes, a stereo VO aligning multiple lines is proposed in [8], and [12] utilizes points with lines based on a monocular semi-direct approach [2]. In [21], a robust rgb-d direct VO combining points and lines is suggested. In those works, lines are not used as major features, but for additional constraints for point-based

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

VO, because both end points of a line are not consistently extracted even by using state-of-the-art line descriptor [19].

Edges: Edges are generalized features including points, lines, and arbitrary curves, and easily observed in most scenes. Although some attempts to adopt edges into VO began in the early days [22], full-fledged edge-based VO systems have emerged only recently.

Edge-based VO methods can be largely divided into two types. The first one regards edges only as assistive profiles for photometric error minimization, not as major features. The work in [14] estimates rgb-d camera motions by minimizing both photometric and geometric errors of distance transform maps. Similarly, [10] suggests a rapid rgb-d VO by minimizing photometric errors around sparsely-sampled edge pixels. [23] develops a sub-gradient image aligning method using a distance transform around edges. They can enhance robustness to low-textured scenes by imposing additional constraints by using edges. Nonetheless, they are still vulnerable to lighting changes due to the dependency on photometric properties.

The other type of edge-based VO methods utilizes intact edge pixels as the main features. In these methods, explicitly matching edge pixels is one of the most crucial parts. To mitigate difficulties in matching caused by the absence of proper descriptors for edges, several approaches are proposed in [9], [15], [17], [24], [25].

We can further categorize the matching methods of the latter type into two approaches: 1) *searching from geometry* and *2) searching from data structures*. Geometric approaches confine search regions by utilizing edge normal directions. In [24], searching is conducted along the normal direction of edge curves. [25] suggests an rgb-d VO using approximated neighbor fields (ANNFs) for fast edge matching, and the matching completeness is further improved via oriented edge neighbor fields (ONNFs) in [17]. The other approaches use data structures to find the most likely pair. This idea is originally from ICP algorithm [9] using a k-d tree structure. In [15], the robustness is improved by adopting a k-d tree considering image gradient vectors to compare edge similarities in cluttered regions. Note that these edge VO methods still exploit photometric information and most methods rely on rgb-d sensors to obtain the 3-D information of edge regions.

Our main goal is to further improve an edge-based VO with an explicit match process by proposing a new dedicated data structure and efficient edge pre-processing steps.

B. Contributions

In this paper, we give in-depth consideration on solving two problems when realizing an efficient edge-based stereo VO: high ambiguities on edge matching and redundancy of edge pixels. The key contributions can be summarized as follows:

- We propose an ICP-based efficient stereo visual odometry system using the dedicated multiple quadtrees structure and the edge culling method.

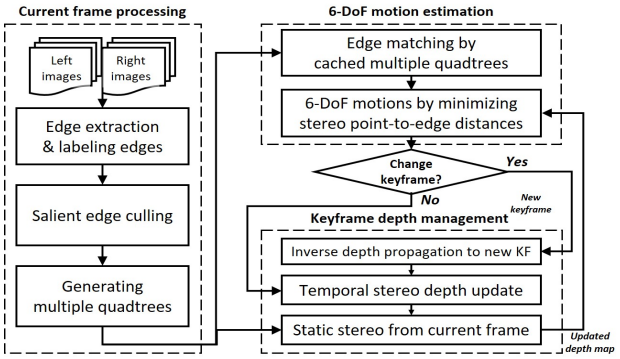

Fig. 2: **The flowchart of the proposed system.**

- By using the proposed edge culling method, many cluttered edge responses are suppressed, consequently, the required computational load is reduced while maintaining VO performance.
- Edge matching speed and success rates are improved by the proposed multiple quadtrees and node caching schemes.
- We demonstrate that our method has robust and competitive performance with state-of-the-art stereo VO on publicly available datasets and author-collected scenes.

C. Overview

A flowchart of our system is illustrated in Fig. 2. For every stereo stream, all the edge pixels are classified into eight orientation bins with mutually inclusive regions according to their image gradient directions. In Section. III, to reduce redundant edge pixels, we additionally condense several thousands of raw edge pixels into well distributed structural edgelets by the proposed edge culling method. We propose an efficient multiple quadtrees structure composed of eight orientation bins, and store the previously matched nodes for warm-starting in the next iteration, which are detailed in Section IV. Section V details the ICP-based camera motion estimation by minimizing stereo point-to-edge normal distances between current images and key frame images, and the static and temporal stereo method for updating edge inverse depths. The extensive analysis on each core part and experimental results are following in Sections VI and VII.

II. Preliminaries

A. Notation

We use bold letters for column vectors and matrices, and right superscripts c and k to denote variables represented in the current frame and key frame, respectively. The secondary right superscripts l and r express the left and right frames of stereo cameras. For example, we denote the i-th pixel coordinate on a left key image as $\mathbf{p}_i^{k,l} \in \mathbb{R}^2$ with its inverse depth $\rho_i^{k,l} \in \mathbb{R}^+$ as suggested in [20]. The perspective relationship between $\mathbf{p}$ having the inverse depth ρ and corresponding 3D point $\mathbf{x} \in \mathbb{R}^3$ can be represented as $\mathbf{p} = \pi(\mathbf{x}) : \mathbb{R}^3 \mapsto \mathbb{R}^2$, and its inverse mapping is $\mathbf{x} = \pi^{-1}(\mathbf{p}, \rho)$. We define an image gradient vector of $\mathbf{p}$ on the left key image as $\mathbf{g}^{k,l}(\mathbf{p}) : \mathbb{R}^2 \mapsto \mathbb{R}^2$. For simplicity, all image gradient vectors are assumed to be normalized.

B. 3-D geometry of camera motions

The 6-DoF camera motion from a current frame to a key frame is parametrized by Lie algebra $\xi^l_{c,k} \in se(3)$ where corresponding rotational matrix on special orthogonal group $\mathbf{R}^l_{c,k} \in SO(3)$ and translation $\mathbf{t}^l_{c,k} \in \mathbb{R}^3$. The 3-D warping function transferring $\mathbf{p}^{k,l}$ to a corresponding pixel point $\mathbf{p}^{c,l}$ on the current frame is defined as,

$$\mathbf{p}^{c,l} = w\left(\mathbf{p}^{k,l}, \xi^l_{c,k}\right) = \pi\left(\mathbf{R}^l_{c,k} \cdot \pi^{-1}\left(\mathbf{p}^{k,l}, \rho^{k,l}\right) + \mathbf{t}^l_{c,k}\right). \tag{1}$$

A right camera motion $\xi^r_{c,k}$ can be denoted with an operator $\oplus$ on $se(3)$ and a fixed stereo pose $\xi^l_{l,r} \in se(3)$,

$$\xi^r_{c,k} = \xi^l_{c,k} \oplus \xi^l_{l,r}. \tag{2}$$

C. ICP-based edge alignment

Our method estimates camera motions by successively aligning matched pairs of edges within the ICP framework. To get it working, the most probable pixel correspondences among current and key edges should be established in advance. This matching process is called the nearest neighbor searching (NNS) [9]. For a query $\mathbf{q}$, the nearest pair in a pixel set $\mathcal{R} \subset \mathbb{R}^2$ can be founded by a NNS function $nn_{\mathcal{R}}(\mathbf{q})$,

$$nn_{\mathcal{R}}(\mathbf{q}) = \underset{\mathbf{p} \in \mathcal{R}}{\arg\min} \|\mathbf{q} - \mathbf{p}\|_2 \in \mathcal{R}, \tag{3}$$

where $\|\cdot\|_2$ is a 2-norm operator.

The well-distributed edges, however, are not always guaranteed, and many false edge responses hinder the correct matching. To characterize and find more informative edges, we develop an edge culling method by making use of the fact that structural edges along object boundaries commonly possess long series of pixels with regular and high image gradients. The detailed explanation follows in the next section.

Using the correspondences, an ICP algorithm estimates an optimal camera motion $\xi^*_{c,k}$ by minimizing the sum of squared distances of the residual vector $\mathbf{d} \in \mathbb{R}^{N_p}$,

$$\xi^*_{c,k} = \underset{\xi \in se(3)}{\arg\min}\, \mathbf{d}^{\mathrm{T}}\mathbf{d}, \tag{4}$$

where N_p is the number of matched pixel pairs. The n-th element of residual vector, d_n, formulated between the n-th pixel $\mathbf{p}^k_n$ on the key image and its matched point in the pixel set $\mathcal{R}^c$ of the current image can be noted as 2-norm distance,

$$d_n = \left\| w\left(\mathbf{p}^k_n, \xi_{c,k}\right) - nn_{\mathcal{R}^c}\left(w\left(\mathbf{p}^k_n, \xi_{c,k}\right)\right)\right\|_2. \tag{5}$$

As our system uses both stereo images to track camera motions, we design a new stereo cost function and additional methods to restrain outliers, which are also discussed in the following sections.

III. Edge Extraction and Culling

In this section, we detail how we distinguish edge pixels and extract salient structural edgelets out of raw pixels.

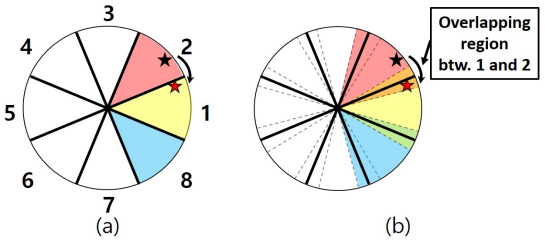

Fig. 3: **Edge label bins with overlapping regions.** (a) the original binning method proposed in [17] divides eight exclusive directional sets where pixels adjacent to boundaries could be wrongly matched, (b) the proposed method with overlapping regions gives flexibility to some extent for boundary pixels.

A. Edge labeling using overlapping regions

When using edges, the major difficulty originates from the absence of dedicated descriptors for edges. To relieve this problem, [15] and [17] exploit image gradient directions to distinguish edges in different ways.

In [15], a similarity score between two pixels is quantified by an weighted sum of a pixel Euclidean distance and an inner product of normalized image gradients. Despite improved matching success rates, this method relies on a heuristically-defined weighting parameter between two terms, making it improper to be used in universal situations.

The second work [17] divides edges into eight bins according to image gradient directions, and *absolutely* labels each edge as one of eight mutually exclusive groups like Fig. 3(a). In this way, the search space can be effectively reduced and at the same time, matching success rates be increased.

We follow the main concept of the *absolute* labeling method, but additionally augment overlapping regions between every neighboring bins where pixels can belong as duplicates. As can be seen in Fig. 3(a), only with small rotations on image, the black point labeled as group 2 easily crosses the decision boundary between groups 1 and 2, and both black and red points become mutually exclusive. In this case, the original approach could fail to find the correct pair.

In contrast, by setting up the proposed overlapping regions, the red one is now be labeled as a duplicate of groups 1 and 2, and can included into searching candidates of the black one regardless of some extent of rotations as depicted in Fig. 3(b). We utilize this labeling result to extract salient edgelets and make multiple quadtrees in the following sections.

B. Finding salient edgelets out of labeled edges

A high signal-to-noise ratio from a number of pixels can be helpful for more accurate motion estimations [17]. However, many points could be redundant to get sufficient estimation performance, and spurious edges could hinder finding correct pairs rather than improve overall performance.

To address both problems, we propose an efficient edge culling method that filters false responses on cluttered edges and only sorts out prominent structural edgelets. As shown in Fig. 4(a), edges along object rims generally have regular image gradients, and edges in cluttered regions show many unconnected pixels. Given these observations, we consider a long series of connected edge pixels with the same labeling group as structural edgelets, and if not, as cluttered edges.

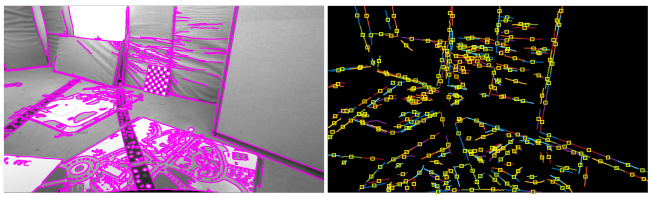

Fig. 4: **Extracted salient edgelets and center points.** (a) raw edges (24,672 pixels), (b) salient edgelets extracted by the proposed method in different colors. The yellow squares represent center points of edgelets. Total 574 center points are well distributed across the image.

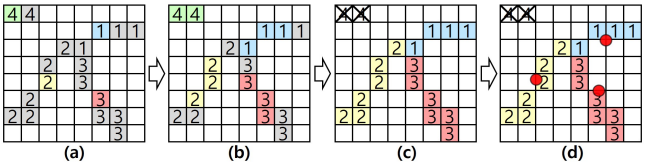

Fig. 5: **Example of depth-first search for finding salient edgelets.** Each color represents a different edgelet. Red points denote centers of respective edgelets.

The structural edgelets can be determined by recursively connecting adjacent pixels with the same directions. A simple example is described in Fig. 5. At the first recursion, we initiate at one of the color-coded starting query points. For the current query point, 3×3 adjacent pixels are stored into a list L if the pixels have the same labels with the query point. If there is no more pixel to be connected, the list L is regarded as a new structural edgelet. The algorithm restarts at non-visited new query points, and ceases when all the edge points in image are visited.

After grouping, we check the length of each edgelet by double threshold values, l_{min} and l_{max}. As the step (c) in Fig. 5, fragmented edgelets under the minimum length l_{min} are more likely to be spurious responses on cluttered regions. Thus, we reject those edgelets.

For more compact query sets, we additionally compute centers of the edgelets and consider them as representative query points. We found that several long edgelets can yield very sparse center points. To prevent this, we set the maximum length limit l_{max} to get more uniform length edgelets. We finally obtain the prominent edgelets along the object profiles and the query points evenly distributed over the entire image area as small yellow squares shown as Fig. 4(b). In our experience, we found that the practical value for l_{max} is about 30 in general images like Fig. 4.

All procedures can be efficiently performed by adapting the conventional depth-first search algorithm, and a pseudo code of the method is written in Algorithm 1.

IV. Edge Matching via Oriented Quadtrees

For estimating the camera motion between key and current frames, one of the most important and exhaustive parts is to repetitively match point pairs. To realize faster and more accurate ICP-based edge alignments, it is crucial to efficiently find the correct correspondences among the massive number of edge pixels. To this end, we propose an accelerated NNS strategy using oriented multiple quadtrees dedicated to our edge-based VO.

Algorithm 1 Find Salient Edgelets and Centers

E; an image of labeled edge pixels
$edgelets$; a list of detected edgelets
pts_c; center pixels of selected edgelets
for All pixel q in E **do**
  $L \leftarrow$ an empty list for a new edgelet;
  if $\mathbf{E}(q)$ is an edge **then**
    Create an empty $stack$ and push q to $stack$;
    while $!isEmpty(stack)$ **do**
      $p \leftarrow frontAndPop(stack)$;
      for All p_n neighbor of p **do**
        if $size(L) < l_{max} \& E(p_n) = E(q)$ **then**
          Add p_n to $stack$ and $edgelet$;
        end if
      end for
    end while
    if $size(L) > l_{min}$ **then**
      Add L to $edgelets$, and $mean(L)$ to pts_c;
    end if
  end if
end for

A. Generating multiple oriented quadtrees

To begin with, we regard extracted edgelets of a current image as reference points for making quadtrees, and calculated center points of a key image as query points to be matched. Based on this assumption, multiple quadtrees are built in accordance with eight orientation labels by using a set of salient edgelets on the current image $\mathcal{R}^c \subset \mathbb{R}^2$. We denote each set of edge points consisting of its relevant tree as $\mathcal{R}_i^c \subset \mathcal{R}^c$ where an indicator i denotes each directional bin. The difference between a normal quadtree and the proposed trees is illustrated in Fig. 6.

As depicted before in Fig. 3(b), the pixels near boundaries among neighboring bins are doubly inserted into two neighboring trees. Thus, both boundary query and its correct match can remain reachable to each other regardless of a certain degree of image rotation, which makes the matching process more robust to rotational motions.

B. Fast NNS strategy by storing visited nodes

At every iteration of the ICP-based motion estimation, we have to warp key points onto the current image and find their correspondences within the current quadtrees.

For a simple explanation, we assume a situation to find a matching pair for a single key point $\mathbf{p}^k$ with an orientation labeling i. Given the motion $\xi_{c,k}$, let $\mathbf{p}^{k\prime} = w\left(\mathbf{p}^k, \xi_{c,k}\right)$ be a warped point of $\mathbf{p}^k$. Thanks to the orientation labeling, we can directly narrow down potential candidates within $\mathcal{R}_i^c$, and the nearest pixel can be determined by the NNS function $nn_{\mathcal{R}_i^c}\left(\mathbf{p}^{k\prime}\right)$.

Due to gradual motion updates of the ICP-based approach, warped points move only few pixels at each iteration. It implies that the previously matched node is much likely to be re-matched at the very next iteration.

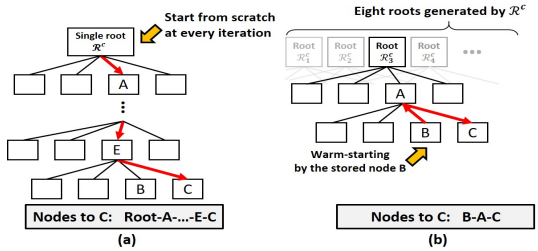

Fig. 6: **Comparison between a normal single-rooted quadtree and the multiple cached quadtrees.** (a) the original quadtree travels through all the nodes from a root to get C. (b) the proposed quadtrees cache the previously visited nodes and warmly start from the cached nodes, which considerably reduces nodes to be traveled.

Inspired by this observation, the matching process can be further accelerated by caching the address of the matched nodes at the previous iteration, and warm-starting the search from the cached nodes. If the best match does not reside in the cached node, we restart to search from the root node.

When seeking a true match in the node C depicted in Fig. 6(a), for the normal case, exhaustive re-entry to the root is required at every iteration, and many nodes are visited on the way from the root to the node C. In contrast, the proposed trees in Fig. 6(b) can compactly tighten the search space by the orientation label, and moreover, the warm-start from the previously cached node B reduces a large number of visited nodes to reach the node C compared to the normal case.

V. Motion Tracking and 3-D Reconstruction

A. Stereo Point-to-Edge Distances Minimization

We estimate the left camera motion $\xi^l_{c,k}$ by minimizing the stereo point-to-edge normal distances generated by matched edge pairs of the current and key frames. In Fig. 7, we assume that the n-th query pixel $\mathbf{p}^{k,l}_n$ has an orientation label i, and its warped point by the camera motion $\xi^l_{c,k}$ is denoted by $\mathbf{p}^{k,l\prime}_n = w\left(\mathbf{p}^{k,l}_n, \xi^l_{c,k}\right)$. The warped pixel $\mathbf{p}^{k,l\prime}_n$ is matched to the current pixel coordinate $\tilde{\mathbf{p}}^{k,l\prime}_n := nn_{\mathcal{R}^{c,l}_i}\left(\mathbf{p}^{k,l\prime}_n\right)$ by the NNS function. The flow vector of two pixels is defined as,

$$\mathbf{F}^l_n := \mathbf{p}^{k,l\prime}_n - \tilde{\mathbf{p}}^{k,l\prime}_n \in \mathbb{R}^2. \tag{6}$$

A scalar value formed by $\mathbf{F}^l_n$ projected onto a unit gradient vector $\mathbf{g}^{c,l}\left(\tilde{\mathbf{p}}^{k,l\prime}_n\right)$ can be used as a signed residual $d^l_n \in \mathbb{R}$,

$$d^l_n = \mathbf{g}^{c,l}\left(\tilde{\mathbf{p}}^{k,l\prime}_n\right)^{\mathrm{T}} \cdot \mathbf{F}^l_n. \tag{7}$$

To make use of the absolute scale of the fixed stereo position, we define an additional residual term induced by the current right image. Analogous to the left case, a warped point onto the right current image of the query point $\mathbf{p}^{k,l}_n$ is denoted by $\mathbf{p}^{k,l\prime\prime}_n = w\left(\mathbf{p}^{k,l}_n, \xi^l_{c,k} \oplus \xi^l_{l,r}\right)$, and its matched right current pixel is represented by $\tilde{\mathbf{p}}^{k,l\prime\prime}_n := nn_{\mathcal{R}^{c,r}_i}\left(\mathbf{p}^{k,l\prime\prime}_n\right)$.

We also define a flow vector on the right current image,

$$\mathbf{F}^r_n := \mathbf{p}^{k,l\prime\prime}_n - \tilde{\mathbf{p}}^{k,l\prime\prime}_n, \tag{8}$$

and the right residual term can be written as,

$$d^r_n = \mathbf{g}^{c,r}\left(\tilde{\mathbf{p}}^{k,l\prime\prime}_n\right)^{\mathrm{T}} \cdot \mathbf{F}^r_n. \tag{9}$$

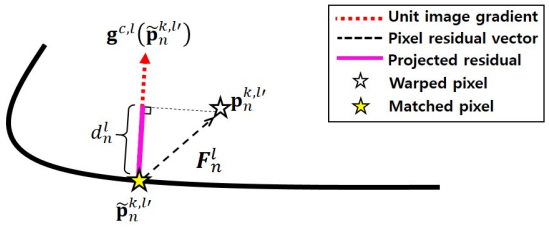

Fig. 7: **An illustration of a point-to-edge normal distance induced by a matched pair of points.**

By arranging total N_p pairs of residuals into one column vector, the residual vector can be formulated as,

$$\mathbf{r} = \left[d^l_1, ..., d^l_{N_p}, d^r_1, ..., d^r_{N_p}\right]^{\mathrm{T}} \in \mathbb{R}^{2N_p}. \tag{10}$$

The optimal motion can be estimated by minimizing the weighted sum of squared residuals,

$$\xi^*_{c,k} = \underset{\xi^l_{c,k} \in se(3)}{\arg\min}\ \mathbf{r}^{\mathrm{T}}\mathbf{W}\mathbf{r} \tag{11}$$

where $\mathbf{W}$ is a weighting matrix. The motion update $\delta\xi \in se\,(3)$ can be calculated by the second-order Gauss-Newton method as,

$$\delta\xi = -\left(\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{r}, \tag{12}$$

with the Jacobian matrix $\mathbf{J} = \frac{\partial \mathbf{r}}{\partial \xi} \in \mathbb{R}^{2N_p \times 6}$. The motion update is iteratively implemented until convergence,

$$\xi^l_{c,k} \leftarrow \xi^l_{c,k} \oplus \delta\xi. \tag{13}$$

To suppress inevitably occurring wrong match pairs, we use a t-distribution weighting scheme [25] for $\mathbf{W}$ and recalculate it by the current residual distribution at every iteration. If the matching results are correct in both stereo images, two gradient vectors on the matched pixels should have a consistent direction. From this, we can detect additional outliers by testing whether their inner product is under a certain margin or not,

$$\mathbf{g}^{c,l}\left(\mathbf{p}^{k,l\prime}_n\right)^{\mathrm{T}} \cdot \mathbf{g}^{c,r}\left(\mathbf{p}^{k,l\prime\prime}_n\right) < \eta \tag{14}$$

where a threshold value η is empirically set to 0.99 in this work, corresponding to about ten degrees angular difference.

B. Edge Inverse Depth Reconstruction and Propagation

As reported in [6], a laterally-fixed stereo can yield reliable 3-D information only for vertical edges because horizontal features cannot be distinguished by the static stereo. For the sake of complete 3-D depth maps in all directions, we follow the static and temporal stereo inverse depth estimation scheme [6]. The depth reconstruction procedure consists of four steps as illustrated in Fig. 8.

For probabilistic updates, we assume that an inverse depth observation ρ follows the normal distribution with the standard deviation σ around itself. We use the geometry error model to compute σ proposed in [13]. Note that the disparity is searched by evaluating the normalized cross correlation (NCC) with a 5×9 patch along an epipolar line, and we refine the sub-pixel disparity by using the parabolic interpolation.

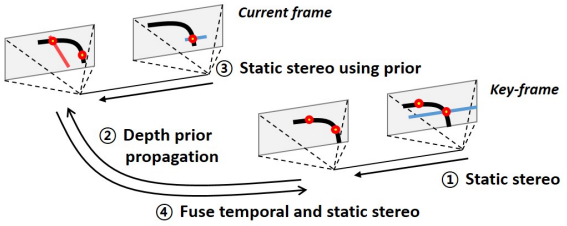

Fig. 8: **Static and temporal stereo configurations.**

Using the static stereo of the key frame, we first find the initial inverse depth ρ_k with σ_k. In this step, only vertical edges can be reliably reconstructed (step ① of Fig. 8).

Then, the previous observation ρ_k is warped to the left current image for temporal stereo update. In Fig. 8, by the static stereo configuration between current and key frames, horizontal edges can now be recovered and ρ_k can be updated if a red generalized epipolar line for each observation is almost perpendicular to the edge profile as depicted in Fig. 8 (steps ② and ③).

The inverse depth value ρ_c with σ_c detected by the temporal stereo is re-projected back, and updated with ρ_k to find an optimal estimation ρ_k^*,

$$\rho_k^* = \frac{\sigma_k^2\rho_c + \sigma_c^2\rho_k}{\sigma_k^2 + \sigma_c^2},\ \sigma_k^{*2} = \frac{\sigma_k^2\sigma_c^2}{\sigma_k^2 + \sigma_c^2}, \tag{15}$$

where σ_k^* is the fused standard deviation (step ④ of Fig. 8).

The inverse depth values are consecutively updated by this procedure for every incoming image, and if the new key frame is received, inverse depth values are propagated to the new key frame like [6].

VI. Performance Analysis

We found that the overall VO performance is affected by three dominant factors: 1) usage of multiple quadtrees, 2) storing matched nodes, and 3) the edge culling method with the minimum length l_{min} for edgelets. In this section, we evaluate the benefits of each factor with extensive variations of parameter settings. All computations are conducted in C++ with `-O2` compiler flag on AMD Ryzen 5 3.6 GHz CPU.

A. Analysis 1: normal quadtree vs. multiple quadtrees

In this analysis, we demonstrate the enhanced performance of the matching process by using the proposed multiple quadtrees. We consider two settings: a normal quadtree and multiple quadtrees without storing the visited nodes. To separately evaluate each part, we only use labeled raw edges without the edge culling method in this analysis.

In the beginning, for an intuitive example, we show matching results on the polyhedron image with the 15 degrees of the camera roll in Fig. 9. The result of using the proposed quadtrees shows the more robustness to rotations and qualitatively desirable matching tendencies.

To evaluate quantitatively, we warp the monotonous and cluttered images in Fig. 10 along u, v, and camera roll axes and compare the iterations and elapsed times required for the matching sequences of the ICP algorithm to converge. We apply 15 pixel deviations along u and v axes, and 10 degrees rotations with respect to the roll axis.

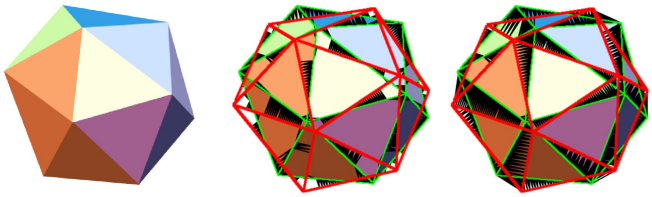

Fig. 9: (a) a polyhedron model, (b) a matching result using the normal quadtree, (c) a matching result using the multiple oriented quadtrees. Key and current edges are in green and red, respectively. Black lines connect matched pairs.

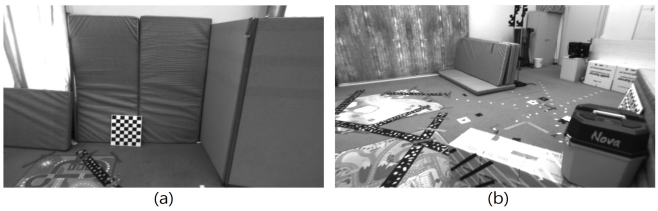

Fig. 10: **The selected two images on the EuRoC V1_01 dataset for quantitative evaluations.** (a) a simple image and (b) a cluttered image.

According to the results on the monotonous image described in Figs. 11(a-c), the number of iterations slightly decreases by using the multiple quadtrees. In contrast, the improvements by the multiple quadtrees on the cluttered image are more prominent in regions with large displacement as shown in Figs. 12(a-c). Note that the time consumption of the multiple quadtrees decreases by about 30 % on average compared to the normal one. We notice that the number of calculations for the NNS function is considerably reduced because a query point is compared to a fraction of reference points only within its related tree out of eight trees.

For the rolling motions larger than ± 5 degrees, iterations are saturated as in Fig. 12(c), which means that the ICP algorithm falls into local minima. In our experiences, some very cluttered regions in Fig. 10(b) yield a lot of wrong matches and the estimation fails in this case. We will address it in Section VI-C.

B. Analysis 2: effect of storing the previously matched nodes

We demonstrate further improvements on the matching speed by storing the previously matched nodes. In this analysis, we use the same simulation settings in the previous section and only switch on the node storage functionality.

As can be seen in Figs. 11 and 12, the number of iterations remains almost the same with the pure multiple quadtrees because the result of starting from the stored nodes is theoretically identical to starting from the root node. Thanks to the warm-start from the stored nodes, the number of nodes traverse considerably decreases as depicted in Fig. 6, and consequently, the time consumption is further reduced up to 70 % compared to the normal quadtree in most cases.

C. Analysis 3: effect of the edge culling method

We now evaluate the effect of the edge culling method on the efficiency and robustness of the ICP process by changing l_{min}. In this analysis, we fix l_{max} to 30 and use $l_{min} = \{5, 15, 25\}$. Compared to the previous simulation setting, the only difference is to replace raw edges with the culled edges.

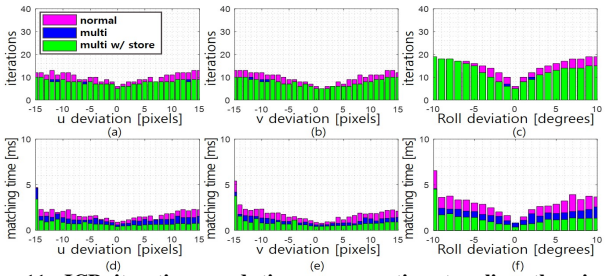

Fig. 11: **ICP iterations and time consumption to align the simple image in Fig. 10(a)**. (a-c) ICP iterations versus camera motions, (d-f) time consumption versus camera motions. Overall 9,031 query points are used.

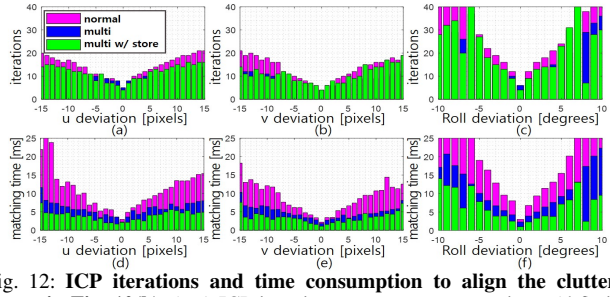

Fig. 12: **ICP iterations and time consumption to align the cluttered image in Fig. 10(b)**. (a-c) ICP iterations versus camera motions, (d-f) time consumption versus camera motions. Overall 33,602 query points are used.

The edge culling results with the three l_{min} values are reported in Fig. 14. The cluttered responses on the floor and false responses on the wall are effectively suppressed by the increasing l_{min} while the prominent edgelets remain.

Fig. 13 presents the comparison results. The results of $l_{min} = 5$ case are similar to the result of the setting without the edge culling method because many complex responses still exist in the left image like Fig. 14. For higher values of l_{min}, the number of iterations and the calculation time significantly decrease despite large motions thanks to the reduced cluttered edges as depicted in the right two images of Fig. 14. From these results, we can conclude that the edge culling method considerably enhances both the robustness and efficiency of the proposed edge-based ICP motion estimation.

VII. Implementation Results

We evaluate the overall performance of our method using EuRoC stereo datasets [26]. To verify the practical usability, we additionally demonstrate our method on author-collected datasets gathered in the low-textured indoor situations. We compare the proposed method with two state-of-the-art stereo VO algorithms: stereo ORB-SLAM2 [3] and stereo DSO [27]. To compare in terms of pure VO, the SLAM functionality of the ORB-SLAM is switched off. For the quantitative comparison of VO performance, we use the relative pose error (RPE) proposed in [28]. We publicly share our datasets used in the paper at `https://chkim.net/iros2020`.

A. EuRoC datasets

To show the robustness to the illumination changing, we additionally implement the previously proposed illumination changing model used in [15], and the modified datasets are distinguished by suffix of *change*. V1_01 contains moderated

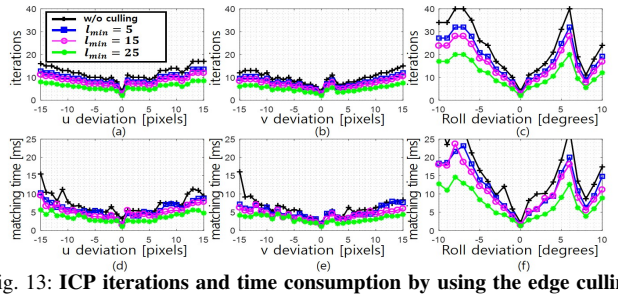

Fig. 13: **ICP iterations and time consumption by using the edge culling method to align the cluttered image in Fig. 10(b)**. (a-c) ICP iterations versus camera motions, (d-f) time consumption versus camera motions.

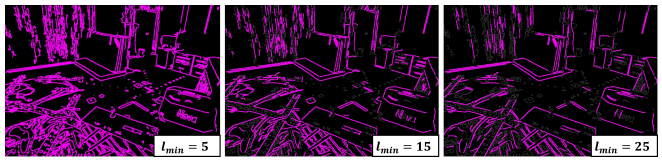

Fig. 14: **The results of the edge culling method with $l_{min} = \{5, 15, 25\}$.**

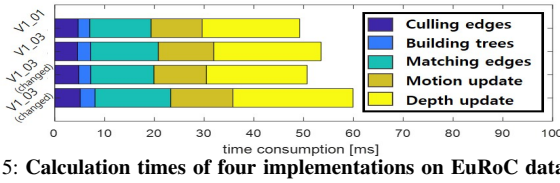

Fig. 15: **Calculation times of four implementations on EuRoC datasets.**

TABLE I: Performance comparison on EuRoC datasets. **Bold** denotes the best performance for each dataset.

Relative pose errors (RPE) [m/s]			
Dataset	Ours	ORB(stereo) [3]	stereo-DSO [27]
V1_01	0.038	0.031	**0.024**
V1_03	**0.046**	0.052	fail
V1_01(*change*)	**0.031**	0.055	fail
V1_03(*change*)	**0.055**	0.068	fail

motions and illuminations with abundant textures, and V1_03 has severe blurs from aggressive motions and illumination changes induced by the auto exposure control. The comparison results are shown in Table. I. According to the results, the proposed method shows comparative performance with the others in the normal V1_01. In V1_03, due to the high illumination, DSO fails to operate when auto exposure control excessively intervenes. ORB-VO continues to track motions for all datasets, however, the performances on datasets with changed illuminations are significantly degraded. Nevertheless, the proposed VO shows robust and accurate performance throughout the datasets. The average calculation times per stereo frame are about 50-60 ms for EuRoC datasets as seen in Fig. 15.

B. Author-collected datasets

To verify the real-world applicability of the proposed method, we collected challenging man-made scenes that include few free-formed edges only. Because there is no ground truth trajectory, we carefully move a camera to maintain the same height throughout the loop, and intentionally set the starting and end positions identical to check whether the results are consistent. As depicted in Fig. 16, the rectangular skeleton of the corridor is accurately recovered. Moreover, our method maintains the stable height estimate and exactly returns back to the starting point blue circle without any help of re-localization ability like SLAM.

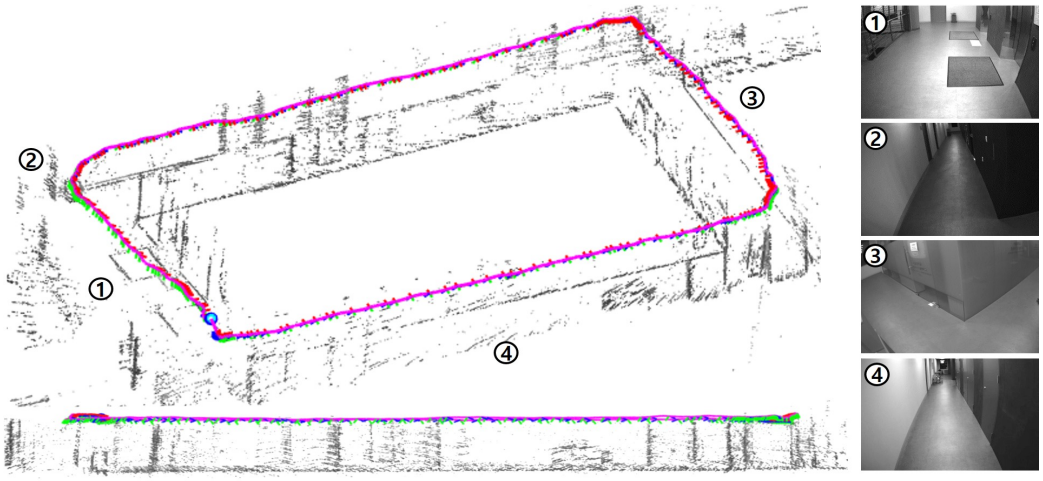

Fig. 16: **VO and 3-D reconstruction results on the author-collected dataset.** The estimated trajectory is in magenta, ranging 7 m×12 m.

VIII. Conclusions

In this paper, we proposed an efficient edge-based stereo visual odometry using the efficient multiple-quadtree structure. Extracted edges were classified into eight overlapped subsets with respect to their image gradient directions. To mitigate high ambiguity on matching edges, we segmented them into eight orientation bins with shared regions between neighboring bins. In addition, the matching procedure was accelerated by multiple-quadtree structures memorizing visited nodes. To effectively reduce a large number of edge pixels observed in an image, we only culled hundreds of well distributed prominent edgelets. Camera motions were estimated by minimizing stereo edge divergences, and our method simultaneously updated edge inverse depths by the static and temporal stereo. We demonstrated the robust and accurate performance of our method by using EuRoC datasets and author-collected datasets with almost no texture and severe brightness changes.

Acknowledgment

This work was supported by Institute of Information Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2019-0-00399, Development of A.I. based recognition, judgement and control solution for autonomous vehicle corresponding to atypical driving environment).

References

[1] G. Klein, and D. Murray. "Parallel tracking and mapping on a camera phone," in IEEE ISMAR, 2009.
[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: fast semi-direct monocular visual odometry," in IEEE ICRA, 2014.
[3] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: an open-source slam system for monocular, stereo, and rgb-d cameras," IEEE Transactions on Robotics, vol. 33, no. 5, 2017, pp. 1255-1262.
[4] R. A. Newcombe, S. Lovegrove, A. J. Davision, "Dtam: dense tracking and mapping in real-time," in IEEE ICCV, 2011.
[5] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in IEEE ICCV, 2013.
[6] J. Engel, J. Stuckler, and D. Cremers, "Large-scale direct slam with stereo cameras," in IEEE/RSJ IROS, 2015.
[7] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in IEEE ISMAR, 2011.
[8] J. Witt and U. Weltin, "Robust stereo visual odometry using iterative closest multiple lines," in IEEE/RSJ IROS, 2013.
[9] I. Dryanovski and R. G. Valenti and J. Xiaom, "Fast visual odometry and mapping from rgb-d data," in IEEE ICRA, 2013.
[10] S. Li and D. Lee, "Fast visual odometry using intensity-assisted iterative closest point," IEEE Robotics and Automation Letters, vol.1, no.2, 2016, pp. 992-999.
[11] J. Engel, T. Schops, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in ECCV, 2014.
[12] R. Gomez-Ojeda, J. Briales, and J. Gonzalez-Jimenez, "Pl-svo: Semi-direct monocular visual odometry by combining points and line segments," in IEEE/RSJ IROS, 2016.
[13] S. Yang and S. Scherer, "Direct monocular odometry using points and lines," IEEE ICRA, 2017.
[14] X. Wang, et al., "Edge Enhanced Direct Visual Odometry," in BMVC. 2016.
[15] C. Kim, P. Kim, S. Lee, and H. Kim, "Edge-based robust rgb-d visual odometry using 2-d edge divergence minimization," in IEEE/RSJ IROS, 2018.
[16] J. Canny, "A computational approach to edge detection," Readings in computer vision, Elsevier, 1987, pp. 184-203.
[17] Y. Zhou, H. Li, and L. Kneip, "Canny-vo:visual odometry with rgb-d cameras based on geometric 3-D–2-D edge alignment," IEEE Transactions on Robotics, vol. 35, no. 1, 2019, pp. 184-199.
[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in IEEE ICCV, 2011.
[19] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," Journal of Visual Communication and Image Representation, Elsevier, vol. 24, no. 7, 2013, pp. 794-805.
[20] J. Civera, A.J. Davision, and J.M.M. Montiel, "Inverse depth parametrization for monocular slam," IEEE Transactions on Robotics, vol. 24, issue. 5, 2008, pp. 932-945.
[21] Y. Lu and D. Song, "Robust rgb-d odometry using point and line features," in IEEE ICCV, 2015.
[22] M. Tomono, "Robust 3d slam with a stereo camera based on an edge-point icp algorithm," in IEEE ICRA, 2009.
[23] M.Kuse and S. Shen, "Robust camera motion estimation using direct edge alignment and sub-gradient method," in IEEE ICRA, 2016.
[24] J. J. Tarrio and S. Pedre, "Realtime edge-based visual odometry for a monocular camera," in IEEE ICCV, 2015.
[25] Y. Zhou, L. Kneip, and H. Li, "Semi-dense visual odometry for RGB-D cameras using approximate nearest neighbour fields," in IEEE ICRA, 2017.
[26] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. Achtelik and R. Siegwart, "The euroc micro aerial vehicle datasets," International Journal of Robotics Research, vol. 35, issue. 10, 2016, pp. 1157-1163.
[27] R. Wang, M. Schworer, and D. Cremers, "Stere dso: large-scale direct sparse visual odometry with stereo cameras," in IEEE ICCV, 2017.
[28] J. Sturm, et al. "A benchmark for the evaluation of rgb-d slam systems," in IEEE/RSJ IROS, 2012.