2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

VioLA: Aligning Videos to 2D LiDAR Scans

Jun-Jee Chao†, Selim Engin†, Nikhil Chavan-Dafle, Bhoram Lee, and Volkan Isler

***Abstract*— We study the problem of aligning a video that captures a local portion of an environment to the 2D LiDAR scan of the entire environment. We introduce a method (VioLA) that starts with building a semantic map of the local scene from the image sequence, then extracts points at a fixed height for registering to the LiDAR map. Due to reconstruction errors or partial coverage of the camera scan, the reconstructed semantic map may not contain sufficient information for registration. To address this problem, VioLA makes use of a pre-trained text-to-image inpainting model paired with a depth completion model for filling in the missing scene content in a geometrically consistent fashion to support pose registration. We evaluate VioLA on two real-world RGB-D benchmarks, as well as a self-captured dataset of a large office scene. Notably, our proposed scene completion module improves the pose registration performance by up to 20%.**

I. Introduction

Generating 3D semantic maps of home environments enables numerous applications in immersive technologies, home-robotics, and real estate. Even though commercial grade solutions exist for generating 3D maps of home environments, they require expensive and specialized hardware, as well as meticulous scanning procedures. Alternatively, using widely available cameras such as those on mobile phones can be used to reconstruct a local area. However, scanning an entire house with a single camera is difficult; getting all details in one scan is tedious even for a single room. Moreover, merging scans across rooms is error-prone due to the lack of images with overlapping features. In this paper, we present a method to overcome these challenges using a 2D floor layout, such as those obtained by Robot Vacuum Cleaners (RVCs), and user-scanned videos recorded *independently* with an RGB-D camera.

We study the following problem: Given a 2D LiDAR map of an environment and an RGB-D image sequence recorded from a local section of the same environment, the task is to align the pose of the first image to the LiDAR map as shown in Fig. 1. The benefits of aligning videos to 2D LiDAR scans are twofold. First, image sequences with dense scene information can be used to augment 2D LiDAR maps with 3D geometry, texture and semantics information. This is useful for example to disambiguate walls and furniture in 2D LiDAR maps, which may allow for better experiences of user-robot interaction. Second, the LiDAR map can serve as a common coordinate frame to align short clips of videos captured from different locations of the same house. Therefore, the entire house can be scanned by aligning independent video sequences to the common LiDAR map, for example, as seen in Fig. 2.

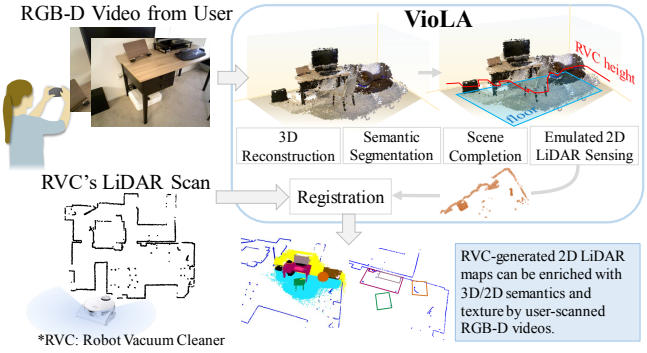

Fig. 1: Given an RGB-D image sequence from an indoor scene and a 2D LiDAR scan of the same environment, our task is to align the video to the LiDAR map by registering the first camera pose to the LiDAR coordinate frame. After registration, our method allows augmenting the LiDAR map with 3D geometry, texture and semantics information.

Currently, there is no existing solution for the problem of registering raw 2D point clouds obtained by LiDAR measurements to 3D reconstructions from RGB-D image sequences. This alignment task poses unique challenges in several aspects. In particular, point clouds from 2D LiDAR scans contain information from the entire floor plan of an apartment but only at a fixed height, with no semantic context. Whereas reconstruction point clouds have denser 3D information but only from a local area. Therefore, neither point set is a superset of the other, which makes matching or registration challenging. Furthermore, the sensing modalities are different and the representation to use for registering a 3D reconstruction to a LiDAR map is not trivial.

We propose VioLA (Video-LiDAR Alignment), a method for aligning videos to 2D LiDAR maps. The underlying idea of VioLA is to reconstruct the 3D from RGB-D and perform 2D ray casting to emulate LiDAR measurements of RVC. If the video scan captures majority of the scene at the RVC height, then we can use the ray cast hit points directly to register with the LiDAR point cloud. However, this may not be always the case. User-captured videos may not contain enough information of the scene at the RVC height leading to poor registration performances. To fill in these missing regions of the reconstruction, VioLA leverages a pre-trained text-to-image model to synthesize images at novel viewpoints depending on the scene geometry and lifts them to 3D with a depth completion module in a geometrically consistent fashion. Then, the completed point cloud can be used to perform 2D ray casting at the RVC height and 2D-2D point cloud registration with respect to the LiDAR map. The contributions in this paper can be summarized as follows.

† indicates equal contribution.

Work done while authors were with Samsung AI Center NY, USA. Email:
chao0107@umn.edu, kazimselimengin@gmail.com

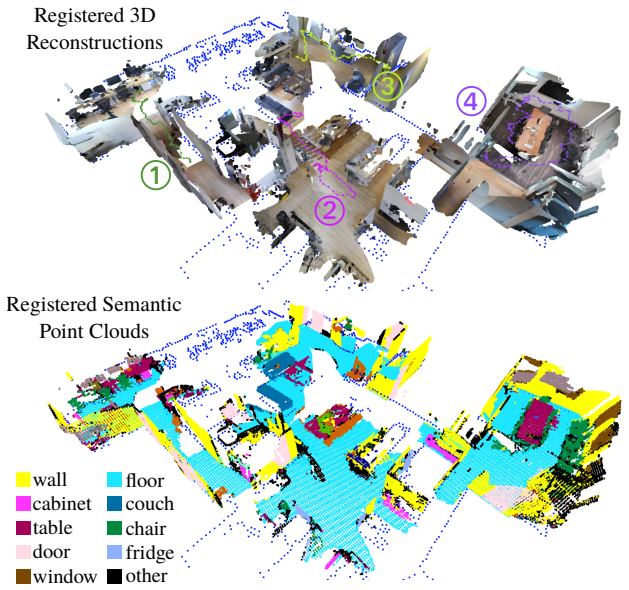

Fig. 2: Using VioLA, 3D reconstructions obtained over multiple scans (4, in this case) can be fused via registration to the LiDAR map (blue dots). In addition, our method can generate the semantic labels for the fused point cloud. The colored lines in the top figure show the camera trajectories and the point colors in the bottom figure indicate the classes given in the legend.

- We introduce the novel task of registering videos to 2D LiDAR scans, which is motivated by *i*) enriching LiDAR maps with texture and semantics for better user experience, and *ii*) making use of LiDAR maps as a common frame to localize short clips of videos taken from different locations in an indoor environment.
- We present VioLA: A technique to register 3D reconstructions of local areas to 2D LiDAR maps captured from indoor environments such as apartments.
- As part of VioLA, we design a strategy for viewpoint selection and leverage a state-of-the-art text-to-image model to synthesize images at locations with missing information to aid our point cloud registration module.
- We evaluate VioLA on two benchmarks using real videos and synthetic LiDAR maps, in addition to a self-collected dataset of real videos and LiDAR scans.

II. Related Work

In this section, we provide an overview of existing work that is related to our problem setup.

Cross-modality registration. There is a line of research that studies the problem of cross-modality registration between 2D images and 3D point clouds. [1] uses 2D-3D line correspondences to estimate the camera pose. Other recent works apply deep learning to register 2D images to 3D point clouds [2]–[7]. However, none of these methods directly addresses the problem of registering videos to 2D point clouds. More similar to our setting, [8] and [9] make use of floor plan images for aligning RGB-D scans. While it operates on building-scale floor plans, the work in [8] requires RGB-D panorama images that cover a large portion of the floor plan. On the other hand, [9] takes as input an RGB-D sequence that scans the entire indoor scene to refine the camera poses, which can be a tedious task for end users.

Point cloud registration. Our problem can be formulated as point cloud registration by first reconstructing the 3D scene from the video. However, it is still non-trivial to perform pose registration between 3D point clouds of local area and 2D LiDAR point clouds of the entire floor plan.

ICP [10], [11] solves the registration problem by building correspondences between closest points, which is sensitive to initialization and prone to local minima. To overcome the initialization issue, FGR [12] and TEASER [13], [14] optimize one-to-one correspondence-based object functions with the help of 3D point feature descriptors like PFH [15] and FPFH [16]. Instead of building one-to-one correspondences in a single shot, [17] proposes an optimization loss function that consider multiple correspondences at the initial stage.

Deep learning is also applied to learn point feature descriptors [18]–[22]. PointNet [23] is a popular architecture to extract features directly from raw point clouds [18], [24]. Some other works apply DGCNN for feature learning [21], [25]. Recently, more works have been developed for partial-to-partial point cloud registration [26]–[34]. However, most of them focus on 3D-to-3D registration, which are not directly applicable to the 3D-to-2D case studied in this paper.

Point cloud completion. There has been a lot of attention on the task of object-level point cloud completion [35]–[40], for which priors for object shapes are learned from data. Scene-level completion methods have been proposed with a focus on self-driving car datasets [41], [42]. Other works that focus on single-view scene completion [43]–[47] are usually limited to scene completion only in the field of view. However, completing the scene in the field of view is not enough for our task, as we need to fill in the missing regions that are not seen by the cameras to aid pose registration. There are recent methods that utilize text-to-image inpainting models and depth estimation models to generate scenes given text prompts [48], [49]. However, they require predefined camera trajectories, and the synthesized 3D content is not grounded to any actual scene, which makes them difficult to use for pose registration purposes.

III. Method

The input to VioLA consists of an RGB-D video sequence $V = \{I_1, \ldots, I_n\}$ and a 2D point cloud $\mathbf{P} \in \mathbb{R}^{2\times M}$ captured by the LiDAR sensor on a ground robot such as a robot vacuum cleaner (RVC). The output is an estimate of the transformation ${}^{L}\mathbf{T}_{V} \in SE(2)$ that aligns the pose of the first image in the video to the LiDAR map, as well as a 3D semantic map of the scene registered to the LiDAR coordinate frame. For brevity, we denote the pose estimates by $\mathbf{T}$ in the remainder of the paper. The main module of VioLA includes: a) 3D reconstruction and semantic segmentation from the RGB-D sequence, b) 2D point cloud extraction from (a) by floor estimation and ray casting, and c) 2D registration of point clouds from (b) and LiDAR point cloud from the RVC. Additionally, to address the case where the registration suffers from having too few points from ray casting, VioLA utilizes d) a 3D scene completion module using a pre-trained inpainting model to synthesize the 3D geometry in missing

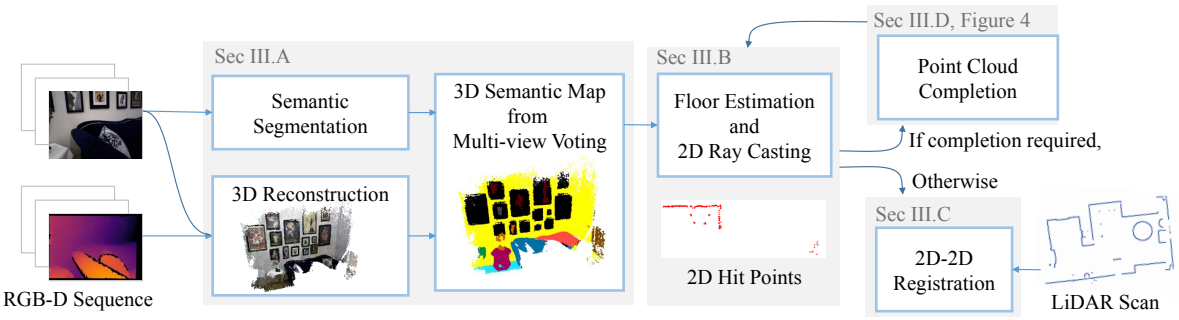

Fig. 3: **Method Overview.** VioLA takes as input an RGB-D sequence and a 2D LiDAR scan from uncalibrated sensors, and aligns these two sources of measurements. Our approach first reconstructs the 3D scene and extracts semantics from each image which are then fused by multi-view voting to obtain a semantic point cloud. After finding the floor surface, it performs 2D ray casting at a desired height to emulate LiDAR measurements. If the hit points of these ray casts cover a large portion of the reconstruction they are used directly to align to the input 2D LiDAR scan with our 2D-2D point cloud registration module. If not, VioLA uses our novel strategy for view selection coupled with inpainting-based scene completion. The final result of our method is the $SE(2)$ transformation relating the first camera frame to the LiDAR coordinate frame, as well as the semantic map.

regions. An overview of our approach is shown in Fig. 3. In the following subsections, we detail each of the modules.

A. 3D reconstruction, semantic segmentation

We use off-the-shelf SLAM algorithms [50]–[52] to reconstruct 3D point clouds from the RGB-D videos. We further estimate the segmentation class for each reconstructed 3D point by fusing 2D semantic segmentation into 3D. The semantic segmentation provides rich information about the scene including the floor which is critical for the next step (Sec. III-B). Following [53], we first apply 2D color-based augmentations on the key frames of the SLAM algorithm. These augmented images are then used to predict the semantic masks using Mask2Former [54]. These estimated masks of a single frame can then be fused into a categorical probability distribution $\mathrm{p}_u = [p_u^{(1)}, ..., p_u^{(C)}]$ over C classes for each pixel u paired with a confidence score $\mathrm{s}_u \in [0, 1]$. To determine the semantics of the reconstructed 3D points, we first project each point $\mathbf{x}$ onto the m SLAM camera frames in which the point is visible and store the set of pixels $\mathcal{U} = \{u_j\}_{j=1}^{m}$ that $\mathbf{x}$ projects to. Then, we sum over the corresponding probability distribution weighted by the normalized confidence scores. The final semantic label $c(\mathbf{x})$ of the point $\mathbf{x}$ is determined as:

$$c(\mathbf{x}) = \underset{i \in \{1,...,C\}}{\arg\max} \sum_{u \in \mathcal{U}} w_u \cdot p_u^{(i)} \tag{1}$$

where $w_u = \mathrm{s}_u / \sum_{v \in \mathcal{U}} \mathrm{s}_v$ is the normalized confidence score, and $u = \Pi(\mathbf{x})$ is the projection of $\mathbf{x}$ in each of the m frames where $\mathbf{x}$ is visible.

B. Floor normal estimation and RVC viewpoint projection

After reconstructing the scene from the RGB-D video, we simulate what the RVC would see in this scene in order to perform 2D point cloud registration with the LiDAR map. To do so, we first estimate the floor surface by fitting a plane to the points that are labeled as floor. The ground is assumed to be visible in the video, which is usually the case in casually recorded videos as shown in the experiments section. Then we move all SLAM camera poses down to a predefined RVC height while preserving only the yaw angles, i.e., the y-axis of the camera is parallel to the estimated floor normal. From these downprojected camera poses, we cast rays from the camera centers to the reconstructed 3D points to emulate LiDAR hit points.

C. Pose initialization and optimization

Since SLAM reconstructs the 3D points with respect to the first camera frame, the hit points $\mathbf{H} \in \mathbb{R}^{2 \times N}$ obtained from the 2D ray casting module are also in the same frame. VioLA performs 2D-2D point cloud registration between the LiDAR map $\mathbf{P}$ and the ray cast hit points $\mathbf{H}$. Similar to most iterative algorithms for registration, our method requires an initial guess for estimating the relative pose between the two point clouds. We rasterize both the LiDAR map and the simulated hit points into images. We then perform template matching and select the top k poses with the highest normalized cross correlation scores as our initial poses. With these initial poses, we apply GPU parallelization to simultaneously update the poses by minimizing the following loss function using gradient descent:

$$\mathcal{L} = \sum_{i=1}^{k} d(\mathbf{T}_i \cdot \mathbf{H}, \mathbf{P}) \tag{2}$$

where $d(\cdot, \cdot)$ denotes the one-directional Chamfer distance given by $d(X, Y) = \frac{1}{|X|} \sum_{\mathbf{x} \in X} \min_{\mathbf{y} \in Y} ||\mathbf{x} - \mathbf{y}||_2$, and the point clouds are in homogeneous coordinates. We compute the Chamfer distance from the hit points to the LiDAR map after applying the estimated transformations. Finally, the estimated pose is selected as the pose that minimizes the one-directional Chamfer distance after optimization.

This Chamfer-based optimization can be replaced with other pose optimization methods like ICP. In a preliminary experiment, we found out that ICP performs very similarly to our method given the same initialization. However, ICP on average takes a total of 49.5 seconds for matching with $k = 100$ initializations, while our method takes 15.5 seconds on average. Therefore, we use the Chamfer-based pose optimization.

D. Scene completion based on image inpainting

As we will demonstrate later in experiments in Section IV-A, reconstructed points at the RVC height is critical for

registration success. However, these points might be missing due to the video not capturing the lower part of the scene or the SLAM algorithm suffering from matching featureless points. In this section, we show how state-of-the-art inpainting methods can be used to provide this missing information. Note that we are not trying to fill in every detail of the scene. Instead, we show how one can judiciously select good viewpoints so that *i*) resulting images contain a sufficient amount of observed pixels so that inpainting is successful, and *ii*) the inpainted pixels correspond to locations that would have been observed by the RVC.

Virtual viewpoint selection. The key idea of our strategy for viewpoint selection is to incrementally add 3D content from a sequence of virtual views. Our strategy involves starting from a view that sees the boundary between observed and unobserved pixels, then moving the camera back to increase the field of view, and finally rotating to cover more unobserved pixels.

To describe our strategy for placing virtual cameras, we define five types of point sets (see Fig. 4). a) The 2D ray cast hit points at the RVC height obtained from Section III-B, b) *downprojected points*: the reconstructed points projected onto a horizontal plane at the RVC height, c) *missing points at the RVC height*: the subset of (b) whose vicinity does not include (a), d) *boundary point*: the point in the largest cluster of (a) that is closest to (c), and e) *frontier points*: which approximately represent the boundary of the area to be completed at the RVC height.

We are interested in placing the frontier points in such a way that as the virtual cameras view these points, they gradually cover the missing part extending from the seen part. Therefore, we first find the boundary point that represents the boundary of the seen part and area to be completed. Next, to place the frontier points, we fit a concave hull [55], [56] on the downprojected points. Then, we sample points on the hull boundary starting from the point that is closest to the boundary point and extend towards the direction where there is no ray cast hit points for 2m.

Finally, to generate the virtual camera trajectory, we search among all the frames in the video to find cameras that see the boundary point and pick the one with least pitch angle (i.e., camera looking down) as the first camera. Then, we move the camera back along its z-axis with a step size of 0.2m until it sees half of the frontiers, and apply rotation along the floor normal direction with a step size of 10° until the camera sees all the frontiers or it reaches the maximum rotation threshold (30°). As shown in Fig. 4, these sequential viewpoints are later used in the point cloud completion module.

Point cloud completion. Given a target viewpoint, we first render an RGB-D image from the reconstructed point cloud using a differentiable point cloud renderer [57], along with a binary occupancy image. The rendered RGB image is taken as input to the Stable Diffusion inpainting model [58] to fill in the missing part indicated by the mask and a text prompt. In all our experiments we use the prompt *"a realistic photo of an empty room"*, so as to discourage the inpainting model from hallucinating objects that do not exist in the actual

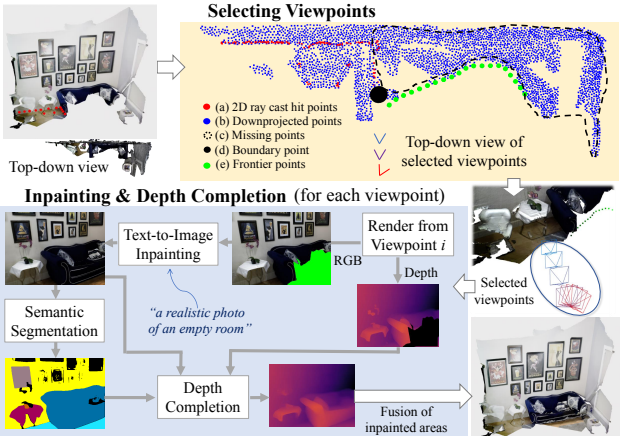

Fig. 4: Given the 3D reconstruction, VioLA first places virtual viewpoints whose union cover the frontier points. At each viewpoint, VioLA renders an RGB-D image from the 3D points and performs image inpainting followed by depth completion with the help of the predicted semantics. Finally, it fuses the newly inpainted pixels back into 3D.

scene. Next, we use a monocular depth estimation method, IronDepth [59], on the inpainted RGB images for lifting the synthesized scene content to 3D. We backproject and fuse the inpainted areas into the reconstructed point cloud with the known camera parameters in an auto-regressive fashion. To better align the geometry of the new content to the reconstructed point cloud, we use the depth values rendered from the initial reconstruction as input to the depth estimation model and complete the rest of the pixels. Furthermore, we obtain a semantic segmentation of the inpainted image, and for the pixels predicted to be in the *floor* class, we use their depths to the estimated floor plane as additional supervision to the depth completion module.

After inpainting new scene content from all the virtual viewpoints and completing the reconstructed point cloud, we again cast rays from the RVC height cameras to obtain the hit points as mentioned in Section III-B and follow the same pose estimation method in Section III-C to perform 2D point cloud registration with the LiDAR map.

IV. Experiments

In this section we describe the set of experiments we conducted to evaluate VioLA's performance.

Datasets. In our experiments, we use three real-world RGB-D datasets: Redwood [60], ScanNet [61] and a self-captured dataset from a large office. Both Redwood and ScanNet provide ground truth camera poses as well as the reconstructed meshes. We simulate the LiDAR maps by placing virtual sensors at the RVC height in the provided meshes and cast rays in all directions to collect hit points. To imitate the noise observed in real LiDAR scans, we further perturb the hit points with a 2D Gaussian with standard deviation of 1cm and drop points with a probability of 10%. Note that the provided meshes from ScanNet contain missing parts and holes in the scene, therefore, the simulated LiDAR map is not perfect even before adding noise. We use all apartment scenes from ScanNet and manually select several other scenes that are large and have better reconstructed

meshes, which result in 36 scenes. For each scene, we randomly sample 6 videos from the provided camera stream. We include all scenes in the Redwood dataset and sample 20 videos from each scene. We additionally include 10 difficult videos for Redwood that capture mostly only the upper part of the scene. Among all the sampled videos, there are only 1 from ScanNet and 2 from Redwood that do not see the floor, which justify our assumption in Section III-B that the floor is usually partially visible. We manually exclude these 3 videos. To collect data from the office scene, we mount a LiDAR on a ground robot and move it around the office to build the 2D map using GMapping [62], [63]. We also record 5 video sequences at different locations in the office with a hand-held RGB-D camera.

Metrics. We report mean and median of the rotation and translation errors between the estimated pose registration and the ground truth. We denote R_μ, R_{med} as the mean and median of the estimated 2D rotation error in angles, and T_μ, T_{med} as the corresponding translation error in meters. Additionally, we report the success rate (SR) where a predicted registration is counted as successful if it has rotation error less than 10° and translation error less than 0.3m.

A. *Camera pose estimation on real-world RGB-D scans*

We present quantitative results for 2D pose estimation on both Redwood and ScanNet. Since there is no existing work that directly estimates the relative pose between a video and a 2D LiDAR point cloud, we perform a comparative study using varied versions of our proposed methods as baselines. The *base method* denotes our method without the scene completion module.

To verify our assumption that the missing points at the RVC height is one of the major failure causes, we investigate the correlation between the pose registration error and a *coverage metric*. Specifically, for each sample, we align the ray cast points with the LiDAR map using the ground truth pose, then we measure the proportion of the LiDAR map covered by the ray cast points as the percentage of coverage. In Fig. 5, we plot the pose estimation error of the base method as a function of this coverage metric. It is clear that as the coverage reduces, more data lie above the error bound. This motivates our scene completion module for filling in the missing data and improving the registration accuracy.

Next, we consider the full VioLA pipeline on all the data by activating the scene completion module on every video. Indicated by "VioLA-all" in Table I, we see that the scene completion module can hurt the pose registration performance if applied to all the videos. One major failure case is when the scene already has sufficient coverage at the RVC height, the completion module will be forced to generate geometry beyond the boundary of the observed scene. Therefore, the synthesized areas might not match the actual scene.

To decide whether we should perform scene completion, we devised a decision criterion which effectively finds out if there are multiple local minima among the optimized poses in Section III-C. Specifically, we consider all poses and check

TABLE I: The effect of the scene completion module based on different activating criteria. Base method is without the scene completion module. VioLA-all applies the scene completion on all data and VioLA-w/ gt. applies on the data that is considered failed measured with ground truth pose. VioLA activates the completion module with the proposed decision criterion.

	Redwood				
	$R_\mu(^\circ)\downarrow$	$R_{med}(^\circ)\downarrow$	$T_\mu(m)\downarrow$	$T_{med}(m)\downarrow$	SR(%)↑
Base method	27.407	1.389	1.324	0.1	0.667
VioLA-all	27.934	1.695	1.264	0.148	0.648
VioLA	18.787	0.975	0.794	0.09	0.741
VioLA-w/ gt.	15.829	1.12	0.646	0.084	0.806
	ScanNet				
	$R_\mu(^\circ)\downarrow$	$R_{med}(^\circ)\downarrow$	$T_\mu(m)\downarrow$	$T_{med}(m)\downarrow$	SR(%)↑
Base method	15.631	1.557	0.778	0.069	0.805
VioLA-all	29.203	2.862	0.86	0.111	0.660
VioLA	14.311	1.46	0.488	0.073	0.833
VioLA-w/ gt.	10.451	1.343	0.433	0.064	0.879

if there are two poses which have small loss values and at the same time, are not close. We define $\mathbf{T}_i$ and $\mathbf{T}_j$ to be close if the relative rotation is smaller than θ_R and relative translation is smaller than θ_T. More concretely, we start with the pose $\mathbf{T}_1$ with the smallest loss value after optimization. We then remove its neighbors which are close from consideration and pick the second best pose $\mathbf{T}_2$. Let $\mathbf{L}_1$ and $\mathbf{L}_2$ be the loss values associated with $\mathbf{T}_1$ and $\mathbf{T}_2$ respectively. If these two poses have similar loss values (i.e. $|\mathbf{L}_1 - \mathbf{L}_2| < c$), then it means there are multiple local minima, therefore, scene completion is needed. To determine the values of θ_R, θ_T, c, we use 20% of data from ScanNet and perform parameter search. We select the parameter set that allows VioLA to have the minimum rotation error on this 20% of the data. Finally, we use $\theta_R = 20^\circ$, $\theta_T = 0.3$m and $c = 20$ for VioLA on all data. As shown in Table I, this activation criterion captures the cases that require completion well and therefore VioLA performs better than the base method on both datasets. Fig. 6 shows qualitatively how scene completion in VioLA helps pose registration on the scenes where the base method fails.

To verify the performance of the designed decision criterion for the completion module, we further conduct an experiment that activates the point cloud completion module using the pose registration error measured with ground truth, which represents the upper bound of VioLA's performance. "VioLA-w/ gt." in Table II shows that our designed decision criterion allows VioLA to perform close to its upper bound, and with a better developed criterion, the performance of VioLA can be further improved.

Floor map augmentation with VioLA. In addition to public datasets for which we simulated the LiDAR scans, we present qualitative results on the self-captured office scan with a noisy, real-world LiDAR map. We demonstrate

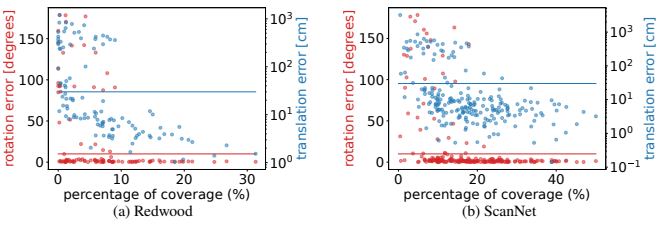

Fig. 5: The effect of the reconstruction's coverage of the scene on the registration performance. The red line indicates the rotation error bound of 10° and the blue line indicates the translation error bound of 0.3m. It is shown that more data lies above the error bound as coverage decreases.

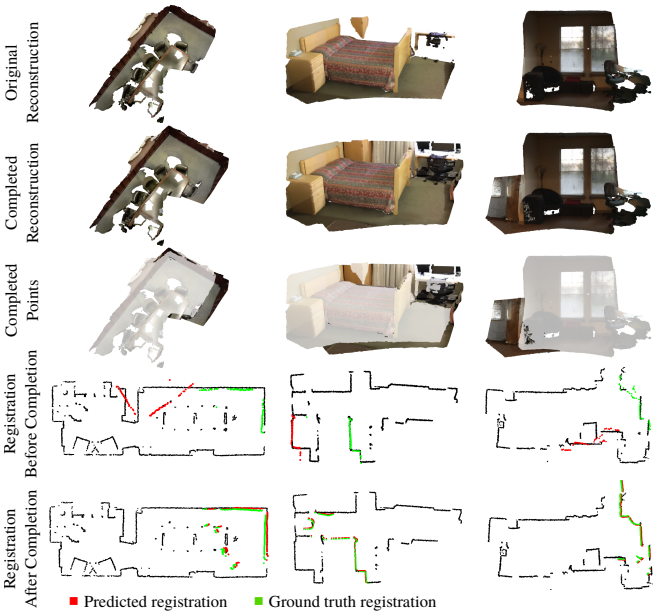

Fig. 6: Qualitative results of VioLA using and without using the point cloud completion module. We see that synthesizing the scene content at the RVC height improves the overall registration performance. The first two rows show the original and completed reconstructions, respectively, with the third row highlighting the added new points. In the last two rows, red points are the predicted and green ones are the ground truth registrations.

that VioLA can help augment the LiDAR map after pose registration. As shown in Fig. 2, 4 videos recorded separately are registered to the same LiDAR scan to achieve 3D reconstruction of a large scene. Furthermore, we show that semantics information can be added to the LiDAR map, which is useful for enabling more informative floor layouts.

B. Comparison of strategies for target viewpoint selection

We observe that the inpainting model is sensitive to how we select the novel viewpoints for rendering RGB images and for performing inpainting. Therefore, we investigate the effect of different viewpoint selection strategies. In Section III-B, we move the cameras down to the RVC height to cast rays for emulating LiDAR hit points. One idea is to directly use these virtual cameras at the RVC height to maximize the amount of new content. Another approach is to move the SLAM camera poses back by 0.5m along the camera's z-axis to increase the field of view while guaranteeing to keep a certain portion of the seen content. We compare these two additional view selection strategies with VioLA as mentioned in Section III-D. As shown in Table II, VioLA outperforms these two baseline viewpoint selection strategies. We observe that directly inpainting from the viewpoints at the RVC height gives the most information for the 2D hit points. However, sometimes these viewing angles are much different from the original camera poses, and therefore are relying on the inpainting model to synthesize a large empty space without enough information for grounding to the actual scene. Stepping back from the original camera poses is a way to guarantee that the input RGB images to the inpainting model retains certain amount of content. However, naively moving the cameras back can result in undesired

TABLE II: The effect of different viewpoint selection strategies for scene completion on pose registration performance. *Step back* moves SLAM camera poses back by 0.5m along the camera's z-axis. *RVC* projects the viewpoints down to the RVC height from the SLAM poses.

	Redwood				
	$R_\mu(^\circ)\downarrow$	$R_{med}(^\circ)\downarrow$	$T_\mu(m)\downarrow$	$T_{med}(m)\downarrow$	SR(%)↑
Step back	27.179	1.467	1.303	0.102	0.685
RVC	29.32	1.306	1.097	0.096	0.694
VioLA	**18.787**	**0.975**	**0.794**	**0.09**	**0.741**
	ScanNet				
	$R_\mu(^\circ)\downarrow$	$R_{med}(^\circ)\downarrow$	$T_\mu(m)\downarrow$	$T_{med}(m)\downarrow$	SR(%)↑
Step back	16.076	1.63	0.645	0.073	0.833
RVC	15.082	1.458	0.657	0.073	0.814
VioLA	**14.311**	**1.46**	**0.488**	**0.073**	**0.833**

viewpoints or viewpoints that do not help with filling in missing area at the RVC height.

In contrast, VioLA's viewpoint selection strategy allows the inpainting model to start from already seen parts and move toward the missing part at the RVC height with gradual viewpoint change. As shown in Fig. 6, VioLA is able to fill in reasonable geometry at the RVC height. Note that VioLA's main goal is to complete the points for the purpose of pose registration, instead of generating the whole scene. Therefore, although the completed area does not cover a large space, it can improve the registration performance as long as the missing areas at the RVC height are reconstructed well.

V. Conclusion

We presented VioLA for aligning RGB-D videos to 2D LiDAR maps obtained by RVCs. After building a 3D semantic map using the image sequence, VioLA performs 2D ray casting to emulate the LiDAR measurements, and inputs the hit points to our pose optimization module for registering to the LiDAR map. Our key observation is that the registration error correlates with the missing points at the RVC height. To fill in the missing information, we introduced a scene completion technique that leverages a pre-trained text-to-image model paired with a novel viewpoint selection strategy. We show that after using the completed point clouds, our registration module can solve up to 20% of the instances where it has failed before. In addition, we demonstrate that our method can be used to align short sequences of videos collected independently from different rooms, which enables augmenting LiDAR maps with 3D geometry and semantics.

While VioLA shows promising results for video to LiDAR alignment, it has some limitations that need to be addressed. First, we assume that a portion of the floor is visible in the video, since our method relies on estimating the floor plane. One way to eliminate this assumption is to use additional sensing such as IMUs to estimate the gravity direction, then performing ray casting at multiple heights as possible inputs for registration. Moreover, we note that using pre-trained inpainting and depth completion models can generate implausible geometries that may be detrimental to the pose registration performance. Although our strategy for selecting viewpoints mitigates this issue, we see few instances impacted by inaccurate completions of the scene, which leaves an exciting open problem for future studies.

REFERENCES

[1] Huai Yu, Weikun Zhen, Wen Yang, Ji Zhang, and Sebastian Scherer. Monocular camera localization in prior lidar maps with 2D-3D line correspondences. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4588–4594. IEEE, 2020.

[2] Mengdan Feng, Sixing Hu, Marcelo Ang, and Gim Hee Lee. 2D3D-MatchNets: Learning to match keypoints across 2d image and 3d point cloud. In *The IEEE International Conference on Robotics and Automation (ICRA)*, May 2019.

[3] Jiaxin Li and Gim Hee Lee. DeepI2P: Image-to-point cloud registration via deep classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15960–15969, 2021.

[4] Siyu Ren, Yiming Zeng, Junhui Hou, and Xiaodong Chen. CorrI2P: Deep image-to-point cloud registration via dense correspondence. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(3):1198–1208, 2022.

[5] Jinyu Miao, Kun Jiang, Yunlong Wang, Tuopu Wen, Zhongyang Xiao, Zheng Fu, Mengmeng Yang, Maolin Liu, and Diange Yang. Poses as queries: Image-to-lidar map localization with transformers. *arXiv preprint arXiv:2305.04298*, 2023.

[6] Yurim Jeon and Seung-Woo Seo. EFGHNet: A versatile image-to-point cloud registration network for extreme outdoor environment. *IEEE Robotics and Automation Letters*, 7(3):7511–7517, 2022.

[7] Kuangyi Chen, Huai Yu, Wen Yang, Lei Yu, Sebastian Scherer, and Gui-Song Xia. I2D-Loc: Camera Localization via Image to LiDAR Depth Flow. In *ISPRS Journal of Photogrammetry and Remote Sensing*, volume 194, pages 209–221, 2022.

[8] Erik Wijmans and Yasutaka Furukawa. Exploiting 2d floorplan for building-scale panorama rgbd alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 308–316, 2017.

[9] Anna Sokolova, Filipp Nikitin, Anna Vorontsova, and Anton Konushin. Floorplan-aware camera poses refinement. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4857–4864, 2022.

[10] Paul J Besl and Neil D McKay. Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[11] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[12] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European conference on computer vision*, pages 766–782. Springer, 2016.

[13] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Trans. Robotics*, 2020.

[14] Heng Yang and Luca Carlone. A polynomial-time solution for robust registration with extreme outlier rates. *arXiv preprint arXiv:1903.08588*, 2019.

[15] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ international conference on intelligent robots and systems*, pages 3384–3391. IEEE, 2008.

[16] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[17] Jun-Jee Chao, Selim Engin, Nicolai Häni, and Volkan Isler. Category-level global camera pose estimation with multi-hypothesis point cloud correspondences. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3800–3807. IEEE, 2023.

[18] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global context aware local features for robust 3D point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 195–205, 2018.

[19] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017.

[20] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: S3D point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5545–5554, 2019.

[21] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019.

[22] Dvir Ginzburg and Dan Raviv. Deep confidence guided distance for 3d partial shape registration. *arXiv preprint arXiv:2201.11379*, 2022.

[23] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[24] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. PointNetLK: Robust & efficient point cloud registration using PointNet. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[25] Yue Wang and Justin M Solomon. PRNet: Self-supervised learning for partial-to-partial registration. *Advances in neural information processing systems*, 32, 2019.

[26] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust point matching using learned features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[27] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. *Internaltional Conference on Computer Vision and Pattern Recogintion (CVPR)*, 2021.

[28] Haobo Jiang, Yaqi Shen, Jin Xie, Jun Li, Jianjun Qian, and Jian Yang. Sampling network guided cross-entropy method for unsupervised point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6128–6137, 2021.

[29] Bingli Wu, Jie Ma, Gaojie Chen, and Pei An. Feature interactive representation for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5530–5539, 2021.

[30] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *European Conference on Computer Vision (ECCV)*, 2020.

[31] Donghoon Lee, Onur C Hamsici, Steven Feng, Prachee Sharma, and Thorsten Gernoth. DeepPRO: Deep partial point cloud registration of objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5683–5692, 2021.

[32] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, and Konrad Schindler Andreas Wieser. PREDATOR: Registration of 3d point clouds with low overlap. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.

[33] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2514–2523, 2020.

[34] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11143–11152, June 2022.

[35] Xuancheng Zhang, Yutong Feng, Siqi Li, Changqing Zou, Hai Wan, Xibin Zhao, Yandong Guo, and Yue Gao. View-guided point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15890–15899, 2021.

[36] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PoinTr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, 2021.

[37] Ruihui Li, Xianzhi Li, Ke-Hei Hui, and Chi-Wing Fu. SP-GAN:sphere-guided 3d shape generation and manipulation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 40(4), 2021.

[38] Xingguang Yan, Liqiang Lin, Niloy J. Mitra, Dani Lischinski, Danny Cohen-Or, and Hui Huang. ShapeFormer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[39] Nicolai Häni, Jun-Jee Chao, and Volkan Isler. 3D surface reconstruction in the wild by deforming shape priors from synthetic data. *arXiv preprint arXiv:2302.12883*, 2023.

[40] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. AutoSDF: Shape priors for 3d completion, reconstruction and generation. In *CVPR*, 2022.

[41] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. VoxFormer: Sparse voxel transformer for camera-based 3d semantic scene

completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[42] Zhaoyang Xia, Youquan Liu, Xin Li, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, and Yu Qiao. SCPNet: Semantic scene completion on point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17642–17651, 2023.

[43] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017.

[44] Yingjie Cai, Xuesong Chen, Chao Zhang, Kwan-Yee Lin, Xiaogang Wang, and Hongsheng Li. Semantic scene completion via integrating instances and scene in-the-loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 324–333, 2021.

[45] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3991–4001, 2022.

[46] Huy Ha and Shuran Song. Semantic abstraction: Open-world 3D scene understanding from 2D vision-language models. In *Proceedings of the 2022 Conference on Robot Learning*, 2022.

[47] Fengyun Wang, Dong Zhang, Hanwang Zhang, Jinhui Tang, and Qianru Sun. Semantic scene completion with cleaner self. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–877, 2023.

[48] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models, 2023.

[49] Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. SceneScape: Text-driven consistent scene generation. *arXiv preprint arXiv:2302.01133*, 2023.

[50] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*, 2021.

[51] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5556–5565, 2015.

[52] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *Proceedings of the IEEE international conference on computer vision*, pages 143–152, 2017.

[53] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for D3D scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023.

[54] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[55] Matt Duckham, Lars Kulik, Mike Worboys, and Antony Galton. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern recognition*, 41(10):3224–3236, 2008.

[56] Adriano Moreira and Maribel Yasmina Santos. Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points. 2007.

[57] Jen-Hao Rick Chang, Wei-Yu Chen, Anurag Ranjan, Kwang Moo Yi, and Oncel Tuzel. Pointersect: Neural rendering with cloud-ray intersection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[58] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[59] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. IronDepth: Iterative refinement of single-view depth using surface normal and its uncertainty. In *British Machine Vision Conference (BMVC)*, 2022.

[60] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *ICCV*, 2017.

[61] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[62] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1):34–46, 2007.

[63] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 2432–2437. IEEE, 2005.