Pacific Graphics 2013
B. Levy, X. Tong, and KK. Yin
(Guest Editors)

Volume 32 (2013), Number 7

Interactive Learning for Point-Cloud Motion Segmentation

Yerry Sofer[1], Tal Hassner[1] and Andrei Sharf[2]

[1] Open University of Israel, Israel
[2] Ben-Gurion University, Israel

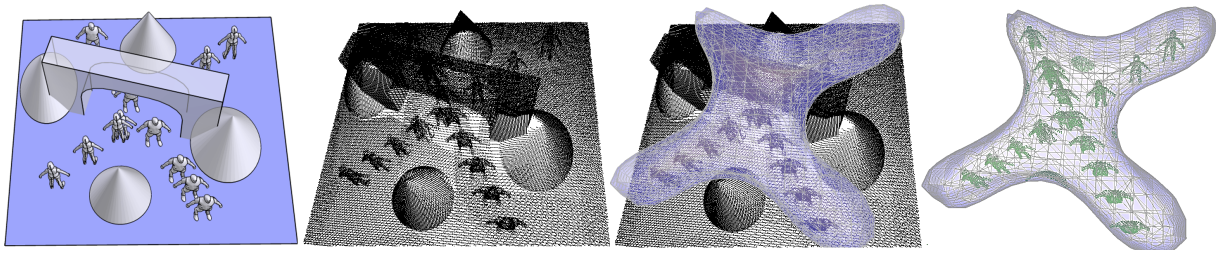

Figure 1: *Motion segmentation of dynamic point-cloud scenes. Given a dynamic scene of two people walking under a bridge with their paths intersecting (left), we simulate its scan (mid-left) and compute its fg/bg segmentation. We adaptively refine the decision border of our classifier (mid-right) resulting in an accurate motion (fg) separation (right).*

Abstract

Segmenting a moving foreground (fg) from its background (bg) is a fundamental step in many Machine Vision and Computer Graphics applications. Nevertheless, hardly any attempts have been made to tackle this problem in dynamic 3D scanned scenes. Scanned dynamic scenes are typically challenging due to noise and large missing parts. Here, we present a novel approach for motion segmentation in dynamic point-cloud scenes designed to cater to the unique properties of such data. Our key idea is to augment fg/bg classification with an active learning framework by refining the segmentation process in an adaptive manner.

Our method initially classifies the scene points as either fg or bg in an un-supervised *manner. This, by training discriminative RBF-SVM classifiers on automatically labeled, high-certainty fg/bg points. Next, we adaptively detect unreliable classification regions (i.e. where fg/bg separation is uncertain), locally add more training examples to better capture the motion in these areas, and re-train the classifiers to fine-tune the segmentation. This not only improves segmentation accuracy, but also allows our method to perform in a coarse-to-fine manner, thereby efficiently process high-density point-clouds. Additionally, we present a unique interactive paradigm for enhancing this learning process, by using a manual editing tool. The user explicitly edits the RBF-SVM decision borders in unreliable regions in order to refine and correct the classification. We provide extensive qualitative and quantitative experiments on both real (scanned) and synthetic dynamic scenes.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and scanning

1. Introduction

Recent advancements in scanning technologies, together with increasing computational power, allow real-time acquisition of 3D objects as they move and deform in time. Systems such as [ZSCS04, KvG06] produce dense point samples over large parts of the surface of a moving object, sampled at anywhere from ten to thirty frames per second. As these technologies mature, they make it possible to capture medium to large scale scenes containing multiple objects and their motions. Processing and modeling

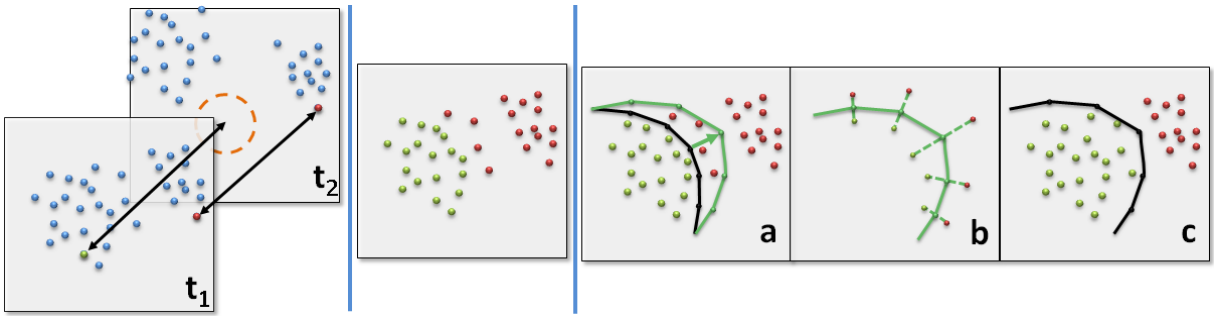

Figure 2: *Pipeline 2D illustration of our algorithm. Given a scanned dynamic scene, we compute our training set using closest distances between frame pairs (left), yielding a classification (mid). On the right, we compute the decision border (black line) and manually deform it in unreliable regions (a), compute a new training set (b) and recompute our classifier (c).*

the data captured by these scanners is a challenging computational problem which is only just beginning to be addressed [MFO*07, WJH*07, PG08].

An essential and particularly challenging problem within the field of motion processing is the problem of consistent foreground (fg) motion separation from the static background (bg). Such fg/bg segmentation tasks play key roles in many computer graphics and vision applications, including motion analysis, tracking, gesture recognition and animation. The particular problem of motion segmentation in videos has been extensively addressed [WC07, BEBV*10, BHH11]. This problem is difficult as typically there exists no clear dichotomy to extract the foreground, which is often highly entangled with its background. Additionally, the background is not perfectly static due to noise, camera movement and changes in illumination.

We transfer the motion segmentation problem to scanned 3D motion. Our problem domain is not only of higher dimensionality than videos; dynamic 3D point-clouds are complex and difficult to visualize and handle. Typically, scanned dynamic scenes are acquired by a small set of synchronized fixed cameras [PG08, LAGP09, LZW*09]. Due to the limited number of views, large parts of the surface are occluded, leading to gaping holes that often persist across many frames. In addition, illumination variations and scan noise generate flickering effects that cause inconsistent sampling in time. Finally, 3D scanned data is typically sparse and unstructured, lacking the underlying regular grid organization which exists in 2D video. Thus, understanding and modeling the static background in scanned dynamic scenes is a very challenging problem even if accurate high-end scanners are used (see Figure 1).

Segmentation problems are often solved using supervised classification techniques, provided that reliably labeled examples of fg and bg are available for training. Of course, manually labeling raw scans is infeasible, as they are complex, sparse, and lack visual accessibility. In lieu of such labels, we describe a novel classification system. It learns the scene partition into fg/bg in an unsupervised manner, yet provides natural means for adaptive and interactive segmentation refinement.

Our work makes the following contributions:

- **Definition of discriminative classifiers.** Our method sparsely samples the scene in space and time to produce an initial, automatically labeled training set consisting of points for which fg/bg labels can be determined with high-confidence. These are used to train RBF-SVM [CV95] classifiers which are then applied to label the entire scene.
- **Adaptive refinement of classifiers.** Sparse sampling, along with possible mis-labeling of the training points, can produce inaccurate classifiers. We describe an automatic, adaptive process which re-samples the training-set in a coarse to fine manner, refining the classification wherever the segmentation is determined to be unreliable.
- **Interactive learning by interactive decision-border editing.** We interpret the RBF-SVM classifier's decision boarder as a geometric surface in 3D. We describe a unique interface, allowing users to modify the segmentation by directly editing and manipulating this surface. We show that editing of the decision border can be used to re-train and refine the underlying classifier.

Our method is efficient, requiring a few minutes to process even large scenes. In order to test our system we apply it to both real (scanned) dynamic scenes, as well as synthetic scenes with known ground-truth labels. We provide qualitative and quantitative results demonstrating the accuracy of our system, its robustness to sampling noise, and its efficiency.

2. Related Work

Methods for motion segmentation have been extensively studied in both the computer vision and graphics communities. To our knowledge, ours is the first work to consider motion segmentation in dynamic, point-cloud scenes. Thus,

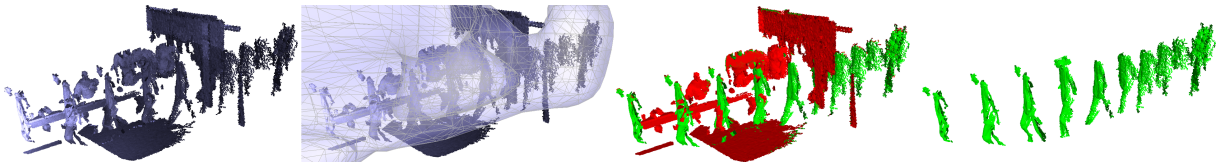

Figure 3: *Motion segmentation of a real scanned dynamic scene of people walking through a door. Left-to-right, the scanned input frames, the computed classifier yields a natural surface in 3D defining the decision border, resulting fg/bg segmentation in green and red respectively, and finally the extracted motion.*

we separately discuss motion segmentation in videos and 3D dynamic scan processing.

Motion segmentation in videos. There has been immense progress in foreground segmentation of videos since the original frame-difference techniques in the late 70's [JN79]. Generally speaking, existing approaches assume that the background can be modeled statistically and use these models to extract the foreground. These methods use global background models [LSS05, WBC*05, BS07, BM10, OB11] and pixel-wise models [Ziv04, GA11, CJC12] which require either a static camera or strong alignment of the video frames. Similar to us, interactive segmentation has been recently proposed by the SnapCut [BWSS09] and Livecut [PMC09] systems. Theirs, however, were designed for the underlying grid structure of images and videos. Recently hand-motion tracking was computed by clustering the scene flow into three predefined motion clusters using EM [HB12].

Few methods were developed to handle background motion induced by camera motion. These are often handled by recovering and compensating for the camera ego-motion [IRP94, RB96, MH00, HE03, RCH03] relying on motion coherency to segment point trajectories [FP98, Kan01, VM04, SJK09], and modeling the non-static backgrounds [MMPR03, MP04, ZSXW12]. Our work is related to the recent method of Han and Davis [HD12]; they used classifiers to model background noise and dynamic textures. Here we use a similar approach on dynamic 3D scans augmented within an adaptive and interactive framework.

All the methods mentioned above operate on regular grids, which allow fg/bg models to be assigned to specific, integer, spatial coordinates (i.e. pixels). In contrast, our method segments the moving points of a real-valued 3D scene, where the absence of an underlying regular grid raises the question of where in space-time these fg/bg models should be computed.

Active learning. Our system bears some resemblance to "active learning" techniques in machine learning. There, small subsets of the training data, for which classification is unreliable or unknown, are identified and deferred as queries to an oracle (typically, a human operator) for labeling. We refer to [Set09] for a survey on this topic. Despite its popularity, only few methods have recently employed active learning for image segmentation [VG09, SG10, VBF12, YWSC13], Nevertheless, their extension to our domain remains undetermined.

Similarly to the active learning approach, we refine classification by introducing additional training instances wherever segmented points are close to a decision border. This has long been an effective training of SVM classifiers (e.g., [SC00, TK02]). Here, we automatically re-sample the scene more densely, thereby introducing finer resolution into the training set near such decision borders. We further introduce the novel approach of interactive editing of the decision border. Rather than having users manually label new training points, as is typically performed in active learning systems, we describe a novel interface allowing users to directly manipulate the 3D surface of the decision border. As far as we know, we are the first to propose such interactive control over a classification system.

3D dynamic scans. In 3D scanned data, previous work has focused on segmentation of static scenes. Their emphasis is on extracting geometric primitives (such as in [UH03] and [RVDHV06]) using cues such as normals and curvature. Golovinski and Funkhouse [GF09] use graph-cut for segmenting *static* 3D scenes into foreground and background elements.

Much of the attention in 3D dynamic-scan processing has been focused on modeling of specific shapes such as humans and garments where subjects are acquired independently using controlled environments [ACP02, GKB03, ZSCS04, DSK*05, WCF07, VB*10, LXL*12]. In real-world conditions, however, it is practically impossible to capture a dynamic subject independently of its surrounding environment and background. The methods proposed here may therefore be considered as preprocessing, before applying these techniques.

Anuar and Guskov [AG04] and Sharf et al. [SAL*08] use flow techniques to estimate object deformation in the scene. Pekelny and Gotsman [PG08] introduce another method for tracking a piecewise rigid articulated model. In all these cases, however, a single foreground object is assumed, with little or no sampling noise; a scenario very different from

the one we handle here.

With the popularity of Kinect sensors, motion segmentation in dynamic range data has become an important problem. In [GBPG12], depth and intensity data are fused to generate dense trajectories for motion segmentation. The KinectFusion system of [IKH*11] uses a moving Kinect sensor and a GPU processor to reconstruct the 3D scene in real-time. They segment the static background from foreground motion by assuming that parts of the static scene have been initially reconstructed. Our method does not make such an assumption which is also infeasible in crowded environments.

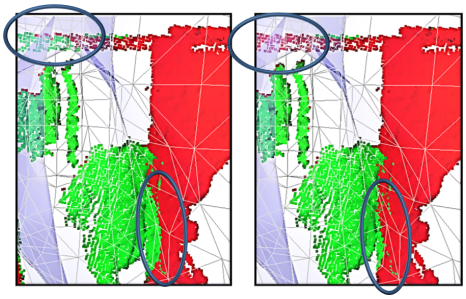

Figure 4: *Adaptive segmentation refinement in the door region of the scene from Fig. 3. The door is unreliable in terms of segmentation as subjects pass very close to it. Without adaptive refinement (left), top and side parts of the door are incorrectly labeled as fg (in green). After adaptive refinement (right), a majority of these labels are corrected.*

3. Overview

Our input data is a sequence of 3D raw scans sampling a dynamic scene over time. In particular, our data consists of 7D points $\mathbf{P} = \{\mathbf{p_i}\}_{i=1}^{N} = \{(x_i, y_i, z_i, u_i, v_i, w_i, t_i)\}_{i=1}^{N}$ where x_i, y_i, z_i are real 3D coordinates in the scene, $(u_i, v_i, w_i)^T$ is the measured normal of the surface at that spatial coordinate, and t is the integer frame index. We assume these points were acquired by one or more synchronized scanners positioned relatively still during the scanning process (e.g., steady hand-held). Our goal of separating foreground from background can be stated as designing a function h for labeling each point $\mathbf{p} \in \mathbf{P}$ as belonging to the moving foreground or the (semi-)static background. More formally we seek a function h, such that.

$$h(\mathbf{p}) = \begin{cases} \text{BG} & \text{if } \mathbf{p} \in \text{static background} \\ \text{FG} & \text{if } \mathbf{p} \in \text{moving foreground} \end{cases} \quad (1)$$

Our algorithm consists of the following major components (see Fig. 2 for a 2D illustration of our pipeline):

1. **Automatic, coarse segmentation.** A small subset of scene points is automatically labeled as belonging to either fg or bg by considering points in frame-pairs and assigning fg/bg labels based on their similarity (Fig. 2 (left)). To keep computation costs low, we compute labels for only a small number of points, randomly sampled in space and time. The labeled points are used to train a statistically robust, RBF-SVM classifier which is then applied to the whole scene.
2. **Adaptive segmentation refinement.** We augment the seed training set by adding more samples wherever scene points lie close to the fg/bg decision border. This expanded training set is used to recompute the classification model and re-segment the scene.
3. **Interactive decision border editing.** We provide an interface which visualizes the SVM decision border as a surface in 3D and allows its manual editing to correct segmentation errors (Fig. 2 (right)). Following this editing interaction, the modified surface is converted back to an RBF-SVM, and applied to the entire scene.

In the next section we provide details to these steps.

4. Technical details

Unsupervised motion segmentation. To compute motion segmentation, we seek a function h (Eq. 1) that partitions the whole 3D volume of the scene into two sub-volumes: a volume occupied by static objects, and a volume through which dynamic objects pass. Thus, applying h to classify a point $\mathbf{p}$, equals to determining in which of the two sub-volumes the 3D coordinates of $\mathbf{p}$ lie (see Figure 3).

With this in mind, we design a system which partitions the 3D scene into fg/bg using Gaussian, Radial Basis Function (RBF), Support Vector Machines (SVM) [BGV92, CHC*10]. We select the RBF kernel for its low computational costs in both training and testing and fewer numerical instabilities than other non-linear kernels. Moreover, the decision border produced by an RBF-Kernel SVM has a natural interpretation in 3D, which we employ in our system (Sec. 4). These RBF-SVM are trained on a subset of scene points, for which we compute label information.

Automatic training set labeling We begin by assigning a small number of sample points from throughout the scene (space and time) as either fg, bg, or unknown, using a conservative and efficient decision criteria. While this step is often manual in photos and videos, it is performed automatically here, as manual labeling of point-clouds would be unrealistic due to their complexity.

Labels are assigned by a modified frame-difference technique [JN79]. Specifically, given frame (point-set) F_{t1} at time $t1$, $F_{t1} = \{\mathbf{p_i} = (x_i, y_i, z_i, u_i, v_i, w_i, t_i) | t_i = t1\}$, and frame $F_{t2} = \{\mathbf{p_j} = (x_j, y_j, z_j, u_j, v_j, w_j, t_j) | t_j = t2\}$, at time $t2$, we assign the labels FG for foreground, BG, for background, and UN for unknown, by computing for each $\mathbf{p_i}$ ($\mathbf{p_j}$):

$$L(\mathbf{p_i}) = \begin{cases} \text{FG} & \text{, if } \forall \mathbf{p_j} \in F_{t2}, \text{dist}(\mathbf{p_i}, \mathbf{p_j}) > \theta_{FG} \\ \text{BG} & \text{, if } \exists \mathbf{p_j} \in F_{t2}, \text{dist}(\mathbf{p_i}, \mathbf{p_j}) < \theta_{BG} \\ \text{UN} & \text{, Otherwise} \end{cases} \quad (2)$$

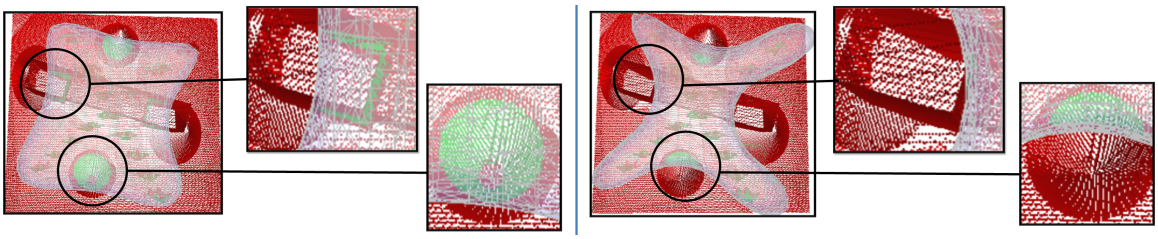

Figure 5: *Coarse sampling of our training set can lead to an incorrect initial segmentation model with many mis-classifications (left). Using a single adaptive step (right) reduces most of these errors (few mis-classifications still exist in the cone).*

Here, *dist* is a distance function designed to reflect how likely are the 7D points $\mathbf{p_i}$ and $\mathbf{p_j}$ to have been produced by scanning the same static object in two different frames. A point in F_{t1} is selected as fg if no point in F_{t2} is close enough to be considered from the same object (see Figure 2(left)). The two thresholds, θ_{FG} and θ_{BG} were manually set to reflect the scale of the scene, but were otherwise not optimized in our trials. We used the following values for these thresholds: $\theta_{FG} = 10cm$, and $\theta_{BG} = 5cm$ (in scanner coordinates).

Our implementation uses the following distance function:

$$\text{dist}(\mathbf{p_i},\mathbf{p_j}) = \begin{cases} ||(x_i,y_i,z_i),(x_j,y_j,z_j)||_2 & \text{, if } \mathbf{n_i}^T\mathbf{n_j} > 0.9 \\ \theta_{UN} & \text{, Otherwise} \end{cases}$$

Where $\mathbf{n_i}$ and $\mathbf{n_j}$ are vectors of the normal components, $(u,v,w)^T$, of the points $\mathbf{p_i}$ and $\mathbf{p_j}$, used here in order to consider only points which have nearly identical normal directions; points with greater normal differences are filtered out as being uncertain. In such cases, the constant value θ_{UN}, $\theta_{BG} < \theta_{FG} < \theta_{UN}$ causes the distance to be ignored in Equation 2. We note that consequent to Equation 2, different objects moving at different speeds (or the same object moving at different speeds in different times) would be labeled FG, so long as the motion is faster than implied by θ_{FG}.

A straightforward application of Equation 2 to all the points in the scene would be impractical due to its computational cost. Each point in the scene would require a near-neighbor search throughout all points in the matching frame. This process is additionally susceptible to errors. Very often, occlusions result in holes in the point-cloud, which cause bg points $\mathbf{p_i} \in F_{t1}$ to have no sufficiently near neighbors in F_{t2} (see Figure 4). Conversely, when foreground points in frame F_{t1} are near background objects, they will find sufficiently close background points in F_{t2}, despite the object having moved on by that time.

As a means of addressing these problems, we label only a small, randomly selected subset of points in $t1$. In addition, we sparsely sample the pairs of frames by selecting every k'th time-step as $t1$, and setting $t2 = t1 + r$. Thus, only a small number of frames are evaluated and only a limited number of high-confidence points are labeled as either FG or BG, and used to train the RBF-SVM classifiers. Consequently, training involves only modest computational costs. In our implementation, we sampled one-in-five frames as $t1$ ($k = 5$), with the gap from $t2$ being $r = 11$ frames. Finally, only 10% of the points in $t1$ were randomly sampled and used to compute the initial labels.

Adaptive segmentation refinement. Once training samples have been selected and labeled, we train an RBF-SVM classifier (using libSVM [CL11]) which is then applied to predict fg/bg labels for all the points in the scene. Specifically, for each point $\mathbf{p} = (x,y,z,u,v,w,t)^T$, and $\mathbf{x} = (x,y,z)^T$, we compute the following decision value:

$$c = \sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x_i},\mathbf{x}) + b. \tag{3}$$

Where K is the standard RBF-kernel, defined by $K(\mathbf{x_i},\mathbf{x_j}) = exp(-\gamma\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2), \gamma > 0$. The value for γ was automatically set by libSVM from a single real ($\gamma = 32$) / synthetic ($\gamma = 8$) example, then used in all subsequent real/synthetic tests respectively. l support vectors $\mathbf{x_i} = (x_i,y_i,z_i)^T$ were selected from the sample set used for training, y_i are their FG/BG labels, represented by 1/-1 values (Eq. 2), α_i, their weights, and b is a bias term, all computed during training. The initial fg/bg predicted labels are then determined by $\mathbf{sign}(c)$.

The cost of prediction (Eq. 3) is dominated by l, the number of support vectors selected during training. This, in turn, is bound by the number of training instances, and is therefore controlled by the sampling rate of frames and points in the initial labeling step, above. By aggressively down-sampling the points used for training, we can control prediction time, but at a cost to the accuracy (see Figure 5).

To maintain low computational costs, without sacrificing accuracy, we perform an adaptive refinement step, which automatically adds additional samples wherever the fg/bg segmentation is considered unreliable. Specifically, we consider points for which the decision value $|c| \leq \tau$ (Eq. 3), as lying close to the decision boundary. Here, τ is a threshold value which was kept constant throughout our experiments, its value manually set to $\tau = 5cm$ (in practice, point coordinates in Eq. 3, are normalized to the range of [-1,1] and the same normalization is applied to τ.) Generally speaking, these points were likely not have been included in the

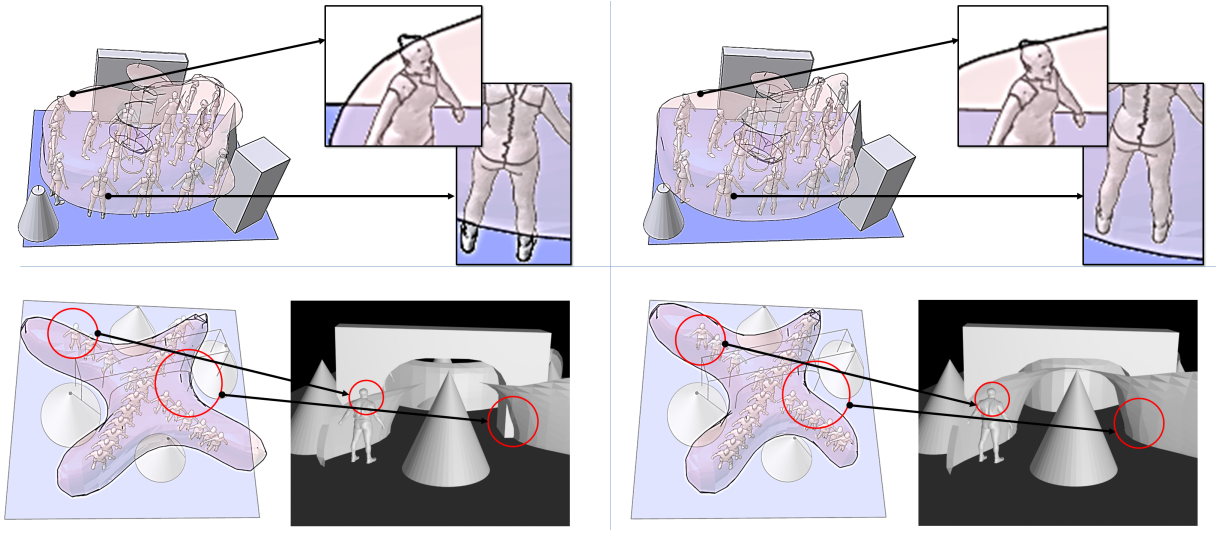

Figure 6: *Interactive editing of the decision boundary surface in two dynamic scenes (top/bottom). Left column, is the automatically computed separating surface, yielding incorrect labeling. Right column, the manually edited decision boundary yields a correct classification of the fg/bg regions.*

original training set. We now add them to the training by repeating, for each one, the same labeling process described in Eq. 2. The resulting, expanded, training set is then used to re-train the SVM classifiers. Figures 4, and 5 show the significant improvement of applying a single refinement step to an incorrect initial segmentation, resulting from over-aggressive sampling of points.

Interactive decision border editing. The process described so far is fully automatic, efficient, and as we show in Sec. 5, robust to changes of parameter values. Yet, as noted by others for related problems in images [BWSS09, PMC09], it is very often beneficial to provide means for human interaction and manual refinement. Here, we develop a novel editing framework to allow manipulation of the classification process (see Figure 6).

We treat the fg/bg labels computed by the automatic process, as the inside/outside labels for a Marching Cubes process applied to the 3D coordinates of our scene points. In our system, we used the Marching Cubes implementation of [Blo94]. This results in a 3D surface, represented as a triangulated mesh, which reflects the fg/bg decision boundary. Our system visualizes this surface, rendered transparently, along with the scene points, in order to allow the user to inspect the segmentation for any evident mis-classifications. Beyond visualization, however, we allow this surface to be manually manipulated. Specifically, a user can select and move the decision border surface, using local thin-plate splines deformations [DB02]. Thus, mis-classifications are corrected, by moving the surface to desirable locations where it better discriminates between fg/bg (Fig. 2(a)).

Following this manual surface adjustment, we revert back to an RBF-SVM classifier and re-apply the modified classifier to the points in the scene. To this end, We first produce a new training set of points, reflecting the user-modified segmentation (Fig. 2(b)). Let $\mathbf{q} = (x, y, z)$ be a vertex on the (modified) surface. We note that the position of $\mathbf{q}$ may or may not have been changed by the user. Regardless, $\mathbf{q}$ is *not* a part of the scanned scene, but instead a point originating from the Marching Cubes process. For each such $\mathbf{q}$, we produce two points $\mathbf{q_{out}}$ and $\mathbf{q_{in}}$, where $\mathbf{q_{out}}$ ($\mathbf{q_{in}}$) is a point at a distance ε away from $\mathbf{q}$, along the (opposite) direction of the surface normal at $\mathbf{q}$. The points $\mathbf{q_{out}}$ and $\mathbf{q_{in}}$ are then labeled as either FG or BG, according to the original normal direction assignments (e.g., in our implementation, normals point to the bg, and so $\mathbf{q_{out}}$ ($\mathbf{q_{in}}$) is labeled as BG (FG)). Once labeled, these points are used to train a new RBF-SVM model, which is then reapplied to the scene (Fig. 2(c)).

5. Results

We evaluate our system on multiple dynamic scenes with various complex motions. We have experimented with motions of one or more persons as well as various obstacles and non-trivial motion paths.

Description of our input. We have acquired raw scans sampling real dynamic scenes, using high resolution and frame-rate scanners. In our setup, scanners were hand-held, relatively still, or mounted on a tripod. We used either one or two synchronizes scanners. Our equipment is the Mantis Vision, Active Light Scanner, with a spec of up-to 100k points per frame and camera resolution of 640-480 pixels, at 45fps.

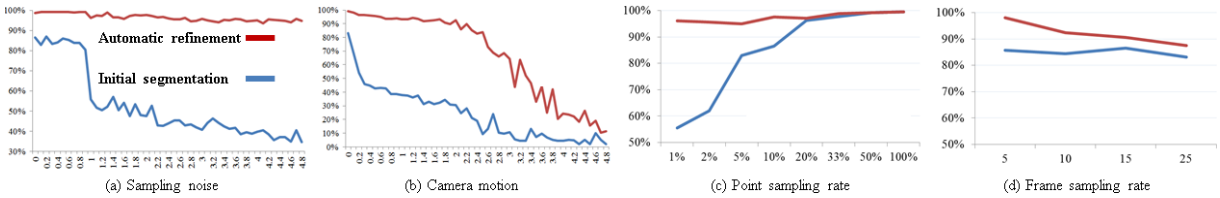

Figure 7: *Robustness of our system for initial (blue) and automatically refined (red) segmentations. Performance is measured as the percent of points correctly classified. We evaluate our method's robustness with increasing sampling noise (a), camera motion (b), varying point sampling rates (c) and temporal sampling rates (d).*

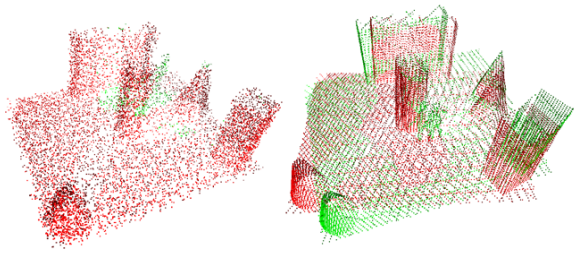

Figure 8: *Visualization of noise tests added to the scene in Fig. 9. Left, after 4% noise in points positions, right 4% camera motion (showing two subsequent frames in green and red).*

We additionally produced synthetic scenes with known ground-truth segmentation (Fig. 1, and 9). These scenes consist of complex motions, intersecting paths and various obstacles. We applied a virtual scanner which samples these scenes using a raycast-like method and generates 3D scans which simulate real-world scanning.

Summary of our results. Various properties of some of our scenes are summarized in Tab. 1 which also lists the run-times of the two automatic steps of our system (coarse segmentation and automatic refinement) and the accuracy of our segmentation of the synthetic scene. Accuracy is measured as the percent of scene points correctly classified as fg/bg. Run-times were measured on a standard Lenovo Think Pad, T420, with a 2.6 GHz CPU and 4GB of ram, running Win7.

Evident from Tab. 1 is that our process requires only few minutes to compute accurate segmentations, even for scenes containing millions of points. Also noteworthy is the significant boost in accuracy due to the refinement step. Additional tests on the synthetic scene of Fig. 9 were performed in order to assess our system's robustness. Fig. 7 shows accuracies before and after the automatic adaptive segmentation step (in the final interactive step, any errors can be manually corrected and so accuracy is considered perfect).

Fig. 7(a) provides an evaluation of our system's tolerance of simulated sampling noise. To this end, scene points were moved in random directions progressively further from their original positions, in order to simulate increasing amounts of sampling errors due to measurement noise. Thus, the noisier the samples, the further points will be from their ground-truth locations. Here, noise rates are measured as percents of the length of the diagonal of the scenes's bounding box. A 4% noise rate therefore implies substantial changes in the positions of the scene points (Fig. 8). Clearly, with increasing noise levels, the accuracy of the initial segmentation step degrades, but the segmentation refinement step corrects any such errors, and leaves the final accuracy mostly unaffected.

Fig. 7(b) shows our system's accuracy for varying amounts of camera motion (see Fig. 8). Motion was simulated by moving and rotating the whole scene in randomly selected directions and angles. Motion levels are again measured as in Fig. 7(a). Despite not being specifically designed to handle camera motion, following automatic refinement our method remains accurate in up to 2% camera motion.

In Fig. 7(c), we report accuracies for varying point sampling rates. These range from 100% (i.e., Eq. 2 applied to all points in the sampled frames) to the very aggressive rate of 1%. Clearly, as fewer points are sampled, the accuracy of the initial segmentation step drops. This, however, has little effect on the final accuracy, obtained after the segmentation refinement step. Finally, Fig. 7(d) demonstrates the effect of varying the temporal sampling rate (the distance k between consecutive frames $t1$). Lower rates have a moderate effect on the accuracy, both after the initial segmentation and the refinement step, though even when sampling one frame in 25, accuracy remains high.

Our results on both real and synthetic scenes are visualized in Fig. 1, 3, 5, 9, and 10.

6. Conclusions

In this paper we develop a novel approach to adaptive and interactive training of RBF-SVM classifiers, for the purpose of motion segmentation of dynamic, point-cloud scenes. To our knowledge, this is the first attempt to segment such scenes. Our algorithm defines robust, discriminative classification rules via *unsupervised* learning – without requiring manual labeling and supervision. We observe that our classifier has a natural, visual interpretations as a 3D surface, separating fg from bg. Building on this observation, we propose a unique interactive learning paradigm, allowing users to adjust clas-

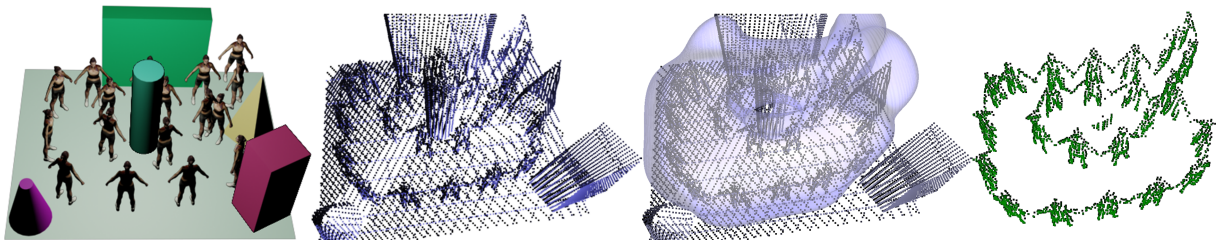

Figure 9: *Motion segmentation of a synthetic, dynamic scene where a person moves around obstacles in a circular path. Left-to-right, the synthetic, dynamic scene (frames overlaid), scanned point-cloud; automatically learned RBF-SVM motion segmentation yielding a spiral-like separation; and finally the accurately segmented fg motion.*

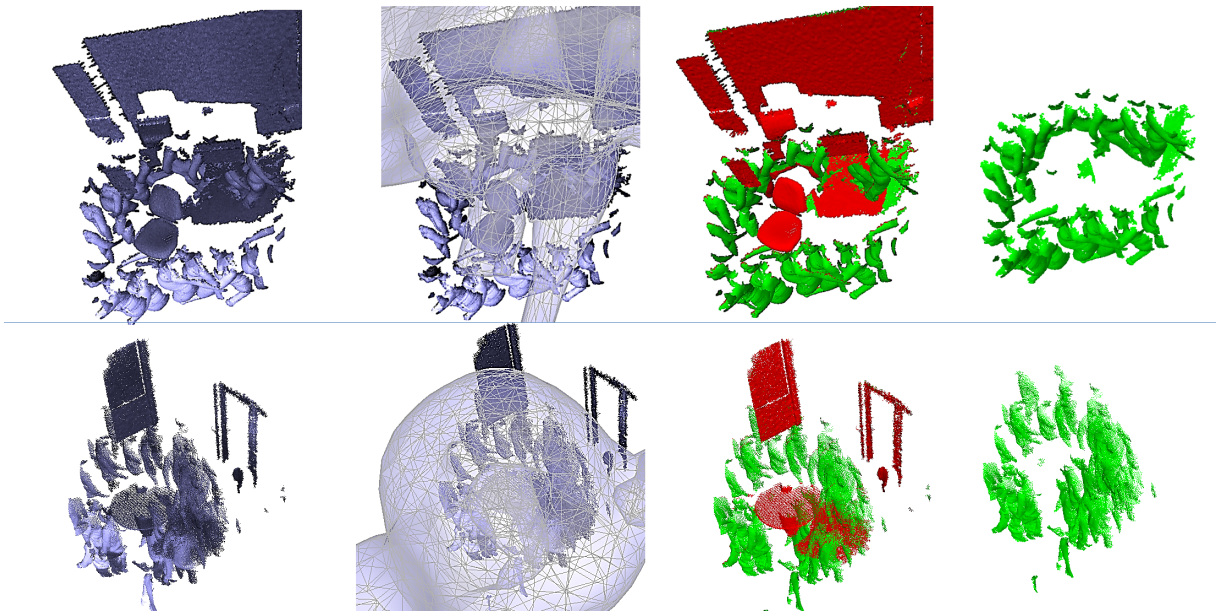

Figure 10: *Two motion segmentations in real, scanned dynamic scenes. A person walking around a chair (top) and table (bottom) in a room. Left-to-right are the dynamic point-cloud data; the automatically computed classifier yielding a separating surface in 3D; resulting fg/bg segmentation colored green and red respectively; and finally, the extracted fg motion.*

Table 1: *Details of scenes and results. R/S signifies Real (scanned) or Synthetic scene, ♯Frms, the number of time-frames, and ♯Pnts, the total number of points in the scene (all frames). Time1 is the runtime of the initial, unsupervised segmentation, in minutes, and % 1 is the percent of points it accurately classified. Time2 is the runtime of the automatic, adaptive step, and % 2 is the final accuracy. For the real, scanned scenes no ground truth is available and so accuracy is not reported.*

R/S	♯Frms	♯Pnts	Time1	% 1	Time2	% 2
S	50	490k	1.5	52.4%	2.9	96.1%
R	50	1,350k	1.8	n/a	2.5	n/a
R	265	9,275k	3.0	n/a	5.7	n/a

sification rules by manipulating these surfaces directly. We show how to use the edited surface to efficiently update the classifiers used for the segmentation. We evaluate the performance of our proposed system on both real, scanned point-cloud data, as well as synthetic scenes with ground-truth segmentation. Our tests show this process to be accurate, efficient and robust to the sampling noise, typical to such data, as well as insensitive to various parameter selections.

Discussion and future work. The work presented in this paper can be extended in two general directions. The first is by exploring ways of improving the performance and accuracy of our system. This, analogously to the evolving progress of motion segmentation techniques designed for videos. One particular issue which can be addressed is the propagation of information between points. Specifically, consider cases

where objects are only partly stationary. This can be, for example, a subject standing still while waving her hands. A different example is a large planer surface moving in a direction perpendicular to its normal. In both cases, the proximity of points can be exploited in order to propagate fg labels (e.g., from the tips of the hands or the rims of the surface) to points which are considered bg by Equation 2. Though such cases are partially handled by our use of SVM classifiers, as well as by manual label corrections, it would be interesting to seek a more principled treatment.

We are additionally interested in improving the system's performance, with the specific goal of employing it in real-time applications, such as surveillance. This would require improving the processing speed, as well as fine-tuning the systems accuracy in scenarios where manual interaction is not possible (e.g., batch processing).

A second avenue for future work is exploring how motion segmentation of point-cloud scenes can benefit subsequent processing of such data. Example applications are point-cloud scene compression and hole filling. By separating the fg from the bg and aggregating BG labeled points from multiple frame, a single bg frame can be produced, and stored only once, separately from frames containing the fg points. This can potentially save storage (a single instance of the bg, instead of one set of bg points for each frame), as well as provide means of filling in the holes in the bg that result from occlusions.

Finally, video understanding methods often rely upon an initial motion segmentation step, prior to subsequent processing [KGHW12]. Doing so for point-cloud data is also an intriguing next step. This includes the development of methods for action recognition, person identification (e.g., "gait recognition") and more.

Acknowledgements. We thank the reviewers and committee for their insightful remarks. Sharf was partially supported by the Israel Science Foundation (ISF) and the European IRG FP7.

References

[ACP02] ALLEN B., CURLESS B., POPOVIC Z.: Articulated body deformation from range scan data. In *Siggraph* (2002), ACM, pp. 612–619. 3

[AG04] ANUAR N., GUSKOV I.: Extracting animated meshes with adaptive motion estimation. In *Vision, Modeling, and Visualization* (2004), IOS, pp. 63–71. 3

[BEBV*10] BOUWMANS T., EL BAF F., VACHON B., ET AL.: Statistical background modeling for foreground detection: A survey. *Handbook of Pattern Recognition and Computer Vision* (2010), 181–199. 2

[BGV92] BOSER B. E., GUYON I. M., VAPNIK V. N.: A training algorithm for optimal margin classifiers. In *Proc. of the workshop on Computational learning theory* (1992), ACM, pp. 144–152. 4

[BHH11] BRUTZER S., HOFERLIN B., HEIDEMANN G.: Evaluation of background subtraction techniques for video surveillance. In *CVPR* (2011), IEEE, pp. 1937–1944. 2

[Blo94] BLOOMENTHAL J.: An implicit surface polygonizer. In *Graphics Gems IV* (1994), Academic Press, pp. 324–349. 6

[BM10] BROX T., MALIK J.: Object segmentation by long term analysis of point trajectories. *ECCV* (2010), 282–295. 3

[BS07] BAI X., SAPIRO G.: A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV* (2007), pp. 1–8. 3

[BWSS09] BAI X., WANG J., SIMONS D., SAPIRO G.: Video snapcut: robust video object cutout using localized classifiers. In *Proc. of SIGGRAPH* (2009), ACM, pp. 1–11. 3, 6

[CHC*10] CHANG Y.-W., HSIEH C.-J., CHANG K.-W., RINGGAARD M., LIN C.-J.: Training and testing low-degree polynomial data mappings via linear svm. *JMLR 99* (2010), 1471–1490. 4

[CJC12] CHANG H., JEONG H., CHOI J.: Active attentional sampling for speed-up of background subtraction. In *CVPR* (2012), IEEE, pp. 2088–2095. 3

[CL11] CHANG C.-C., LIN C.-J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2* (2011), 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 5

[CV95] CORTES C., VAPNIK V.: Support-vector networks. *Machine Learning 20*, 3 (1995), 273–297. 2

[DB02] DONATO G., BELONGIE S.: Approximate thin plate spline mappings. In *Proceedings of the 7th European Conference on Computer Vision-Part III* (2002), ECCV, pp. 21–31. 6

[DSK*05] DRAGOMIRANGUELOV, SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: SCAPE: shape completion and animation of people. *Transactions on Graphics 24*, 3 (2005), 408–416. 3

[FP98] FENG X., PERONA P.: Scene segmentation from 3D motion. In *CVPR* (1998), IEEE, pp. 225–231. 3

[GA11] GORUR P., AMRUTUR B.: Speeded up gaussian mixture model algorithm for background subtraction. In *AVSS* (2011), IEEE, pp. 386–391. 3

[GBPG12] GHUFFAR S., BROSCH N., PFEIFER N., GELAUTZ M.: Motion segmentation in videos from time of flight cameras. In *Systems, Signals and Image Processing (IWSSIP)* (2012), IEEE, pp. 328–332. 4

[GF09] GOLOVINSKIY A., FUNKHOUSER T.: Consistent segmentation of 3d models. *Computers & Graphics 33*, 3 (2009), 262–269. 3

[GKB03] GUSKOV I., KLIBANOV S., BRYANT B.: Trackable surfaces. In *Symposium on Computer Animation* (2003), ACM/Eurographics, pp. 251–257. 3

[HB12] HADFIELD S., BOWDEN R.: Go with the flow: hand trajectories in 3d via clustered scene flow. In *Image Analysis and Recognition*. Springer, 2012, pp. 285–295. 3

[HD12] HAN B., DAVIS L.: Density-based multifeature background subtraction with support vector machine. *TPAMI 34*, 5 (2012), 1017–1023. 3

[HE03] HAYMAN E., EKLUNDH J.: Statistical background subtraction for a mobile observer. In *ICCV* (2003), IEEE, p. 67. 3

[IKH*11] IZADI S., KIM D., HILLIGES O., MOLYNEAUX D., NEWCOMBE R., KOHLI P., SHOTTON J., HODGES S., FREEMAN D., DAVISON A., FITZGIBBON A.: KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *ACM symposium on user interface software and technology* (2011), pp. 559–568. 4

[IRP94] IRANI M., ROUSSO B., PELEG S.: Computing occluding and transparent motions. *IJCV 12*, 1 (1994), 5–16. 3

[JN79] JAIN R., NAGEL H.: On the analysis of accumulative difference pictures from image sequences of real world scenes. *TPAMI*, 2 (1979), 206–214. 3, 4

[Kan01] KANATANI K.: Motion segmentation by subspace separation and model selection. In *ICCV* (2001), vol. 2, IEEE, pp. 586–591. 3

[KGHW12] KLIPER-GROSS O., HASSNER T., WOLF L.: The action similarity labeling challenge. *TPAMI 34*, 3 (2012). 9

[KvG06] KONINCKX T. P., VAN GOOL L. J.: Real-time range acquisition by adaptive structured light. *Transactions on Pattern Analysis and Machine Intelligence 28*, 3 (2006), 432–445. 1

[LAGP09] LI H., ADAMS B., GUIBAS L. J., PAULY M.: Robust single-view geometry and motion reconstruction. *ACM Trans. on Graphics 28*, 5 (2009). 2

[LSS05] LI Y., SUN J., SHUM H.: Video object cut and paste. *ACM TOG 24*, 3 (2005), 595–600. 3

[LXL*12] LIAO B., XIAO C., LIU M., DONG Z., PENG Q.: Fast hierarchical animated object decomposition using approximately invariant signature. *TVC 28*, 4 (2012), 387–399. 3

[LZW*09] LIAO M., ZHANG Q., WANG H., YANG R., GONG M.: Modeling deformable objects from a single depth camera. In *Proc. Int. Conf. on Comp. Vis.* (2009). 2

[MFO*07] MITRA N. J., FLÖRY S., OVSJANIKOV M., GELFAND N., GUIBAS L., POTTMANN H.: Dynamic geometry registration. In *Proc. Eurographics Symp. on Geometry Processing* (2007), pp. 173–182. 2

[MH00] MITTAL A., HUTTENLOCHER D.: Scene modeling for wide area surveillance and image synthesis. In *CVPR* (2000), IEEE, p. 2160. 3

[MMPR03] MONNET A., MITTAL A., PARAGIOS N., RAMESH V.: Background modeling and subtraction of dynamic scenes. In *ICCV* (2003), IEEE, pp. 1305–1312. 3

[MP04] MITTAL A., PARAGIOS N.: Motion-based background subtraction using adaptive kernel density estimation. In *CVPR* (2004), IEEE, pp. 302–309. 3

[OB11] OCHS P., BROX T.: Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV* (2011), IEEE, pp. 1583–1590. 3

[PG08] PEKELNY Y., GOTSMAN C.: Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum 27*, 2 (2008), 399–408. 2, 3

[PMC09] PRICE B., MORSE B., COHEN S.: Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV* (2009), IEEE, pp. 779–786. 3, 6

[RB96] ROWE S., BLAKE A.: Statistical mosaics for tracking. *IVC 14*, 8 (1996), 549–564. 3

[RCH03] REN Y., CHUA C., HO Y.: Statistical background modeling for non-stationary camera. *Pattern Recognition Letters 24*, 1-3 (2003), 183–196. 3

[RVDHV06] RABBANI T., VAN DEN HEUVEL F., VOSSELMANN G.: Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences 36*, 5 (2006), 1–6. 3

[SAL*08] SHARF A., ALCANTARA D. A., LEWINER T., GREIF C., SHEFFER A., AMENTA N., COHEN-OR D.: Space-time surface reconstruction using incompressible flow. *ACM Trans. on Graphics 27*, 5 (2008), 1–10. 3

[SC00] SCHOHN G., COHN D.: Less is more: Active learning with support vector machines. In *ICML* (2000), Citeseer. 3

[Set09] SETTLES B.: *Active Learning Literature Survey*. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. 3

[SG10] SIDDIQUIE B., GUPTA A.: Beyond active noun tagging: Modeling contextual interactions for multi-class active learning. In *CVPR* (2010), IEEE, pp. 2979–2986. 3

[SJK09] SHEIKH Y., JAVED O., KANADE T.: Background subtraction for freely moving cameras. In *ICCV* (2009), IEEE, pp. 1219–1225. 3

[TK02] TONG S., KOLLER D.: Support vector machine active learning with applications to text classification. *JMLR 2* (2002), 45–66. 3

[UH03] UNNIKRISHNAN R., HEBERT M.: Robust extraction of multiple structures from non-uniformly sampled data. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)* (October 2003), vol. 2, pp. 1322–29. 3

[VB*10] VARANASI K., BOYER E., ET AL.: Temporally coherent segmentation of 3d reconstructions. In *Int. Symp. on 3D Data Proc., Visualization and Transmission* (2010). 3

[VBF12] VEZHNEVETS A., BUHMANN J. M., FERRARI V.: Active learning for semantic segmentation with expected change. In *CVPR* (2012), IEEE, pp. 3162–3169. 3

[VG09] VIJAYANARASIMHAN S., GRAUMAN K.: What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR* (2009), IEEE, pp. 2262–2269. 3

[VM04] VIDAL R., MA Y.: A unified algebraic approach to 2-D and 3-D motion segmentation. In *ECCV* (2004), Citeseer. 3

[WBC*05] WANG J., BHAT P., COLBURN R., AGRAWALA M., COHEN M.: Interactive video cutout. In *Proc. of SIGGRAPH* (2005), ACM, pp. 585–594. 3

[WC07] WANG J., COHEN M.: Image and video matting: a survey. *Foundations and Trends® in Computer Graphics and Vision 3*, 2 (2007), 97–175. 2

[WCF07] WHITE R., CRANE K., FORSYTH D. A.: Capturing and animating occluded cloth. In *Siggraph* (2007), ACM, p. 34. 3

[WJH*07] WAND M., JENKE P., HUANG Q., BOKELOH M., GUIBAS L., SCHILLING A.: Reconstruction of deforming geometry from time-varying point clouds. In *Proc. Eurographics Symp. on Geometry Processing* (2007), pp. 49–58. 2

[YWSC13] YAN C., WANG D., SHAN S., CHEN X.: Interactive segmentation with recommendation of most informative regions. In *Intelligent Science and Intelligent Data Engineering*. Springer, 2013. 3

[Ziv04] ZIVKOVIC Z.: Improved adaptive gaussian mixture model for background subtraction. In *ICPR* (2004), vol. 2, pp. 28–31. 3

[ZSCS04] ZHANG L., SNAVELY N., CURLESS B., SEITZ S. M.: Spacetime faces: high resolution capture for modeling and animation. *Transactions on Graphics 23*, 3 (2004), 548–558. 1, 3

[ZSXW12] ZHU Q., SONG Z., XIE Y., WANG L.: A novel recursive bayesian learning-based method for the efficient and accurate segmentation of video with dynamic background. *TIP 21*, 9 (2012), 3865–3876. 3