Towards Spatio-Temporally Consistent Semantic Mapping

Zhe Zhao^(✉) and Xiaoping Chen

University of Science and Technology of China, Hefei, China
zhaozhe@mail.ustc.edu.cn, xpchen@ustc.edu.cn

Abstract. Intelligent robots require a semantic map of the surroundings for applications such as navigation and object localization. In order to generate the semantic map, previous works mainly focus on the semantic segmentations on the single RGB-D images and fuse the results by a simple majority vote. However, single image based semantic segmentation algorithms are prone to producing inconsistent segments. Little attentions are paid to the consistency over the semantic map. We present a spatio-temporally consistent semantic mapping approach which can generate the temporal consistent segmentations and enforce the spatial consistency by Dense CRF model. We compare our temporal consistent segment algorithm with the state-of-art approach and generate our semantic map on the NYU v2 dataset.

Keywords: Spatio-temporal consistency · Semantic mapping · Indoor scene understanding

1 Introduction

Semantic understanding of the environment from images plays an important role in robotic tasks, such as human-robot interaction, object manipulation and autonomous navigation. The release of Microsoft Kinect, and the wide availability of RGB-D sensors, changes the landscape of indoor scene analysis. Many works have investigated single RGB-D image-based indoor scene understanding using different techniques such as SVM [6], CRF [13], CPMC [2]. Although, many advancements have been reported on the indoor image dataset NYU, single image based semantic segmentation algorithms are prone to producing inconsistent segments, such as Fig. 1. Many reasons cause this inconsistent segments including the motion of the camera, lighting, occlusion and even the sensor noises. These inconsistent predictions can have a significant impact on robotic tasks in practice. For example, obstacles may suddenly appear in front of the robot in one frame and vanish in the next.

In order to constrain the semantic segmentations in images to be temporally consistent, we need to find the correspondences between frames. For RGB images, optical flow algorithm is often used: for any pixel $z_{n'}^{t+1}$, find the optical flow m_t from z_n^t to $z_{n'}^{t+1}$ ($n' = n + m_t$), and let $z_{n'}^{t+1}$ takes the same label as z_n^t .

Fig. 1. An example of the inconsistent semantic labels of the indoor scene. Independent segmentations are prone to producing inconsistent superpixels and further resulting in inconsistent semantic precisions. (These images are best seen in color) (Colour figure online).

In this case, the temporal consistent constraints can be easily formulated as a binary relation between one pixel in frame t and one in frame $t+1$, in the form of $f(z_{n'}^{t+1}, z_n^t)$. However, the biggest problem of the exact pixel motion assumption is the optical flow algorithms are often not as reliable as we expect it to be. For example, holes and dragging effects often occur as the results of inaccurate optical flows [3]. In order to improve the performance, complex appearance models or motion-appearance combined approaches [1, 4] are investigated. However, depth information and the RGB-D images are less being researched to improve the performance.

In this paper, we want to build a spatio-temporally consistent semantic map for indoor scenes. Typically, incrementally building a semantic map for RGB-D images involve two steps: (1) generate the semantic segmentation for RGB-D images. (2) fuse the semantic segmentation using the estimated camera poses and get the global semantic map. However, in step 1, simply applying the semantic segmentation algorithms to each image is not sufficient as it does not enforce temporal consistent. Different from previous works that transfer the label through 2D optical flow or complex appearance models, we build the correspondence with the help of depth information. As we observed, the depth information can help build a more easily and more accurate consistent segmentation algorithm to constrain the temporal consistency. In order to explain its effective, we compare our segmentation algorithms with the state-of-art consistent segmentation algorithm in experiments. In step 2, since many points will be fused over time, simply taking the last label as the 3D point label is suboptimal. A straightforward approach is to keep track of the different pixels, that are fused in a

3D point. Then the label is selected by a simple majority vote. Although, this visually improves the results, however it will not lead to a spatial consistency. For this purpose, we use a Dense CRF (Conditional Random Field) model to distribute the information over neighboring points depending on the distance and color similarity. In summary, our algorithm consists of two main contributions:

- We use depth information to help building the temporal consistent segmentations over RGB-D images and probability transfer the semantic label according to the correspondence. In the experiment, we compare our consistent segmentations with the state-of-art algorithm and show that our method is more effective and accurate.
- We use a Dense CRF to enforce the spatial consistency and generate the spatio-temporally consistent semantic maps for NYU v2 dataset. The detailed results will be reported in experiment.

In the rest of the paper, we first briefly describe the related work in Sect. 2. In Sect. 3, we introduce the details of the temporal consistent segmentation algorithm. Section 4 discusses the spatial consistency method. Experimental results and statistics are described in Sect. 5 and the conclusion with discussions about future work in Sect. 6.

2 Related Work

Robots need a semantic map of the surrounding environment for many applications such as navigation and object localization. In RoboCup@Home, a set of benchmark tests is used to evaluate the robots abilities and performance in a realistic non-standardized home environment setting. The ability of building a consistent semantic map is useful to some tests, such as the Clean UP, General Purpose and Restaurant. For example, in the Restaurant test, a robot needs not to be guided by a human if it can build a semantic map by itself. However, the previous works mainly focus on the single RGB-D image based indoor scene understanding [2, 6, 11, 13, 14] which will cause inconsistent result for image sequences. Reference [8] labels the indoor scene over the global point cloud, but it is unclear how long the feature computation takes. For robots, it is typically suitable to build the semantic map incrementally, rather than doing all of the work on the global 3D scene. In this paper, we build a spatio-temporally consistent semantic map by the temporal consistent segmentation and Dense CRF.

3 Temporal Consistent Segmentation

Independently segment images into superpixels is prone to producing inconsistent segments. Given a segmentation S_t of an RGB-D image I_t at time t , we wish to compute a segmentation S_{t+1} of the image I_{t+1} at the time $t + 1$ which is consistent with the segments S_t . An example of the independent segmentations and temporal consistent segmentations are shown in Fig. 2. Compare to independent segmentations, the temporal consistent segmentations can capture the

Fig. 2. A comparison of the independent segmentations and temporal consistent segmentations. (The images are best seen in color) (Colour figure online).

correspondence for superpixels between frames. Each corresponding superpixel presences in every frame and will not vanish in one frame and suddenly appear in the next. These temporal consistent segmentations will lead to more consistent semantic precisions than independent segmentations.

3.1 3D Transformation of the Camera

In the 2D image, researchers often use optical flow to find the motion of pixels. However, the accuracy of the optical flow can not be guaranteed. For RGB-D images, if the scene is static which means only the camera moves and objects in the scene keep static, the 3D transformation can be computed effectively with the help of the depth information. If the 3D transformation T is known, we back-project the image I_{t+1} into image I_t . In this way, the binary relation between one pixel in frame t and one in frame $t + 1$ is founded. These binary relations are more accurate than optical flow method as they use the 3D information.

In order to compute the 3D transformation of the camera, SURF and corner points are employed. We detect Shi-Tomasi [12] corner points and adopt the Lucas-Kanade optimal flow method to track the corner points between frames. For SURF keypoints, a KD-tree based feature matching approach is used to find the correspondence. Then RANSAC (Random Sample Consensus) procedure is running to find a subset of feature pairs corresponding to a consistent rigid transformation. At last, a rigid transformation is estimated by SVD from the subset of feature pairs. If the RANSAC algorithm fails to find enough inliers, an ICP algorithm is used which is initialized by the transformation between the previous two frames.

3.2 Graph Matching

Before generating the consistent segmentations S_{t+1} for image I_{t+1} , we first independently segment the image I_{t+1} into superpixels S'_{t+1} using the 2D segmentation algorithm [5]. Once the image I_{t+1} is independently segmented, we

need to find the correspondence between segments S_t and S'_{t+1} . We use a graph matching procedure to find the correspondence. The vertices of G include two sets of vertices: V_t which are the set of regions of S_t and V'_{t+1} that are the set of regions of S'_{t+1} . We back-project image I_{t+1} into the image I_t by the 3D transformation and make edges if there exist an overlap between the segment v_i and the back-projected segment v'_j . The edge weights are defined by the size of overlap region:

$$w_{ij} = |r_i \cap r_j| \quad (1)$$

where r_i is the number of pixels of region r_i , $|r_i \cap r_j|$ is the number of pixels in the overlap region.

We find the correspondence using the forward flow and reverse flow together. That means for each segment of S'_{t+1} , we find its best corresponding segment in S_t by maximizing the w_{ij} . Symmetrically, for each segment of S_t , its best corresponding segment in S'_{t+1} is identified. These two corresponding sets in forward flow and reverse flow can be used for generating the final consistent segmentation S_{t+1} .

3.3 Final Segmentation

According to the correspondences in the forward flow and reverse flow, we define that segment $f(s)$ in S'_{t+1} is the best correspondence of the segment s in S_t , the segment $r(s')$ in S_t is the best correspondence of the segment s' in S'_{t+1} . Four cases may appear:

- $f(r(s')) = s' \wedge r(f(s)) = s$, that means s' should have the same segment label as $r(s')$ in S_t .
- $r(s') = NULL$ or $f(r(s')) \neq s'$, that means s' has no matching region or s' is a part of a larger segment $r(s')$ in S_t . In this case, s' need have a new segment label.
- $r(f(s)) \neq s$, that means the segment s in S_t disappears in S'_{t+1} . In this case, the segment s should be propagated in S'_{t+1} by the 3D transformation and keep its segment label.

3.4 Semantic Transfer

To classify the superpixels, we compute the geometry and appearance features as in [6] and train SVM (Support Vector Machine) classifiers based on them. The scores obtained by the classifiers are used as the semantic prediction for superpixels. In order to obtain temporal consistent semantic predictions over images, we transfer the semantic predictions based on the consistent segmentations.

For each segment s in S_{t+1} , if the segment label of s comes from the previous segmentation S_t , we transfer the semantic prediction by the following equation:

$$y_i^{t+1} = y_i^{t+1} + cy_j^t \quad (2)$$

where the semantic prediction y_j^t of s_j^t in S_t is transferred into the segment s_i^{t+1} in S_{t+1} , c is the weight that controls the transfer ratio.

4 Build the Semantic Map

We generate and voxelize the 3D global point cloud using the estimated 3D transformation. In order to fuse the temporal consistent semantic segmentation results into the global point cloud, a straightforward approach is a simple majority vote from the different pixels that are fused in the same 3D point. However, this will not lead to a spatial consistent result. For this purpose, we want to use the CRF to model the local interactions between neighbor 3D points. However, for 3D point cloud, the computation cost for finding point neighbors is expensive. So, we adopt the full connected CRF model in this work. This means the graph with node set V is defined by $G = \{V, V \times V\}$, each node can directly influence every other node by a pairwise potential. If we consider a fully connected CRF over the 3D global point cloud, this will result in billions of edges in a graph. Here the traditional inference algorithms are no longer feasible. In order to render this feasible, [9] provides a Dense CRF that limits the pairwise potentials to a linear combination of Gaussian kernels. This approach can produce good labeling performance within 0.2-0.7 s in our experiments.

Formally, we denote object category $y_i \in \{1, \dots, C\}$ where C is the number of the object category. X is the global point cloud and the Dense CRF graph $G = \{V, V \times V\}$ contains vertices v_i corresponding to the random variables y_i . The energy of the Dense CRF model is defined as the sum of unary and pairwise potentials:

$$E(Y|X) = \sum_{i \in V} \psi_u(y_i|X) + \sum_{(i,j) \in V \times V} \psi_p(y_i, y_j|X). \quad (3)$$

For the unary potential $\psi_u(y_i|X)$, we multiply the likelihoods of pixels which are fused in the same 3D point and normalize it using the geometric mean:

$$\psi_u(y_i|X) = -\log(\{\prod_{p_i \in v_i} p(y_i|p_i)\}^{\frac{1}{N}}) \quad (4)$$

where p_i is the image pixel that is fused in the 3D point v_i , N is the number of pixels that are fused in the 3D point v_i , the probability of $p(y_i|p_i)$ is obtained through SVM classifiers.

For the pairwise potential $\psi_p(y_i, y_j|X)$, we compute it by a linear combination of Gaussian kernels:

$$\psi_p(y_i, y_j|X) = \mu(y_i, y_j) \sum_{m=1}^K w^{(m)} k^{(m)}(f_i, f_j) \quad (5)$$

where $\mu(y_i, y_j)$ is a label compatibility function, the vector f_i, f_j are feature vectors and $k^{(m)}$ are Gaussian kernel $k^{(m)}(f_i, f_j) = \exp(-\frac{1}{2}(f_i - f_j)^T \Lambda^{(m)}(f_i - f_j))$. Each kernel $k^{(m)}$ is characterized by a symmetric, positive-definite precision matrix $\Lambda^{(m)}$, which defines its shape.

We define two Gaussian kernels for the pairwise potential: appearance kernel and smoothness kernel. The appearance kernel is computed as follow:

$$w^{(1)} \exp(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}) \quad (6)$$

where p is the 3D location of the point and I is the color vector. This kernel is used to model the interactions between points with a similar appearance. The smooth kernel is defined as follow:

$$w^{(2)} \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right) \quad (7)$$

This kernel is used to removes small isolated regions.

Instead of computing the exact distribution $P(Y)$, the mean field approximation method is used to compute a distribution $Q(Y)$ that minimizes the KL-divergence $D(Q||P)$ among all distributions Q that can be expressed as a product of independent marginal, $Q(Y) = \prod_i Q_i(Y_i)$. The final point label is obtained by $y_i = \operatorname{argmax}_l Q_i(l)$.

5 Experiment

5.1 Dataset

We test the system on the NYU v2 RGB-D dataset. The dataset is recorded using Microsoft Kinect cameras at 640*480 RGB and depth image resolutions. The NYU dataset contains 464 RGB-D scenes taken from 3 cities and 407024 unlabeled frames. It comes with 1449 images with ground-truth labeling of object classes. And it has been split into 795 training images and 654 test images. Furthermore, the dataset contains a large number of raw images, which we use for our semantic mapping. We select 4 RGB-D scenes to generate their semantic map and train 12 core classes on the 795 training images. Firstly, we evaluate our temporal consistent segmentation algorithm with the state-of-art approach StreamGBH for the 4 RGB-D scenes. Secondly, we evaluate our semantic mapping method on the 4 RGB-D scenes.

5.2 Evaluate the Temporal Consistent Segmentation

We compare our consistent segmentation algorithm with the state-of-art approach StreamGBH using the video segmentation performance metrics from [15]. The StreamGBH is a hierarchical video segmentation framework which is available as part of LIBSVX software library, which can be download at <http://www.cse.buffalo.edu/~jcorso/r/supervoxels/>. The 4 RGB-D scenes are densely labeled with semantic category by ourselves. We evaluate consistent segmentation using the undersegmentation error, boundary recall, segmentation accuracy and the explained variation. Instead of evaluating single image metrics, these evaluation metrics are used for 3D supervoxel in the 3D space-time which is mentioned in [15].

2D and 3D Undersegmentation Error: It measures what fraction of voxels goes beyond the volume boundary of the ground-truth segment when mapping the segmentation onto it.

$$UE(g_i) = \frac{\{\sum_{\{s_j | s_j \cap g_i \neq \emptyset\}} Vol(s_j)\} - Vol(g_i)}{Vol(g_i)} \quad (8)$$

where g_i is the ground-truth segment, Vol is the segment volume. 2D undersegmentation error is defined in the same way.

2D and 3D Segmentation Accuracy: It measures what fraction of a ground-truth segmentation is correctly classified by the supervoxel in 3D space-time. For the ground-truth segment g_i , a binary label is assigned to each supervoxel s_j according to the majority part of s_j that resides inside or outside of g_i .

$$ACCU(g_i) = \frac{\sum_{j=1}^k Vol(\bar{s}_j \cap g_i)}{Vol(g_i)} \quad (9)$$

where $\bar{s}_j$ is the set of correctly labeled supervoxels.

2D and 3D Boundary Recall: It measures the spatio-temporal boundary detection. For each segment in the ground-truth and supervoxel segmentations, we extract the within-frame and between-frame boundaries and measure recall using the standard formula as in [15].

Explained Variation: It considers the segmentation as a compression method of a video which is proposed in [10]:

$$R^2 = \frac{\sum_i (\mu_i - \mu)^2}{\sum_i (x_i - \mu)^2} \quad (10)$$

where x_i is the actual voxel value, μ is the global voxel mean and μ_i is the mean value of the voxels assigned to the segment that contains x_i .

The detailed comparison is shown in Table 1. Our approach is obviously better than the StreamGBH method. Sometimes StreamGBH’s performance is not ideal, for example the 3D segmentation accuracy is only 0.3942 for scene 1. However the 3D accuracy in the well-known xiph.org videos for the StreamGBH is about 0.77. The main reasons of the bad performance for StreamGBH are twofold: (1) In the 2D videos, the difference or motion between neighboring frames is small. However, in the semantic mapping procedure, the neighboring frames are keyframes [7] which have a larger difference and motion. StreamGBH can not handle the large motion for neighboring frames. (2) StreamGBH only uses the color cue which is difficult to find the exact correspondence. 3D motion is more helpful and accurate for semantic mapping procedure which only camera moves and objects remain static. The visualizations for the StreamGBH and our algorithm are shown in Fig. 3.

The limit of our method is that we assume the objects in the scene remain static. If the objects move, we can not compute 3D transformation for neighboring frames, the global 3D point cloud can not be obtained, either.

5.3 Evaluate the Spatio-Temporally Consistent Semantic Map

For semantic labeling, we train 12 core classes on the 795 training images. We compute the geometry and appearance features using the algorithm in [6] and

Table 1. The comparison between StreamGBH and our approach.

Method	2D UE	3D UE	2D accuracy	3D accuracy	2D boundary	3D boundary	EV
Scene 1							
StreamGBH	12.5185	15.695	0.4214	0.3942	0.0993	0.5365	0.2823
Our	4.3192	7.339	0.8292	0.7793	0.2	0.9043	0.7835
Scene 2							
StreamGBH	4.5693	6.7917	0.652	0.5955	0.1705	0.8163	0.7166
Our	2.1626	2.41	0.8332	0.8109	0.2531	0.9411	0.8581
Scene 3							
StreamGBH	7.8218	17.068	0.5722	0.3454	0.1427	0.5873	0.4433
Our	1.6341	3.7838	0.8862	0.7072	0.2653	0.9606	0.7647
Scene 4							
StreamGBH	3.3975	10.6291	0.8161	0.7657	0.2047	0.8493	0.7875
Our	1.6685	4.6865	0.8627	0.7706	0.3149	0.954	0.8326

StreamGBH

Our temporal consistent segmentation

Fig. 3. Some examples of the StreamGBH and our temporal consistent segmentation. From the examples, we can see that the green and brown regions often change their locations. This situation happens as the StreamGBH can not handle large motion and difference between neighboring frames. (The images are best seen in color) (Colour figure online).

train SVM classifiers based on these features. We test the classifiers' performance on the 654 test images. The average pixel precision and class precision are listed in Table 2. We compute the class precision as follow:

$$\frac{TruePositives}{TruePositives + FalsePositives + FalseNegatives} \quad (11)$$

We get an average pixel accuracy in 73.45% and average class accuracy in 45.42%.

Table 2. Performance of the labeling on the object category.

wall	floor	cabinet	bed	chair	table	window
73.69	81.98	44.34	49.74	46.81	31.78	35.15
bookshelf	picture	blinds	ceiling	tv	per-pixel	per-class
30.57	37.91	37.91	53.08	22.13	73.45	45.42

Table 3. Semantic mapping results

	Baseline	Our
Scene 1	88.61 %	91.73 %
Scene 2	84.50 %	86.93 %
Scene 3	86.32 %	88.82 %
Scene 4	78.03 %	82.92 %
Average	84.36 %	87.60 %

Fig. 4. The visualization of the semantic maps. (The images are best seen in color) (Colour figure online).

In order to get the global 3D point cloud for sequence RGB-D scene, we use the 3D transformation between frames to get the estimated camera poses and register the RGB-D sequences. As a baseline, we independently segment each RGB-D image and classify the superpixels, the 3D point is labeled by a simple majority vote for the labeled 2D pixels which are fused in the same 3D point. In our approach, the temporal consistent segmentations and consistent semantic labels are obtained for each RGB-D image through the method mentioned in Sect. 3. Then a Dense CRF model is used to inference the global 3D point labels. The comparisons for 4 RGB-D scenes are given in Table 3.

We get an average precision of 87.60 % with an improvement of 3.24 %. Compared to the temporal consistent segmentation, independent segmentations often lead to inconsistent semantic precisions. For example, one superpixel which is labeled as window in frame t may vanish in frame $t + 1$. Through the consistent segmentation, the consistent semantic precisions over superpixels are enforced. Compared to the majority voting strategy, Dense CRF smooths and distributes the information over neighboring points and the spatial consistency is guaranteed. The visualizations of the semantic maps are shown in Fig. 4, which the global point cloud and semantic labels are shown.

Fig. 5. Sample corresponding objects discovered by temporal consistent segmentations.

6 Conclusion

In this paper, we propose a spatio-temporally consistent semantic mapping approach for indoor scene. By using 3D transformation between frames, we can compute more accurate correspondence for superpixels than the optical flow algorithm. In this way, the performance of our temporal consistent segmentation approach is superior than the state-of-art StreamGBH approach. Also, Dense CRF model enforces the spatial consistency by distributing the information over neighboring points depending on the distance and color similarity.

In fact, more consistent constraints can be enforced by the temporal consistent segmentations. Through the consistent segmentations, we can find the corresponding objects on the different images, as shown in Fig. 5. These corresponding objects on different images should be labeled by the same semantic class. This cue can be enforced by CRF or high-order CRF model. Also, the temporal consistent segmentation can be used for unsupervised object discovery. In this further, we will explore them both.

References

1. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. *ACM Trans. Graph. (TOG)* 28, 70 (2009)
2. Banica, D., Sminchisescu, C.: CPMC-3D-O2P: Semantic segmentation of RGB-D images using CPMC and Second Order Pooling. *CoRR*, abs/1312.7715 (2013)

3. Chen, A.Y., Corso, J.J.: Propagating multi-class pixel labels throughout video frames. In: 2010 Image Processing Workshop (WNYIPW), Western New York, pp. 14–17. IEEE (2010)
4. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 53–60. IEEE (2006)
5. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **59**(2), 167–181 (2004)
6. Gupta, S., Arbelaez, P., Malik, J.: Perceptual organization and recognition of indoor scenes from RGB-D images. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 564–571. IEEE (2013)
7. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: using kinect-style depth cameras for dense 3D modeling of indoor environments. *I.J. Robotic Res.* **31**(5), 647–663 (2012)
8. Koppula, H.S., Anand, A., Joachims, T., Saxena, A.: Semantic labeling of 3D point clouds for indoor scenes. In: NIPS, pp. 244–252 (2011)
9. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. *CoRR*, abs/1210.5644 (2012)
10. Moore, A.P., Prince, S., Warrell, J., Mohammed, U., Jones, G.: Superpixel lattices. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2008)
11. Ren, X., Bo, L., Fox, D.: RGB-(D) scene labeling: features and algorithms. In: CVPR, pp. 2759–2766 (2012)
12. Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 593–600. IEEE (1994)
13. Silberman, N., Fergus, R.: Indoor scene segmentation using a structured light sensor. In: ICCV Workshops, pp. 601–608 (2011)
14. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012)
15. Xu, C., Corso, J.J.: Evaluation of super-voxel methods for early video processing. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1202–1209. IEEE (2012)