Viewpoint-Independent Book Spine Segmentation

Lior Talker
The Interdisciplinary Center
Herzliya 46150, Israel
talker.lior@idc.ac.il

Yael Moses
The Interdisciplinary Center
Herzliya 46150, Israel
yael@idc.ac.il

Abstract

We propose a method to precisely segment books on bookshelves in images taken from general viewpoints. The proposed segmentation algorithm overcomes difficulties due to text and texture on book spines, various book orientations under perspective projection, and book proximity. A shape dependent active contour is used as a first step to establish a set of book spine candidates. A subset of these candidates are selected using spatial constraints on the assembly of spine candidates by formulating the selection problem as the maximal weighted independent set (MWIS) of a graph. The segmented book spines may be used by recognition systems (e.g., library automation), or rendered in computer graphics applications. We also propose a novel application that uses the segmented book spines to assist users in bookshelf reorganization or to modify the image to create a bookshelf with a tidier look. Our method was successfully tested on challenging sets of images.

1. Introduction

We propose a method to precisely segment books on bookshelves. Books may have different orientations and the image may have a perspective distortion. The visible part of a book on a bookshelf is the book spine (or *spines* for short), and other parts are typically occluded. Hence, we simplify the task of book segmentation to the segmentation of their spines (see example in Fig. 1).

Book spine segmentation, however, poses unique, non-trivial challenges. Books are typically located very close to each other and are highly textured. Most existing region based segmentation methods assume that objects are separated and surrounded by a smooth background; hence, they are expected to fail on this task, as we demonstrate in Sec. 5. Similarly, active contour methods that initialize a contour considerably far from an object are expected to fail due to convergence to a region that covers a set of spines. Object segmentation methods that learn object instances of a given class are also expected to fail due to the large variability of book spines and their density on the shelves.

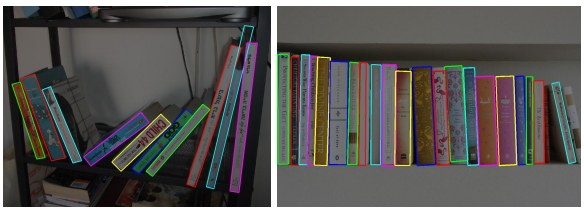


Figure 1: Book spine segmentation results.

We propose a method that overcomes these challenges, where the multiplicity of books is used to obtain robustness. Our method consists of two phases. The first allows us to obtain a set of spine candidates with many false positives, but relatively very few false negatives. In the second phase we use spatial constraints to filter the set of spine candidates.

The first phase is a modification of the active contours paradigm with a predefined contour shape, a perspective projection of a rectangle - PR (Sec. 3.1). We assume that all book spines are coplanar or lie on parallel planes (identical 3D normals), but may have different orientations within these planes. Thus, although the edges of each spine are consistent with a pair of orthogonal vanishing points, different spines may have a different pair of vanishing points (unlike in *Manhattan world*). However, the vanishing points of all spines are incident to the same vanishing line. These observations are used to constrain the shape of a PR. The PR "snake" expands from a seed point to obtain minimal energy.

In the second phase the set of the detected spine candidates is filtered according to the size statistics and relative spatial location (Sec. 3.2). In particular, the selection problem is formulated as the maximal weighted independent set (MWIS) of a graph that corresponds to the spatial relations between the spine candidates.

Applications: Book recognition can be used to automate libraries, e.g., managing an inventory of books. The segmented book spines may be used in a preprocessing step for book recognition systems that are based either on optical character recognition (OCR, e.g., [6, 12, 18]), or on fea-

ture recognition (e.g., [6]). Computer graphics and virtual reality applications may use the precisely segmented book spines to automatically populate virtual bookshelves with real book spines.

We also propose a novel application that assists users in reorganizing books on their bookshelves according to some attribute (e.g., height). A related application reorganizes the bookshelves directly in the image in order to obtain a tidy looking shelf. In Sec. 4 we propose to use a reposition algorithm that preserves the correct size of the spines in the presence of perspective distortion.

2. Related Work

The objective of most book segmentation methods is to automate library related tasks, for example, managing an inventory of books. The majority of these studies [6, 12, 18] segment book spines as a step towards book identification. Book spines are segmented by grouping adjacent long line segments. OCR methods [12, 18], or feature-based methods [6], are then used for book identification. Taira *et al*. [14] focused only on book spine segmentation and used a finite state machine (FSM) in order to define rules for grouping the extracted line segments. For example, a line created by the text of the title often appears between long vertical line segments, which are the boundaries of the book. None of these studies aim to obtain a precise segmentation of the spine, nor, in particular, to detect its upper part, although doing so is essential for graphics and book reorganization applications and might also improve the performance of book identification systems.

Moreover, most algorithms mentioned above [14, 6, 12] assume orthographic projection, and most of these [6, 12, 18] also assume that all of the books have the same orientation in the plane; [12] further assumes that they are roughly aligned with the axis of the image. These assumptions are clearly too restrictive for the general bookshelf images considered in this paper. Furthermore, the performance of these algorithms is affected by book texture, since they rely on long line primitives, coupled by their order. In particular, long line extraction, using line-fitting [6, 12, 18] or the Hough transform [14], often produces false negatives in a cluttered texture-full environment such as a bookshelf. These failures are demonstrated in Sec. 5. In our study we consider perspective projections of book spines taken from any viewpoint, at any orientation.

Perspective rectangle detection algorithms were used for detecting windows and doors in offices and hallways [11, 13], or to segment buildings or windows in urban environments (e.g., [11, 13]). However, these algorithms are expected to provide poor results on spine segmentation since they assume all rectangles are aligned in the scene (the *Manhattan world* assumption). In addition, these methods assume that the perspective rectangles are surrounded by a textureless background.

The segmentation of rectangles using active contours was proposed by [1] for medical segmentation of intervertebral disks in the human spine. A rectangle shaped contour, that undergoes rotation and scaling in the image plane is employed in their algorithm. Their contour energy score is based on Chan *et al*. [5]. This energy score is not suitable for book spine segmentation since it is based on the assumption that the object to segment is of (almost) constant color. Furthermore, the rectangle they employ is a semi-affine transformation applied to the unit square, which represents only right-angle rectangles and not perspective projections of rectangles.

3. Book Spine Segmentation

An image of books placed on bookshelves is the input to our method. We assume that all book spines are parallel to a plane π. The output is a set of spines, each having the shape of a perspective projection of a rectangle (PR). Our method consists of two phases. The first is the identification of a set of PR candidates that are consistent with the image gradients and the expected perspective. The second is the filtering of this set according to the candidates' location, relative size and spatial relations. (See pseudocode in the supplemental material.)

3.1. Identifying PR Candidates

We present a method to detect a set of PRs that are candidates to segment book spines. We formulate the PR detection as a minimization problem over five parameters.

3.1.1 PR Parametrization

A straightforward parametrization of a general quadrilateral is given by the location of its 4 vertices (8 parameters). Here we assume that an internal point s of the PR is available, as well as two vanishing points v and $v_\perp$ that correspond to orthogonal directions in the plane π (see Sec. 3.1.4). Using these assumptions, we show that 5 parameters are sufficient for the parametrization of a PR.

Let us first consider a rectangle (i.e., no distortion due to perspective). In this case, a single orientation parameter, α, can determine the orientation of the 4 edges. The size and location of the edges can be determined by 4 values, the distance from a given internal point s to each of the edges.

When PRs are considered, similar parametrization can be used. In this case, the orientation of each edge depends on the edge location and its corresponding vanishing point, v or $v_\perp$. A single point, q_i, on an edge, e_i, together with its corresponding vanishing point, (say) v, defines the edge orientation. To compute q_i, we use the *oriented distance* d_i from s, along a line in the direction of the orthogonal vanishing point $v_\perp$ given by $\hat{u}(s, v_\perp)$ (see Fig. 2). Formally, q_1

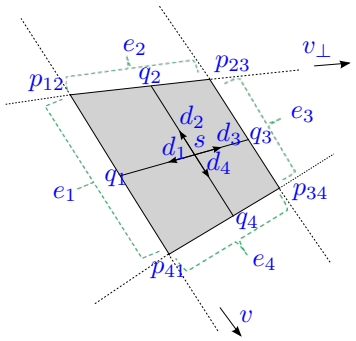


Figure 2: PR parametrization.

and q_3 are given by:

$$q_1 = s - d_1\hat{u}(s, v_\perp) \;\;\&\;\; q_3 = s + d_3\hat{u}(s, v_\perp). \quad (1)$$

In a similar manner we define q_2 and q_4. Let l_1 be a line in the image through q_1 and v, and l_2 be a line in the image through q_2 and $v_\perp$. A vertex p_{12} is defined by the intersection point between l_1 and l_2, in homogenous coordinates:

$$\tilde{p}_{12}(d_1, d_2) = \tilde{l}_1 \times \tilde{l}_2. \quad (2)$$

The other vertices, $p_{ij}(d_i, d_j)$, are defined in a similar manner. Finally, a PR edge, e_i, is the line segment between two vertices, p_{ji} and p_{ik}, that is, $e_i = [p_{ji}, p_{ik}]$. Note that p_{ij} and e_i are functions of the set $D = \{d_i\}_{i=1}^4$ and α; the parameter α defines the pair of orthogonal vanishing points, v and $v_\perp$, as shown in Sec. 3.1.3.

3.1.2 PR Expansion

Given a seed point s and an initial orientation α^0, we search for a PR, defined by D and α that best agree with the image gradients. The search is performed in rounds. In each round each $d_i \in D$ is either incremented by one pixel, or remains unchanged. In the first case we say that the parameter d_i is in an *active* state, and in the second it is in an *inactive* state. After each round, α is updated and each d_i may enter or exit the active state according to the consistency of the PR with the image gradients. When all d_i are inactive simultaneously, the process halts. Each $d_i \in D$ is initially in the active state and $d_i = 1$.

Note that the value of d_i determines both the location of the edge e_i and the length of its adjacent edges. Hence, incrementing d_i affects the consistency of the image gradients with the edge e_i as well as with the extension segments of the adjacent edges (see the yellow segments in Fig. 3). The inactive state is determined by both consistencies, as next defined.

An edge consistency (edge energy) with the image gradient is defined to be the mean angle between the edge normal, $\hat{n}(e)$, and the image gradients along e. Formally,

$$E(e) = \frac{\sum_{p\in e} \arccos(|\hat{\nabla} I(p) \cdot \hat{n}(e)|)}{|e|}, \quad (3)$$

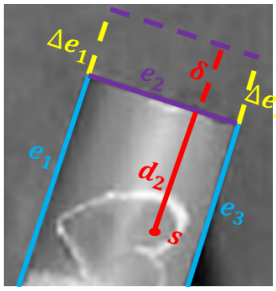


Figure 3: Edge extension segments are marked in yellow.

where $p \in e$ is a pixel on the edge e, $\hat{\nabla} I(p)$ is the gradient at p, and $|e|$ is the edge length measured in pixels.

For d_i to enter the inactive state, three conditions must hold. (i) $E(e_i(D))$ is a local minimum as a function of d_i. (ii) $E(e_i(D)) < t_G$, where t_G is a predefined threshold. (iii) The adjacent edges do not support the incrementing of d_i. This last condition is introduced in order to avoid halting at a local minimum. To formulate this support, we consider the *edge extension segment* of an adjacent edge where d_i is incremented by δ (Fig. 3). Formally, let e_j be an adjacent edge of e_i and define its edge extension segment by:

$$\Delta e_j(d_i, \delta) = [p_{ij}(d_i, d_j), p_{ij}(d_i + \delta, d_j)].$$

If the edge energy of at least one of the edge extension segments of e_i, Δe_j and Δe_k, is higher than βt_G, that is, $max\,(E(\Delta e_j), E(\Delta e_k)) > \beta t_G$, then d_i enters the inactive state.

To avoid convergence to textures inside the spine, e.g., due to text, an inactive d_i may return to the active state, if the consistency of e_i with the image gradients changes. This may occur when e_i lengthens due to changes in the adjacent edges, i.e., the increase of d_j and d_k. A d_i returns to the active state when $E(e_i) < \lambda t_G$, where $\lambda > 1$ is a constant defined to prevent frequent state changes.

At the end of each round, the current orientation of the PR, α_c, is chosen by testing the consistency of the PR edges with the image gradients for various orientations. In our implementation we consider 10 values of α, equally distributed, in the range $[\alpha_c - \frac{\pi}{36}, \alpha_c + \frac{\pi}{36}]$ (all values are in radians). That is,

$$\alpha_c = \arg\min_{\alpha'} \left(\sum_{i=1}^{4} E(e_i(\alpha')) \right).$$

3.1.3 Orthogonal Pairs of Vanishing Points

To preserve a PR shape, a pair of orthogonal vanishing points, $(v(\alpha), v_\perp(\alpha))$, is required for each orientation α. We assume that all spines are on planes parallel to π; hence, all vanishing points are located on a vanishing line, ℓ. As a preprocessing step, we compute ℓ by detecting the two vanishing points that correspond to the dominant orthogonal

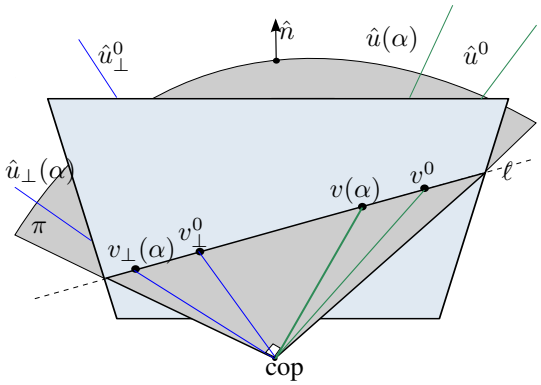


Figure 4: Orthogonal pairs of vanishing points. The gray triangle indicates the plane of book spines π.

directions in the image, v^0 and $v^0_\perp$, using [17]. The dominant orthogonal directions are assumed to be on π, since we assume that the image contains a sufficient number of books. An additional preprocessing step is computing the internal calibration matrix, K. To do so, we assume the camera has square pixels, no skew, and the principle point is in the middle of the image. These assumptions provide sufficient constraints on K for a linear solution, using the additional constraints of the orthogonality of the vanishing points v and $v_\perp$ [8].

Given the internal calibration matrix, K, and the two orthogonal vanishing points, the normal $\hat{n}$ to the plane π in the camera coordinate system can be computed as follows. The vanishing point v that corresponds to a 3D direction $\hat{u}$ is given by $\tilde{v} = M(\hat{u}\ \ 0)^T$. We choose the world coordinate axes and the camera coordinate axes to be identical; hence, $\tilde{v} = K\hat{u}$. In particular, the direction $\hat{u}^0$ that corresponds to the vanishing point v^0 is given by $v^0 = K\hat{u}^0$.

Since $\hat{u}$ and $\hat{u}^0$ are both parallel to the plane π, they are related by $\hat{u} = R(\hat{n}, \alpha)\hat{u}^0$, where R is a rotation matrix about axis $\hat{n} = \hat{u}^0 \times \hat{u}^0_\perp$ and angle α. In a similar manner it can be shown that any pair of orthogonal vanishing points, as a function of α, is given by (Fig. 4):

$$\begin{aligned} v(\alpha) &= K\hat{u} = KR(\hat{n}, \alpha)\hat{u}^0 = KR(\hat{n}, \alpha)K^{-1}v^0. \\ v_\perp(\alpha) &= KR(\hat{n}, \alpha)K^{-1}v^0_\perp. \end{aligned}$$

3.1.4 Initialization

To initialize the set of seed points, we extract line segments using [10], and filter segments shorter than a predefined threshold (we use 10% of the image height). Seed points are placed at equal distances along each side of each segment at a small perpendicular offset (Fig 6(b)). In our implementation we set 10 seed points along each line, and an offset of 5 pixels.

To obtain an initial estimation of α for a given seed point, the dominant gradient direction in a $h \times w$ of its neighborhood (we use 50×50 pixels) is computed. To obtain very accurate gradient directions, we use the third-order edge operator [15]. The intersection of this direction with the vanishing line determines its corresponding vanishing point v. The initial estimation of α^0 is taken as the angle between the directions that correspond to v and v^0. That is, $\alpha^0 = \arccos(\hat{u}^T \cdot \hat{u}^0)$, where $\hat{u} = K^{-1}v$ and $\hat{u}^0 = K^{-1}v^0$ are the directions that correspond to v and v^0 respectively.

3.2. PR Selection

The computed set of PRs consists of the desired book spines as well as many other PRs that should be discarded. These include PRs that segment several books together (e.g., Fig. 5b), sub-regions of a spine (e.g., Fig. 5a), other objects in the scene, or simple noise. In addition, several PRs may segment almost the same region in the image. We present an algorithm to select the set of PRs that most probably represent book spines, using the assembly of books.

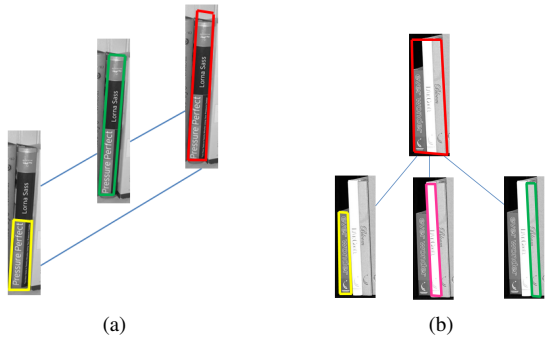
3.2.1 Location and Size

We assume that the detected books are either positioned on a set of parallel bookshelves or supported by other books that are positioned on a bookshelf (e.g., a pile of books). The random sample consensus algorithm (RANSAC) [7] is used for estimating a bookshelf line with the set of lower vertices of the PRs as input. This avoids direct detection of the bookshelf edge. To detect n shelves, RANSAC is executed n times where in each run the inliers from previous runs are excluded (in our implementation we assume that $n \leq 10$). The detected bookshelf lines that do not agree with a single vanishing point are removed using RANSAC. Finally, the set of detected bookshelves is used to recursively select the set of PRs supported by either a bookshelf or by another book. All other PRs can be discarded.

We wish to filter PRs according to their size; however, their size depends on their 3D location due to perspective. To avoid a direct 3D reconstruction, we perform a normalization based on the perspective aware repositioning of planar objects algorithm by Tolba *et al.* [16] (see Sec. 4). The result is the set of PRs as if they were all projections of spines incident to the same scene location. In particular, p_{41}, e_1 and e_4 of all PRs are aligned. The statistics of the length of the repositioned e_1 and e_4 are used to discard PRs. (We discard σ below or 2σ above the mean for both e_1, e_4.)

3.2.2 Spatial Relations

The set of remaining PRs contains large overlap between PRs. To obtain a subset of disjoint PRs that best represent the book spines, we model the PRs and the spatial relations between them as an undirected graph $G = (V, E)$. A node $v \in V$ represents a PR and an undirected edge $(v_i, v_j) \in E$ represents a normalized rate of overlapping of at least γ between v_i and v_j (in our implementation $\gamma = 0.1$).



(a) (b)

Figure 5: MWIS examples. a) Horizontal overlap; b) horizontal span.

That is,

$$\frac{|R(v_j) \cap R(v_i)|}{min(|R(v_i)|, |R(v_j)|)} \geq \gamma,$$

where $|R(v)|$ is the area of the PR region, $R(v)$, that correspond to v. By definition, every independent set defined over this graph, i.e., a subset of non-adjacent nodes, corresponds to a subset of disjoint PRs.

To obtain an independent set which also corresponds to the most probable PRs to represent the book spines, we assign weights, $w : V \rightarrow \mathbb{R}^+$, to the nodes $v \in V$. The weight $w(v)$ captures the preference of choosing v over a subset of its neighbors. The maximal weighted independent set (MWIS) of the weighted graph, $G(V, E, w)$, corresponds to the subset of disjoint PRs that are preferred over their neighbors.

The most common spatial relation between PRs is due to multiple detections of the same spine but truncated at different heights (see Fig. 5a); hence, we set $w'(v)$ to be inverse proportional to the gradient support of the vertical edges, $E(e_1)$, $E(e_3)$, and the height, $h(v)$ (approximated by $|e_1|$). That is,

$$w'(v) = h(v)\left(1 - \frac{E(e_1) + E(e_3)}{2t_{\mathrm{G}}}\right),$$

where t_{G} is the energy threshold introduced in Sec. 3.1.2.

Another common spatial relation between PRs is a single PR that covers a set of PRs that corresponds to adjacent book spines. In this case the set is preferred over the large PR; thus, we set $w(v) = 0$. Formally, $HS(v) \subseteq N(v)$ is a horizontal span of $v \in V$ *iff* three conditions hold. (i) $|HS(v)| \geq 2$. (ii) There is no overlap between any $u_1, u_2 \in HS(v)$, i.e., $(u_1, u_2) \notin E$. (iii) $HS(v)$ covers the horizontal dimension of v. See Fig. 5b. Formally, let $x_l(u)$, $x_r(u)$ be the x coordinates of the bottom left and bottom right corners of $u \in HS(v)$. There exists $u, u' \in HS(v)$ such that $x_l(u) \approx x_l(v)$ and $x_r(u') \approx x_r(v)$. Furthermore, for each $u_i \in HS(v)$, $u_i \neq u'$, there exists $u_j \in HS(v)$ such that $x_r(u_i) \approx x_l(u_j)$.

To determine whether v does in fact have a horizontal span, we sort $u \in N(v)$ by $x_l(u)$. Then, for each $u_1 \in N(v)$, we add it to $HS(v)$ if there exists $u_2 \in HS(v)$ for which the last two conditions in the horizontal span definition hold.

To summarize, the weight of a node v is given by:

$$w(v) = \begin{cases} 0 & \exists HS(v) \subseteq N(v), \\ w'(v) & otherwise. \end{cases} \quad (4)$$

Note that $w(v)$ is non-negative for all $v \in V$.

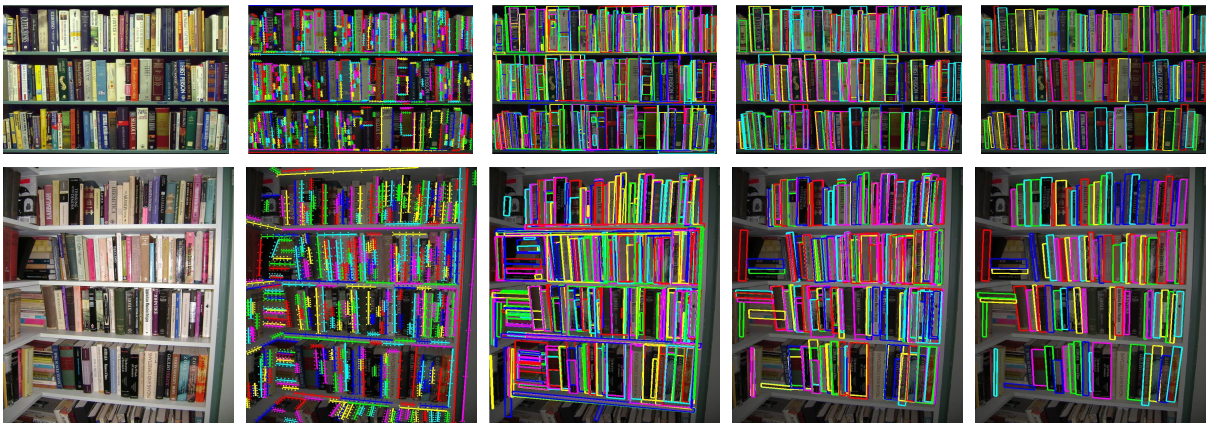
The MWIS problem is NP-hard for a general input. In our method, the input (the PRs) is not bounded in a way that allows the optimal solution to be obtained in polynomial time; therefore, the approximation from [4] is used.

The MWIS was used previously for segmentation by the authors of [4], to select "meaningful" segments from a hierarchy of segmentation maps. In their method, however, the segment weight corresponds to the prediction of the segment appearance using other parts of the image. The (hidden and) underlying assumption is that different segments are fundamentally different in appearance and act as background for each other, which is not true in our case, where, for example, two segments of two different book spines are similar in appearance.

4. The Bookshelf Reorganizer Application

The precise spine segmentation computed by our method may be used as an input for graphic applications or as a preprocessing step for book recognition methods. We propose a novel application in which the spines may be used to generate an image in which the books on the bookshelves are reorganized. The new book order is automatically set according to a user-chosen criterion, e.g., height, width, and color, which as we next show can be directly computed from the spines. If OCR is available, title or author name may be used to set the order as well. This application allows a user to consider several alternatives for the bookshelf look, before actually moving books around. In addition, it allows books to be reorganized directly in the image to obtain a nicer looking and tidier bookshelf (e.g., all books are aligned).

Given the segmented spines, the main challenge is to change the book locations while preserving their size and shape with respect to the perspective view of the scene. To this end, we apply Tolba *et al.*'s [16] method for perspective-aware repositioning of planar objects. Translations and rotation are modeled as families of homographies that depend on the normal to the planar object, the translation distance in the image plane, and the rotation axis and angle. We use this method with the input of the normal $\hat{n}$ to the plane π (computed in Sec. 3.1.3), as both the normal to the planar object and the axis of rotation. The translation distance and rotation angle are derived from the user-chosen reorganization criterion. Sorting the book spines' height



(a) (b) (c) (d) (e)

Figure 6: Book spine segmentation steps. From left to right: a) original image, b) seed point and line segments, c) PR candidates, d) PR candidate after location and size filtering, e) final segmentation results (after MWIS).

Figure 8: The bookshelf reorganizer examples.

(a) (b)

Figure 9: a) Segmentation results of the OWT-UCM method by Arbelaez *et al*. [2]; b) An example of segmentation failure of [6], due to the violation of the uniform orientation assumption, on image Fig. 1. Notice that the grouping of adjacent long line segments suggested in [6] is not applicable in this example.

or width is necessary when the user chooses to reorder the books by these criteria. To sort the height or width despite the perspective distortions, the book spines are aligned to the same image location, as described in Sec. 3.2.1.

To obtain a pleasing image, the remaining challenge is to fill the uncovered parts of the books, e.g., the upper parts, and the gaps that remain after the books are reorganized. We have experimented with two approaches to rectify these gaps. The first is to render the reorganized book spines on a clean bookshelf, i.e., a user-chosen background image. For a realistic look, we arbitrarily chose depth for the books in order to reconstruct their missing parts, e.g., the upper part of the books. The images were rendered in OpenGL, and the results are presented in Fig. 8. The second approach is to inpaint the gaps, which was proven to be difficult and is left for future work.

To overcome segmentation errors, the application may also be in interactive mode and display the segmentation results to the user. The user can delete and add PRs. The latter can be done manually or by choosing PRs from the initial set of book spine candidates (before the PR selection phase). Note that all the results presented in the paper were generated completely automatically.

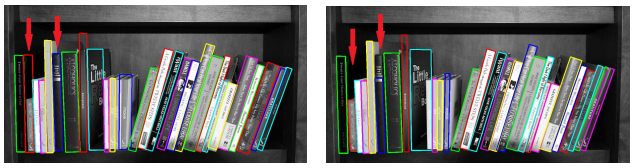
5. Experimental Results

We implemented our algorithm in Matlab and tested it on a representative set of images, some collected from the Internet and some taken by us (the code and the images will be available). We considered 2 datasets. The first contained 45 images (1163 book spines), mostly with uniformly orientated book spines on a single shelf, with only negligible perspective distortion (left in Fig. 7). The second contained 27 images (1235 book spines), of arbitrarily oriented book spines and multiple shelves, mostly under perspective projection view (center and right in Fig. 7).

Segmentation Examples: A few segmentation results are presented in Fig. 7. It can be seen that the majority of book spines are correctly and precisely segmented, and there are relatively few false positive detections. Example of errors e.g., an overly extended spine, a partially segmented book spine, unsegmented spine, and noise, are marked in Fig. 7. Some of these failures are due to the choice of parameters, as we next discuss.

Parameters: For quantitative evaluation, described below, the same set of parameters is used in all experiments, despite the large variability in image resolution, number of

Figure 7: Book spine segmentation results and typical segmentation errors. Left, dataset 1; middle and right, dataset 2. Arrows mark segmentation errors: overextension (yellow); partial segmentation (red); false negative (white); false positive (green).



$t_G = 0.3$ $t_G = 0.35$

Figure 10: Different segmentation results with different parameters. The red arrows indicate segmentation errors.

books, etc. The general performance is not sensitive to parameter changes in the following ranges: (i) $t_G \in [0.25, 0.4]$ (Sec. 3.1.2) in radians. (ii) The parameters for filtering by size $\in [\sigma, 3\sigma]$ and $[0.5\sigma, 1.5\sigma]$ (Sec. 3.2.1) for filtering above and below the mean, respectively. Better results can be obtained if the parameters are tuned for a specific set of images. An example of the effect of different parameters on the segmentation results is presented in figure 10.

Quantitative Evaluation: For quantitative evaluation, the ground truth segmentation was manually generated. We considered only fully visible book spines and discarded those that intersect with image boundaries. We used the *Hoover Index (HI)* [9], in which the computed segments are labeled as correct detection, oversegmentation, undersegmentation, missed or noise. The normalized overlap between a computed segment, $R(c)$, and a ground truth segment, $R(g)$, is used to determined a correct detection. That is, $|R(c) \cap R(g)| / \max(|R(c)|, |R(g)|) \geq \gamma$ where γ is a predefined threshold (we use $\gamma = 0.8$). The precision and recall were calculated using only the correct detections and combined to a total score as a harmonic mean of precision-recall. Since we use a randomized approximation to the MWIS, we present the mean and standard deviation of 10 runs. Dataset 1 reached a mean/std precision-recall of 90.44%/0.3% (precision 92.61%/0.28%, recall 88.38%/0.32%) and dataset 2 reached a precision-recall of 70.78%/0.58% (precision 73.95%/0.35%, recall 67.87%/0.81%).

We next analyze the effectiveness of the first and second phases in terms of their contribution to the final segmentation results. The first phase produces a high recall rate ($\sim$95% for dataset 1 and $\sim$82% for dataset 2) and, as expected, a low precision rate ($\sim$38% for both). The goal of the second phase is to increase the precision rate while preserving most of the recall rate. Table 2 shows the contribution of location and size (Section 3.2.1), as well as that of the MWIS (Section 3.2.2), in the second phase. According to our analysis, the contribution to the increase in precision is mostly due to the size and the MWIS. As a result of the increase in precision, the recall was affected mostly by the MWIS ($\sim$ 6%,13% decrease of recall for datasets 1 and 2, respectively).

Comparison: We did not compare our segmentation method to previous book spine recognition methods ([6, 12, 18, 14]) because the restrictive assumptions they used are not applicable to our data (as demonstrated in Fig. 9). Therefore, we chose instead to compare our results to a state-of-the-art general segmentation algorithm. We use the top-performing segmentation algorithm in the Berkeley Segmentation Data Set 500 (BSDS500) [3]: the OWT-UCM segmentation algorithm suggested by Arbelaez *et al.* [2]. We used the available code from the Web.

An example of a typical result (Fig. 9) demonstrates the expected failure of a general segmentation method that does not use the available domain specific constraints. A quantitative comparison of our and the OWT-UCM methods is presented in Table 1. In addition to the Hoover Index, we also use the evaluation schemes used in the BSDS500: *segmentation covering (SC)*, *variation of information (VI)*, and *probabilistic rand index (PRI)* [2]. Segmentation covering measures the maximum area covered by the computed segmentation for each segment in the ground truth. Variation of information corresponds to the entropy of the disjoint pixels between the segmentation and the ground truth, and the probabilistic rand index measures the probability that a given pair of pixels belongs to the same label in the segmentation and in the ground truth. Arbelaez *et al.* evaluated their algorithm in terms of its *optimal dataset scale (ODS)*,

	HI		SC		PRI		VI	
	Ours	**Arbelaez [2] (ODS/OIS)**	**Ours**	**Arbelaez [2] (ODS/OIS)**	**Ours**	**Arbelaez [2]**	**Ours**	**Arbelaez [2]**
Dataset 1	0.9044	0.3268 (Th:0.3) / 0.3472	0.7849	0.4781 (Th:0.15) / 0.5154	0.9174	0.8691	1.3782	2.1523
Dataset 2	0.7078	0.1759 (Th:0.2) / 0.1932	0.7952	0.4911 (Th:0.05) / 0.5668	0.8397	0.7203	1.3097	2.4090

Table 1: Segmentation results of our method and the OWT-UCM (Arbelaez *et al.*) method.

	Precision / Recall (∼%)			
	Initial	**Location**	**Size**	**MWIS**
Set1	38 / 95	42 / 95	65 / 94	93 / 88
Set2	38 / 82	44 / 82	67 / 81	74 / 68

Table 2: Effectiveness analysis of the second phase. The parameters used for this analysis (as well as the results and generated images in this paper) : $t_G = 0.3$, size filtering above the mean $= 2.5\sigma$, and below $= \sigma$.

which sets a uniform scale for all images, and its *optimal image scale (OIS)*, which sets different and optimal scales for each image. Our method significantly outperforms the OWT-UCM algorithm in every evaluation criterion, e.g., we reach 87.49% and Arbelaez *et al.* reach 32.68% (ODS) and 34.72% (OIS) on dataset 1, with the HI criterion. Note that the VI criterion is higher as the segmentation deteriorates.

Bookshelf Line Detection: The results were evaluated manually. Here, the recall is more important than the precision, because of its influence on the end results of the book spine segmentation, i.e., an undetected shelf line results in multiple unsegmented book spines. The bookshelf line detection reached a recall of 88.14% over the 2 datasets together, and a lower precision of 58.43%. Most of the undetected bookshelf lines consist of a small number of book spines, or book spines that do not form a straight line.

6. Conclusion and Future Work

We proposed a solution to the challenging problem of book spine segmentation in images under perspective projection, where the books are not aligned. Our two phase solution allows us to obtain a large set of spine candidates using a bottom-up computation, and a top-down computation to filter the results. The high recall of the first phase is due to the use of the PR, rather than line segments as a basic primitive. The success of the second phase comes from analyzing the assembly of book spine candidates.

We also proposed methods for dealing with untidy bookshelves. Although book segmentation is non-trivial, the desired reorganization of a bookshelf is relatively easy to define. For short-term future work, the gap inpainting should be resolved. For the long term, object segmentation may be used to reorganize other untidy parts of a room.

Acknowledgments: Research supported in part by the Israeli Ministry of Science and Technology, grant number 3-8700.

References

[1] A. Abufadel, G. Slabaugh, G. Unal, L. Zhang, and B. Odry. Interacting active rectangles for estimation of intervertebral disk orientation. In *ICPR*, 2006.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011.

[4] W. Brendel and S. Todorovic. Segmentation as maximum-weight independent set. In *NIPS*, 2010.

[5] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[6] D. Chen, S. Tsai, C. H. Hsu, J. Singh, and B. Girod. Mobile augmented reality for books on a shelf. In *ICME*, 2011.

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ Press, 2000.

[9] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *TPAMI*, 18(7):673–689, 1996.

[10] J. Košecká and W. Zhang. Video compass. In *ECCV*. 2006.

[11] B. Micusik, H. Wildenauer, and J. Košecká. Detection and matching of rectilinear structures. In *CVPR*, 2008.

[12] N. H. Quoc, K. H. Woo, and W. H. Choi. Segmentation of books and characters for book recognition system of robot intelligence. In *ICCAS-SICE*, 2009.

[13] D. Shaw and N. Barnes. Perspective rectangle detection. In *ECCV, Application Workshop*, 2006.

[14] E. Taira, S. Uchida, and H. Sakoe. Book boundary detection from bookshelf image based on model fitting. In *ISEE*, 2003.

[15] A. Tamrakar and B. Kimia. No grouping left behind: From edges to curve fragments. In *ICCV*, 2007.

[16] O. Tolba, J. Dorsey, and L. McMillan. A projective drawing system. In *I3D*, 2001.

[17] E. Tretyak, O. Barinova, P. Kohli, and V. Lempitsky. Geometric image parsing in man-made environments. *IJCV*, 97(3):305–321, 2012.

[18] S. Tsai, D. Chen, H. Chen, C. H. Hsu, K. H. Kim, J. Singh, and B. Girod. Combining image and text features: a hybrid approach to mobile book spine recognition. In *ICME*, 2011.