The Interdisciplinary Center, Herzliya
Efi Arazi School of Computer Science

Book Spine Segmentation for Bookshelf Reorganization

M.Sc. Dissertation

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science (M.Sc.) Research Track in Computer Science

Submitted by Lior Talker
Under the supervision of Dr. Yael Moses

July 2013

Acknowledgments

I would like to express my immense gratitude to my supervisor, Dr. Yael Moses, who, with her professional attitude, taught me what real research is and how it is done, through countless eye-opening discussions. From time to time, with great patience and constant good will, she put me back on course as I wandered through this interesting experience.

I would also like to express my gratitude to my spouse, Eve, who tolerated me when I was preoccupied with many hours and days of work, and also provided some very good advice.

Abstract

A tidy bookshelf is pleasant and gives a nice impression of a room. However, many bookshelves are disordered and tidying them takes considerable effort. Our study is motivated by the quest to develop an application for tidying bookshelves directly in the image. The main challenge is to correctly segment the book spines. We propose a segmentation algorithm that 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 at the center of our method and used to establish a set of spine candidates. A subset of these candidates is then selected by enforcing spatial constraints on the assembly of books. We use the segmented book spines to generate a new image of the bookshelves, in which the books are reorganized, for example, by aligning the orientations of their spines, or by other criteria such as height, width, or color. The size of each book spine in the reorganized image is preserved with respect to the perspective view, without recovering the 3D structure of the scene.

Table of Contents

List of Figures

List of Equations

Chapter 1

Introduction

Tidying up bookshelves is time consuming and the results are temporary, as the books *tend to become disorganized*. This nuisance motivated our development of a method for organizing the bookshelves directly in the image, so that, for photographic purposes at least, they appear organized. This can be done for example, by rotating all the books to a uniform orientation, or rearranging them according to their height or other criteria (see example in Figure 1.1). The resulting image can also assist in planning a new order for the real bookshelves, to be applied later by the user.

We propose a method to segment books on bookshelves, which is the main challenge for rearranging books in an image. 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.

Challenges: Spine segmentation poses unique, non-trivial challenges. Book spines are typically located very close to each other; therefore, the outline edges of adjacent book spines are noisy, blended together, or in case of adjacent book spines with similar color, fragmented or nonexistent. Moreover, book spines are highly textured; hence, the edge map is dense and noisy, and the outline edges of the spines are further fragmented.

Most existing region based segmentation methods assume that objects are separated and surrounded

(a) (b)

Figure 1.1: The bookshelf reorganizer. a) The original image. b) The output: all books are upright and sorted by height.

by a smooth background; hence, they are expected to fail on this task, as demonstrated by our experimental results. For the same reason, active contour methods that initialize a contour considerably far from an object are expected to fail due to convergence to a set of adjacent spines rather than a single one. Object detection and recognition methods that learn object instances of a given class are also expected to fail due to the large variability of book spines.

Object detection is sometimes divided into two categories – "things and stuff", where "things" refers to objects with distinct outlines and shape, and "stuff" refers to objects with unclear boundaries, e.g., grass or sky. Although a bookshelf full of books could be considered as "stuff," methods for the segmentation of "stuff" do not segment the objects that compose it; therefore, this approach is not suitable for the segmentation of individual book spines.

Finally, since book spines may have different orientations, the *Manhattan world* assumption cannot be used, and orientation estimation methods must be considered (on which we elaborate in Chapter 3).

The outline of our method: We propose a two phase method. The first allows us to detect a set of candidate spines in a scene with cluttered background that consists of shapes similar to the object to be segmented. The first phase outputs a set of candidate spines with many false positive detections, but very few false negatives. In the second phase we use physical constraints to filter the detected set of

candidate spines.

Our spine segmentation method is a modification of the active contours paradigm that is tailored to our task. We use a predefined contour shape, perspective projection of a rectangle (PR), that expands from a small seed. The seed point is gradually expanded while each edge of the PR contour (1) independently searches for convergence, and (2) supports its adjacent edges in determining convergence. We assume that all book spines are coplanar or lie on parallel planes (identical 3D normals), but may have different orientations with respect to these planes. Thus, the edges of each spine are consistent with a pair of orthogonal vanishing points, all incident to the same vanishing line. However, they are not consistent with a single pair of orthogonal vanishing points as in the *Manhattan world* assumption. Hence, to constrain the shape of each spine in the image, we compute the pair of vanishing points that correspond to orthogonal lines in space.

Applications: To demonstrate the application in mind, we propose a reposition of the segmented book spines on a rendered, clean bookshelf. An example of reordering based on height is presented in Figure 1.1. Other possible applications, which we do not consider in this thesis, include using the segmented spines as input to an OCR system or to a book spine identification system (e.g., [7, 28, 35]).

Main contributions:

- The use of expanding seeds with predefined shape. This method allows us to segment objects in a scene with cluttered background, especially if the clutter consists of shapes similar to the object to segment. In our study we use it to (accurately) segment generally oriented book spines in an image taken from a general viewpoint.
- The use of two vanishing points that correspond to orthogonal directions in space, and their corresponding vanishing line. This allows us to avoid the *Manhattan world* assumption.
- The proposed application to tidy up an untidy scene automatically in the image.

Chapter 2

Related Work

Book spine segmentation: The objective of most book segmentation methods is to automate library related tasks, for example, managing an inventory of books. The majority of these methods [7, 28, 35, 8] segment book spines as a step towards book identification. The segmentation of book spines is performed by grouping adjacent long lines. Optical character recognition (OCR) methods [28, 35], or feature-based methods [7], are then used for book identification. Taira *et al.* [31] 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. In a different approach, the identification of book spines by the matching of color quantization indices was studied [21]. However, this approach is not useful in the task of book spine segmentation due to the low precision of the color quantization indices. None of these studies aim to obtain a precise segmentation of the spine, in particular to detect its upper part, which is essential to the proposed application.

Some of the algorithms mentioned above [31, 7, 28, 8] assume orthographic projection, and some also assume that all of the books have the same orientation [7, 28, 35, 8]; Quoc *et al.* [28] further assume that they are roughly aligned with the axis of the image. These assumptions are too restrictive for general bookshelf images that we consider in this thesis (e.g., Figure 3.7), and images of untidy bookshelves in particular. Furthermore, the performance of these algorithms is affected by book texture, since they rely on long line primitives, paired by their order. In particular, long line extraction, using

line-fitting [7, 28, 35], or using the Hough transform [31], often produces false positives in a cluttered, highly textured environment such as a bookshelf.

Other book related methods, which rely on complex setups or interactive interfaces, were suggested in recent years. An interactive book searching method, in which a user touches the shelf below a book spine for a query on this spine, was suggested by Matsushita *et al.* [23]. In [12], two cameras are used to recover the plane of book spines and constrain their location, and a projector is used to display information on queried books.

PR detection: Perspective rectangle detection algorithms were used for detecting windows and doors in offices and hallways, or to segment buildings in urban environments (e.g., [25, 30]). 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 often assume texture free scenes.

Active contours: Many active contour [14, 18] based methods were suggested in recent years, largely for the segmentation of medical images [24]. Shape priors were integrated into the active contours framework(s), e.g., [29, 9, 1]. In particular, Abufadel *et al.* [1] introduced a rectangle shaped contour, which undergoes rotation and scaling in the image plane, for the segmentation of inter-vertebral disks in the human spine. The algorithms mentioned above are not suitable for the segmentation of book spines, since the shape priors they introduced cannot undergo deformation that corresponds to perspective distortion. Moreover, some of the algorithms, e.g., [1], use Chan & Vese's [6] energy score, which is not suitable for book spine segmentation since it is based on the assumption that the object to segment is of (almost) constant color.

General segmentation: Although general segmentation algorithms, e.g., [3, 11, 13], assume almost nothing on the input image, they are not likely to achieve good results on an input of books on bookshelves. This is due to the absence of a critical and fundamental assumption – a book spine is always composed of straight lines. Therefore, even though some results could be considered satisfactory in terms of quantitative evaluation, none of the results (the segments) has the shape of a book spine.

Mishra *et al.* [26] also aim for a general purpose segmentation algorithm, and formulate the problem as the segmentation of objects which contain seed point. Thanks to their seed point centered approach, their method successfully avoids texture inside an object; however, it suffers from the same drawback mentioned above.

Book spine reorganization: The geometric rearrangement of objects in images is usually studied in the context of computer graphics. Several methods [27, 5, 10] have been suggested for object rearrangements, i.e., producing a new image in which an object is located somewhere other than its original location. Although visually pleasing results were obtained by these methods, they do not consider the perspective viewpoint of the scene, and when relocating objects in the scene do not consider the change in shape due to perspective distortion.

Keren *et al.* [19] presented a system for placing artificial objects in a single image of an architectural scene. To recover the extrinsic parameters of the camera, the user is requested to provide the location of the origin of the world in the image, and the 3D world distance between two points in the image. Then, to place an artificial object in the scene, the user is requested to click on a location in the image, and to specify the parameters of a plane on which the artificial object will be placed. Although the projection errors are almost unnoticeable, their method requires considerable input, and full 3D models for the objects to be placed in the scene. We show in Chapter 4 that planar objects can be repositioned in the scene, without any input from the user, except for a click on a target location in the image for the repositioned object. In our method, this click on the target is not required since we automatically deduce the target location of objects, based on the input reorganization criteria.

In summary, to the best of our knowledge, no method addressed the problem of precisely segmenting book spines under perspective projection, where the spines are arbitrarily oriented. Methods that address the spine segmentation problem, under the assumptions considered in this thesis (e.g., [3]) are expected to fail for the reasons specified above.

Chapter 3

Book Spine Segmentation

An image of books placed on bookshelves is the input to our method. 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 location of the PRs, their spatial relations, and their size. We assume that all book spines are parallel to a plane π. Hence, the corresponding vanishing points of their edges lie on a vanishing line ℓ. For the computation of ℓ, see Section 3.1.4.

3.1 Identifying PR Candidates

We present a method to identify a set of PRs that are candidates to segment book spines. The book spine segmentation is formulated as a minimization problem over five parameters that together define its location, orientation, and size.

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 Section 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 $\{D_i\}_{i=1}^{4}$, the distances from a given internal point s to each of the edges (see Figure 3.1a).

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 parameter α defines the pair of orthogonal vanishing points, v and $v_\perp$, as shown in Section 3.1.3. Note that for a PR, unlike the case for a rectangle not under perspective distortion, we use the orientation parameter α implicitly through the use of its corresponding two vanishing points v and $v_\perp$. 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 Figure 3.1b). Formally,

$$\begin{aligned} q_1 &= s - d_1\hat{u}(s, v_\perp), \\ q_2 &= s - d_2\hat{u}(s, v), \\ q_3 &= s + d_3\hat{u}(s, v_\perp), \\ q_4 &= s + d_4\hat{u}(s, v). \end{aligned} \tag{3.1}$$

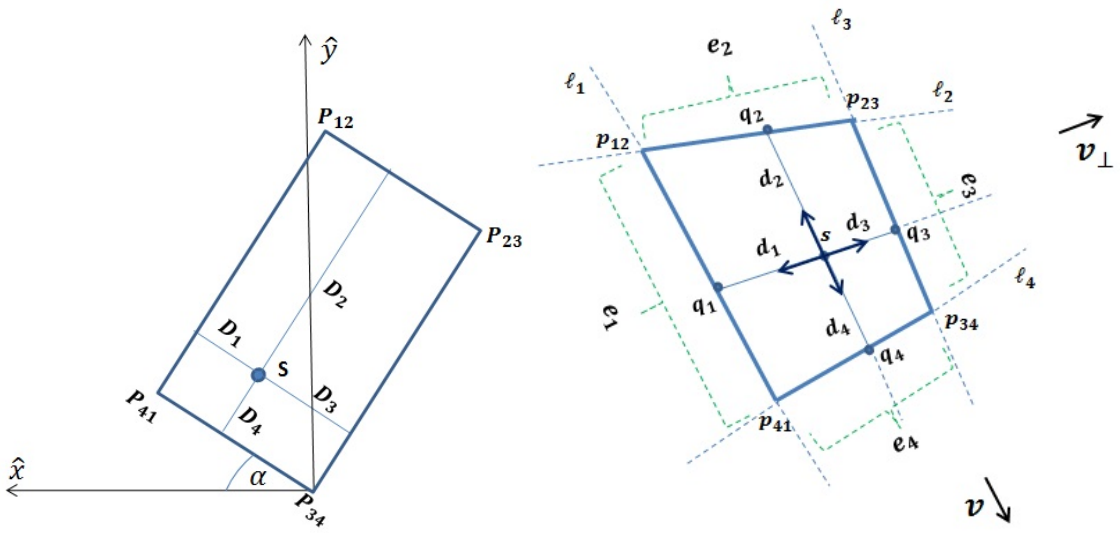
Let l_1 be a line in the image through q_1 and v, that is, $l_1 = q_1 \times v$, and l_2 be a line in the image through q_2 and $v_\perp$, that is, $l_2 = q_2 \times v_\perp$. A vertex p_{12} is defined by the intersection point between l_1 and l_2, hence its homogenous coordinates are given by:

$$\tilde{p}_{12}(d_1, d_2) = \tilde{l}_1 \times \tilde{l}_2. \tag{3.2}$$

The other vertices are defined in a similar way:

$$\tilde{p}_{23}(d_2, d_3) = \tilde{l}_2 \times \tilde{l}_3$$

$$\tilde{p}_{34}(d_3, d_4) = \tilde{l}_3 \times \tilde{l}_4$$



(a) (b)

Figure 3.1: Parametrization of a) a rectangle b) a PR

$$\tilde{p}_{41}(d_4, d_1) = \tilde{l}_4 \times \tilde{l}_1.$$

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 five parameters $D = \{d_i\}_{i=1}^4$ and α.

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 PR's consistency with the image gradients. When all d_i are inactive simultaneously, the process halts. The $d_i \in D$ are initially in active state, all set to 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 green segments in Figure 3.2). The inactive state is determined by both consistencies, as next defined.

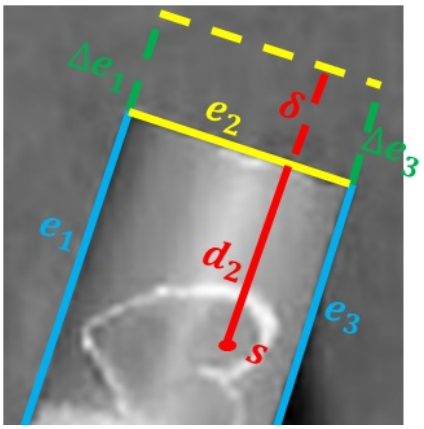


Figure 3.2: Edge extension segments are marked in green.

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,

$$E(e) = \frac{\sum_{p \in e} \arccos(\hat{\nabla} I(p) \cdot \hat{n}(e))}{|e|}, \tag{3.3}$$

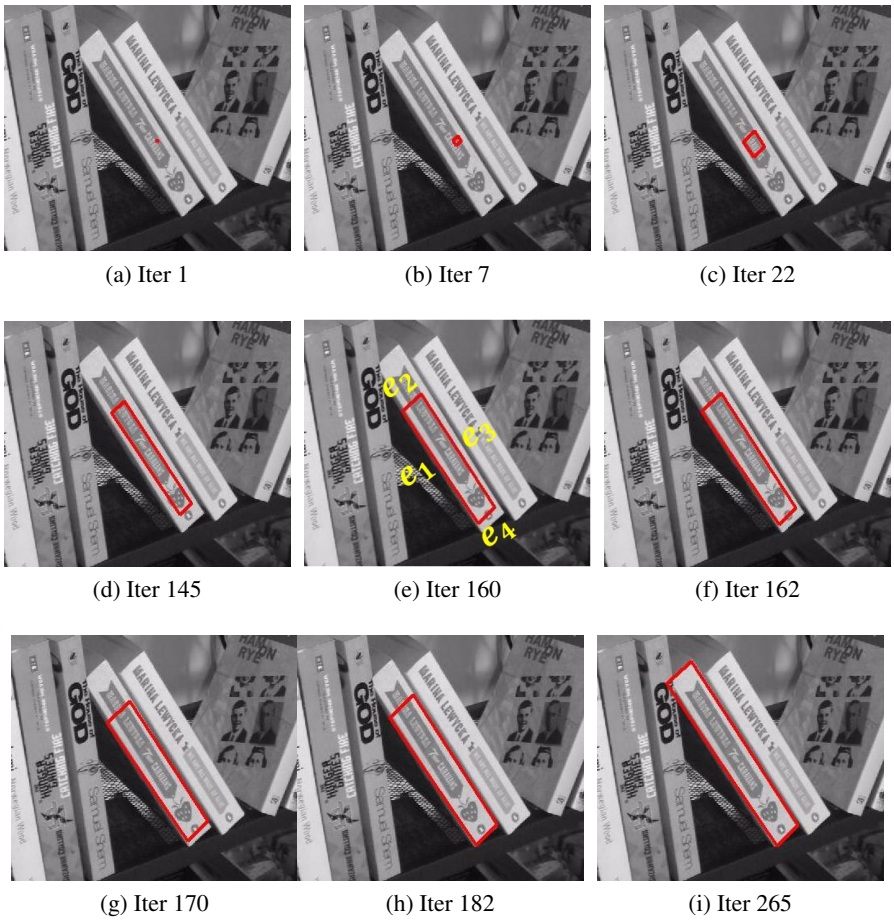
where $p \in e$ is a pixel on the edge e, $\hat{\nabla} I(p)$ is the gradient at p, and $|e|$ is the edge length measured in pixels.

For d_i to enter the inactive state, three conditions must hold. The first is that $E(e_i(D))$ is a local minimum as a function of d_i. The second is that $E(e_i(D)) < t_{\mathrm{G}}$, where t_{G} is a predefined threshold. The third is introduced to avoid halting at a local minimum by considering the support of the adjacent edges to increment d_i. To formulate this support, we consider the *edge extension segment* of an adjacent edge where d_i is incremented by δ. Formally, let e_j be an adjacent edge of e_i, its edge extension segment is defined by:

$$\Delta e_j(d_i, \delta) = [p_{ij}(d_i, d_j), p_{ij}(d_i + \delta, d_j)]. \tag{3.4}$$

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, $max\left(E(\Delta e_j), E\Delta e_k)\right) > \beta t_{\mathrm{G}}$, then d_i enters the inactive state.

After each round, an inactive d_i may return to the active state, if the consistency of e_i with the image gradient changes. This may occur when e_i lengthens due to the increase of d_j and d_k. A d_i returns to the active state when $E(e_i) < \lambda t_{\mathrm{G}}$, where $\lambda > 1$ is a constant defined to prevent frequent state



(a) Iter 1 (b) Iter 7 (c) Iter 22

(d) Iter 145 (e) Iter 160 (f) Iter 162

(g) Iter 170 (h) Iter 182 (i) Iter 265

Figure 3.3: PR expansion. Note that in (e) the edges are denoted with their names. (a) the seed point. (b) e_3 entered inactive mode. (c) e_1 entered inactive mode. (d) e_1 returned to active mode. (e) e_1 entered inactive mode. (f) e_3 returned to active mode. (g) e_3 entered inactive mode. (h) e_4 entered inactive mode. (i) e_2 entered inactive mode; the process halts.

changes. As a result, the search avoids convergence to textures inside the spines, e.g., due to text or a color patch within the spine (see iteration 22 in Figure 3.3). An expansion of a PR is illustrated in Figure 3.3.

At the end of each round, the consistency of the PR edges with the image gradients for various values of the parameter α are tested, and α_c, the current value of the parameter α, is updated accordingly. In our implementation we consider 10 values of α, equally distributed, in the range $[\alpha_c - 5, \alpha_c + 5]$ (all

values are in degrees). That is,

$$\alpha_c = \arg\min_{\alpha'} \left(\sum_{i=1}^{4} E(e_i(\alpha')) \right). \tag{3.5}$$

3.1.3 Orthogonal Pairs of Vanishing Points

To preserve a PR shape, a pair of orthogonal varnishing points is required for each orientation. We assume that all spines are on planes parallel to π; hence, all vanishing points are located on the vanishing line defined by the computed pair of vanishing points corresponding to the dominant orthogonal directions, v and $v_\perp$.

Using the internal calibration matrix, K (see computation below), 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 $\tilde{v} = M(\hat{u}\ \ 0)^T = KR\hat{u}$. We choose the world and the camera coordinate system to be identical, hence, $R = I$, and , $\tilde{v} = K\hat{u}$. 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 is given as a function of α by:

$$\begin{aligned} v(\alpha) &= K\hat{u} = KR(\hat{n}, \alpha)\hat{u}^0 = KR(\hat{n}, \alpha)K^{-1}v^0. \\ v_\perp(\alpha) &= KR(\hat{n}, \alpha)K^{-1}v^0_\perp. \end{aligned} \tag{3.6}$$

To compute the internal calibration matrix, K, we assume the camera has square pixels, no skew and the principle point is in the middle of the image. This assumption provides sufficient constraints on $K = diag(1/f^2, 1/f^2, 1)$ [16, p. 228] for a unique solution, using the additional constraint of the orthogonality of the vanishing points v^0 and $v^0_\perp$. The orthogonality of v^0 and $v^0_\perp$ yields

$$\begin{aligned} &(v^0\ \ f)(v^0_\perp\ \ f)^T = 0 \\ &f^2 = -v^0_x v^0_{\perp x} - v^0_y v^0_{\perp y}, \end{aligned}$$

where f denotes the focal length, $v^0 = (v^0_x\ \ v^0_y)$ and $v^0_\perp = (v^0_{\perp x}\ \ v^0_{\perp y})$.

To obtain an initial estimation of α for a given seed point, the dominant edge orientation in a $h \times w$ of its neighborhood (we use 50×50) is computed. Since an accurate initial estimation of α is critical, we use a high accuracy edge orientation estimation [32]. In contrast to common algorithms, which use a first order operation on the edge's location to estimate its orientation, [32] uses a third order operator, which is shown to be more accurate. 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 3D space 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.

3.1.4 Initialization

Orthogonal direction VPs detection: As a preprocessing step, we compute ℓ by detecting the two vanishing points, v^0 and $v^0_\perp$, that correspond to the dominant orthogonal directions in the image, using Tretyak $et\ al.$'s method [34]. In contrast to most methods, low level geometric primitives, e.g., edges, and high level geometric primitives, e.g., vanishing points, are estimated simultaneously. The problem is expressed as energy minimization of terms that correspond to edges, line segments, lines, vanishing points, the zenith and the horizon:

$$E_{total}(s,l,h,z|p) = \sum_{i=1..P} E_{edge}(p_i|s) + \sum_{i=1..S} E_{segment}(s_i|l) + \sum_{i=1..L} E_{line}(l_i|h,z) + \\ + \sum_{i \le i < j \le H} E_{horizon}(h_i,h_j|z) + E_{prior}(s,l,h), \tag{3.7}$$

where s corresponds to line segments, l to lines, h to horizontal vanishing points, z to the zenith and p to edge pixels. Each energy component, e.g., $E_{edge}(p_i|s)$, corresponds to a probability function that estimates the likelihood of an edge pixel p_i to belong to a non-clutter edge given the line segment s. Such a unified treatment reduces the uncertainty between the layers of the model, and allows greater robustness and accuracy than previous methods.

For example, the term, E_{edge}, is given by:

$$E_{edge}(p|s) = min(\theta_{bg}, min_{i=1..S}(\theta_{dist} d(p,s_i)) + \theta_{grad} d_{angle}(p,s_i)),$$

where $d(p, s_i)$ denotes the Euclidean distance in the image plane between the pixel p and the line segment s_i (the minimum distance), $d_{angle}(p, s_i)$ denotes the angular difference between the local edge direction at pixel p and the direction of line segment s_i, θ_{bg} is the constant, corresponding to the likelihood of the background clutter, and θ_{dist} and θ_{grad} are the constants corresponding to the spread of edge pixels generated by a particular line segment around that line segment.

This minimization problem is computationally hard and requires the use of approximations; hence, an approach of discretization is used to achieve a solution. To do so, known algorithms for extraction of edge points, line segments, lines and vanishing points are used with very low acceptance thresholds; thus, producing large sets of edge points, line segments, etc. Since the resulting set of possible solutions is very large, two observations are used to reduce its size. First, most edge points correspond to only one line segment, most line segments correspond to only one line, etc; hence, the space of possible solutions can be greatly reduced. Following that and secondly, the edge points are the majority of terms in the energy model and as stated above, mostly correspond to single and specific line segments; therefore, the terms that correspond to most edge points can be calculated only once for all solutions. Finally, these observations allow to use a simple and brute force optimization scheme.

The dominant orthogonal directions are assumed to be on π, since we assume that the image contains a sufficient number of books.

Seed points: A preprocessing step that allows initialization of s is the extraction of line segments using [20]. Line segments longer than a predefined threshold (we use 10% of the image height or width) are filtered. Seed points (we use 10) are placed at equal distances along each line segment on both its sides at a small perpendicular offset (we use 5 pixels); see e.g., Figures 3.7(b).

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 spines together (e.g., Figure 3.5(a)), sub-regions of a spine (e.g., Figure 3.5(b)), other objects in the scene, or simple noise. In addition, several PRs may segment almost the same region in the image. We present a heuristic algorithm to select the set of PRs

that most probably represent book spines. In this part of our method, we use the assembly of books in order to improve the segmentation of each spine.

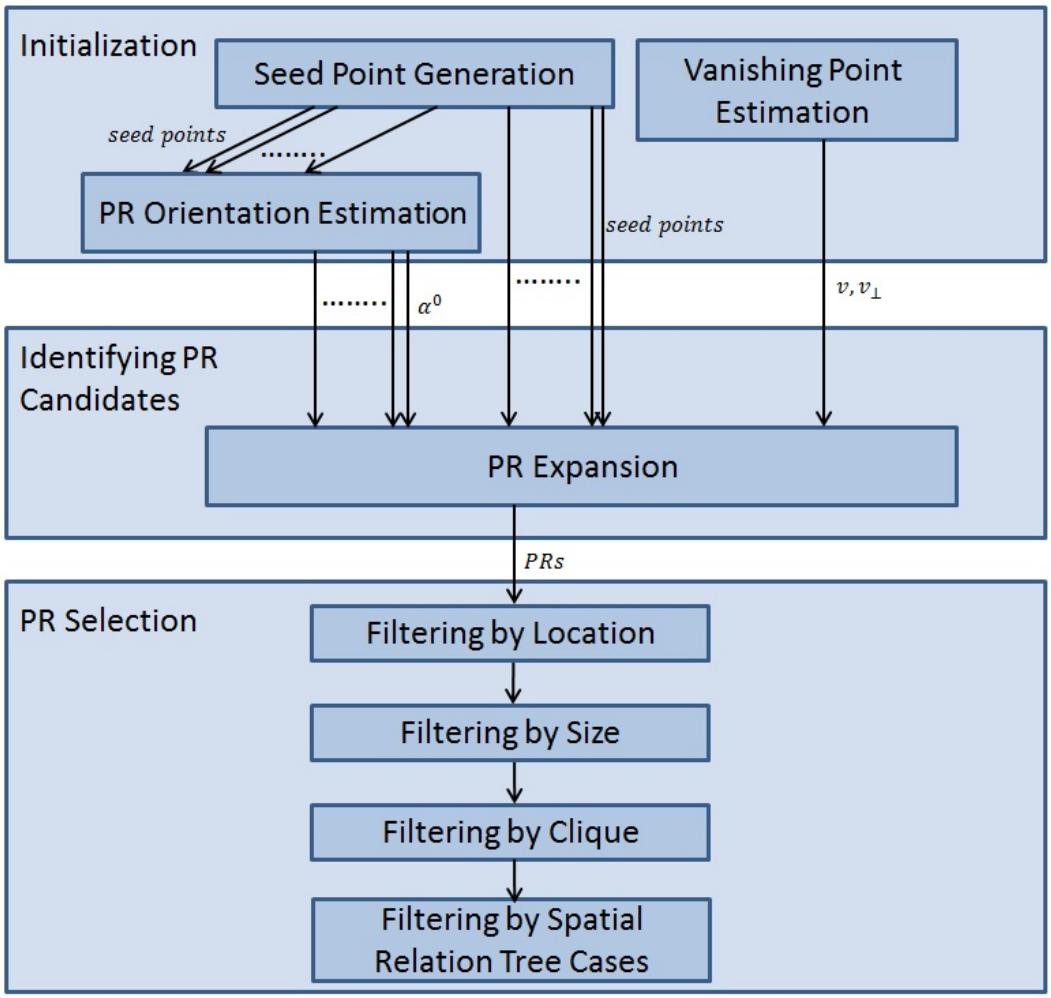


Figure 3.4: Book spine segmentation diagram.

3.2.1 Location and Size

Location: We assume that the detected book spines are either positioned on a bookshelf or supported by other books that are positioned on a bookshelf (e.g., a pile of books). All other PRs can be discarded. The direct detection of the bookshelf's edge, using the Hough transform or line detection algorithms, does not provide satisfying results (see Figure 5.3), due to occlusions caused by book spines and the abundance of lines detected in an image that contains a bookshelf. Therefore, we estimate the location

of the bookshelf edge by the random sample consensus algorithm (RANSAC) [15] with the lower vertices (p_{41}, p_{34}) of the PRs as input. Since we do not know *a priori* the number of shelves, we assume it is not more than n (in our implementation 10) and execute RANSAC n times. In each run, we exclude the inliers from previous runs. Finally, we assume that all bookshelves are parallel in the scene and RANSAC is used to filter those that do not agree with a single vanishing point. The set of bookshelves is used to recursively choose the set of PRs supported by either a bookshelf or by another book.

Size: To overcome the variations in size due to perspective, the PRs are normalized by using the perspective-aware repositioning of planar objects algorithm by Tolba *et al.* [33] (see Section 4). The result is the set of PRs as if they were all projections of spines taken from 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 1std below or 2std above the mean for both e_1 and e_4).

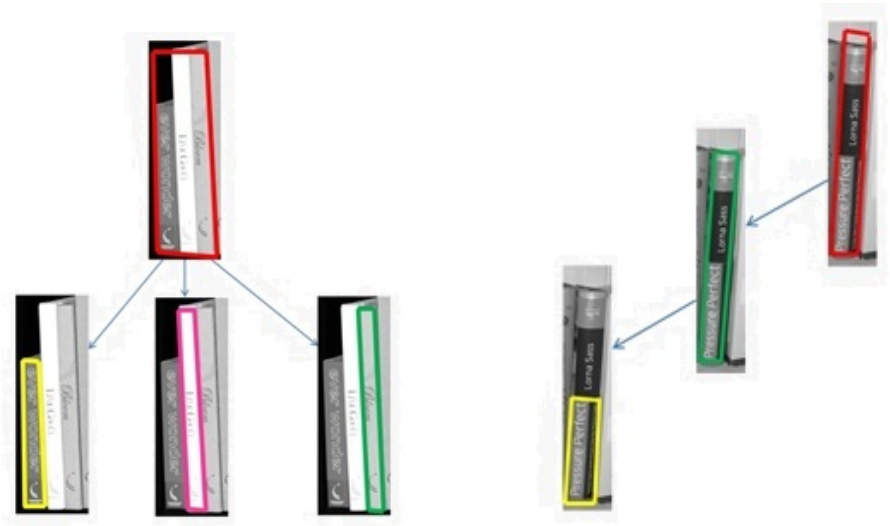
3.2.2 Spatial Relations

We define a graph $G = (V, E)$ to represent the spatial relations between PRs. A node $v_i \in V$ represents a PR, and a directed edge $(v_i, v_j) \in E$ represents a normalized rate of overlapping of at least γ of v_j with v_i. That is, $|R(v_j) \cap R(v_i)|/|R(v_j)| \geq \gamma$, where $R(v)$ is the region defined by the PR that corresponds to v and $|R(v)|$ is its area (in our implementation $\gamma = 0.75$).

Before we describe the heuristics used to obtain a forest from the graph (described below), we describe how it is used. Each tree consists of one or more PRs; some segment several spines together and some correspond to subregions of spines (Figure 3.5). The goal is to select from each tree a subset of nonoverlapping PRs that are most likely to represent a set of spines.

For simplicity, let us first consider a tree T of height 1. Denote by v_f and S_c the root and its children, respectively. To avoid overlapping, either v_f or a subset of its children should be selected. We consider the following three cases:

Horizontal Span: If a subset $U_C \subseteq S_C$ consists of adjacent non-intersecting PRs, and v_f covers this subset (e.g., Figure 3.5 (a)), v_f should be discarded. For this case, we require that at least one of the



(a) (b)

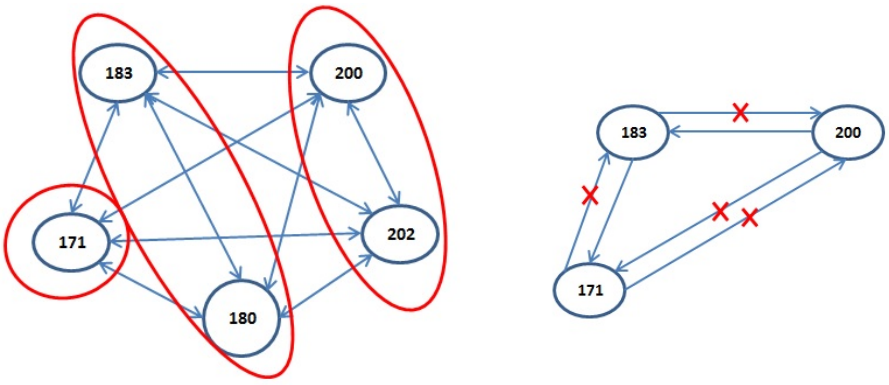
Figure 3.5: Spatial relation tree. a) horizontal span b) vertical oversegmentation and subregions.

spines in U_C have the *perspective height* of v_f, defined by the length of e_1 (up to a predefined allowed difference), and that their *perspective width*, defined by the length of e_4, sum to v_f's perspective width.

Vertical Oversegmentation: If a vertex $v_c \in S_c$ covers the bottom part of v_f (see Figure 3.5 (b)) and v_f is an oversegmentation, then v_c is chosen. To test that v_f is an oversegmentation, an approximation of the non-overlapping parts of the vertical edges is used since the bottom parts of the PRs are not necessarily perfectly aligned. Formally, let e_i^{Δ} be the non-overlapping part of e_i. Then, v_c is chosen, if either $e_1(v_f)$ or $e_3(v_f)$ are unsupported by the image gradients. That is, $max\left(E(e_1^{\Delta}), E(e_3^{\Delta})\right) > t_\delta$, for a predefined threshold t_δ.

Otherwise: When the first two cases do not hold, v_f is chosen. This is the default behavior, which ensures propagation up the tree until a case of *horizontal span* or *vertical oversegmentation* is encountered.

Finally, the general case of a tree T of height > 1 is handled bottom-up, recursively. In each iteration, T is handled as k distinct trees, where each such tree is composed of sibling leaves and their parent. Each such tree is handled as tree of height $= 1$. The root is either replaced by its previously chosen descendants, or all previously chosen descendants are removed.



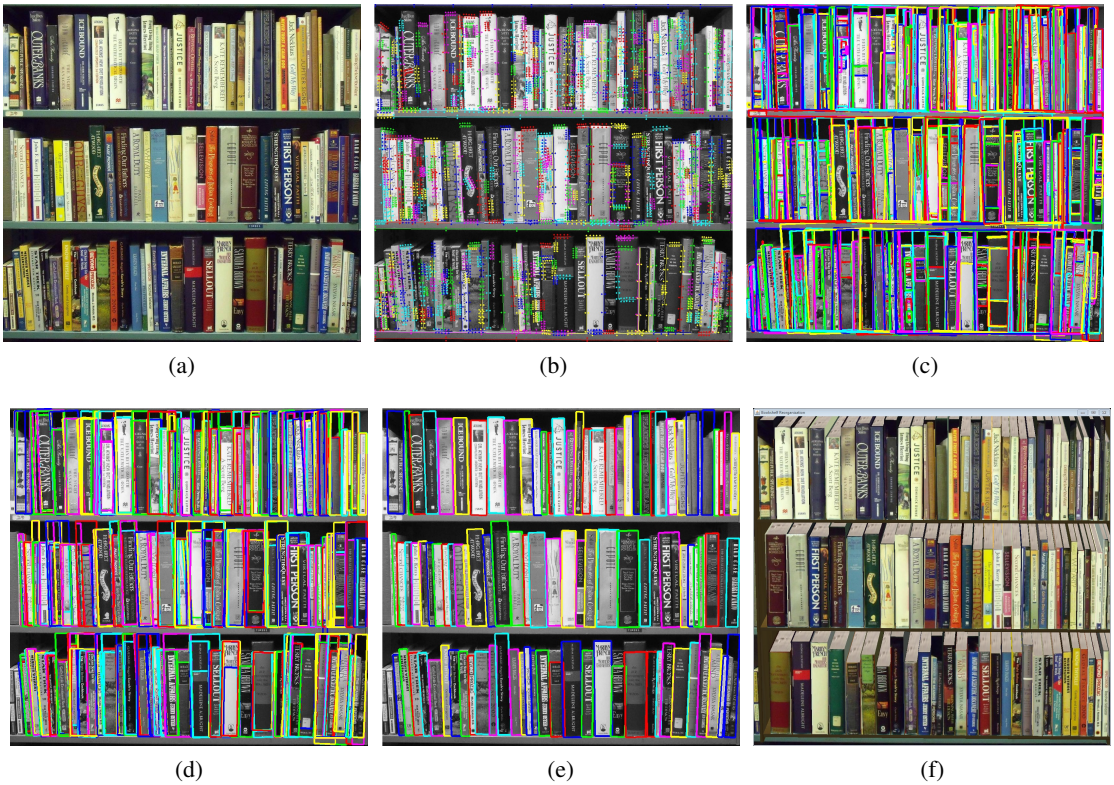
(a) (b)

Figure 3.6: The clique constraint. The numbers inside the nodes represent the height of the PRs. a) "Sub-cliques". b) Clique after choosing representatives and discarding edges.

Spatial graph to spatial forest: To allow us to apply our spatial relation selection cases (described above), the graph is first transformed into a set of directed acyclic graphs (DAG) and then into a forest, a set of directed trees. To transform a DAG into a tree (i.e., remove "shortcuts"), the transitive reduction for DAGs algorithm is applied [2]. Prior to that, to transform the graph into a set of DAGs, the cliques in the graph are transformed into trees. A clique in this graph relates to a set of similar PRs, with slightly different heights and widths. We found that the differences of height – in contrast to those of width – greatly influence the correctness of the segmentation. Therefore, the PRs in the clique are clustered into "sub-cliques" by applying hierarchical clustering (see Figure 3.6a) with the heights of the PRs as input (with a cutoff distance of 5% of the shortest PR in the clique). Then, one representative is arbitrarily chosen from each "sub-clique", and the rest are discarded; thus, a smaller clique remains (see Figure 3.6b). The remaining clique is transformed into a "chain" from longest PR to shortest, i.e., a tree whose root contains the highest PR and whose child is the next highest, etc (see Figure 3.6b). After the transformation, any edges from PRs outside the transformed clique to the PRs in the transformed clique are reconnected only to its root (i.e., the highest PR). Finally, the graph consists of a forest F, a set of directed trees.

In summary, we generate an initial set of PRs using seed points that are placed near long lines in the image. Each seed point is expanded until all four edges meet image gradients that correspond to lines in the image. We further use physical cues (location, size and spatial relations) to define a

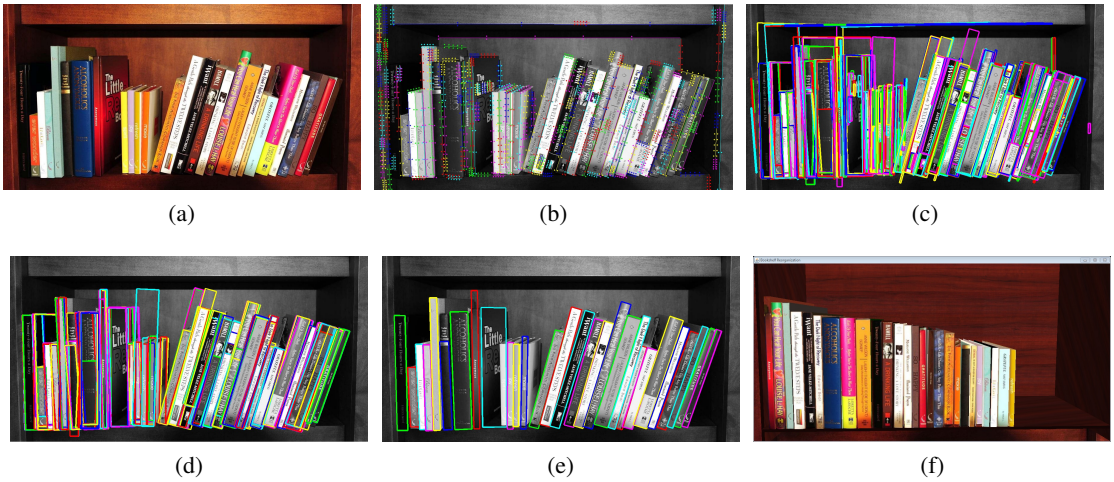
feasible subset of PRs, derived from the initial set of PRs. We first use the smallest distance from the estimated shelf lines to determine which PRs lack the support of either the shelf or other PRs (i.e., "floating PRs"). Then, relative size measures are gathered and used to statistically rule out PRs with unlikely size. Finally, to allow only minor overlap between PRs, redundant PRs (very similar PRs) and significantly overlapping PRs are filtered by introducing heuristic selection rules. A diagram and pseudocode of the algorithm are presented in Figure 4.1 and Algorithm 1 respectively.



(a) (b) (c)

(d) (e) (f)

Figure 3.7: Book spine segmentation steps. a) Original image, b) seed point and lines, c) PR candidates, d) PR candidate after location and size filtering e) final segmentation results. f) reorganization results (by width).



(a) (b) (c)

(d) (e) (f)

Figure 3.8: Book spine segmentation steps. a) Original image, b) seed point and lines, c) PR candidates, d) PR candidate after location and size filtering e) final segmentation results. f) reorganization results (by height).

Algorithm 1 Book Spine Segmentation Algorithm

% Initialization (3.1.4)
compute v and $v_\perp$
S=seed points from edges
% PR expansion (3.1.2)
for $s \in S$ **do**
  $d_i = 1$, $active(d_i) = 1$
  while $\sum active(d_i) > 0$ & $\#Iterations < th$ **do**
    for $i \in \{1..4\}$ **do**
      if $active(d_i)$ **then**
        % Note: $ismin(E(e_i))$ is true when $E(e_i)$ is at minima as a function of d_i
        if $E(e_i) < t_G$ & $ismin(E(e_i))$ & $\max(E(\Delta e_{i-1}), E(\Delta e_{i+1})) > \beta t_G$ **then**
          $active(d_i) = 0$
        else
          d_i++
        end if
      end if
      if $active(d_i) = 0$ **then**
        if $E(e_i) > \lambda t_G$ **then**
          $active(d_i) = 1$
        end if
      end if
      update $\alpha_c = \arg\min_{\alpha'} \left(\sum_{i=1}^{4} E(e_i(\alpha')) \right)$
      update v and $v_\perp$ according to α_c
    end for
  end while
  if $\#Iterations < th$ **then**
    store PR into setPRs
  end if
end for
% PR selection (3.2)
filter setPRs by size
detect bookshelves; filter by location
G=Generate Graph
F=choose one PR for each clique from G
for $T \in F$ **do**
  recursively (from leaves to root) filter by spatial locations
end for

Chapter 4

The Bookshelf Reorganizer Application

The goal of our method is to reorganize books on their shelves. Organizing books on bookshelves is a tedious chore, especially if the bookshelf contains hundreds of books. Often, we would like to try a few arrangements before deciding on the right one. Taking a picture of our bookshelf particularly for this reason, and using the bookshelf reorganizer application, requires considerably less effort, and saves time. Moreover, often among picture collections, one can find a few pictures of rooms with bookshelves in them, or pictures of bookshelves; our application can be applied here too, for the purpose of image aesthetics.

The only input to the bookshelf reorganizer is an image of books on a bookshelf and a (user-chosen) criterion (e.g., book height or color) by which the book spines will be reorganized. To overcome segmentation errors, the application may also be in interactive mode and display the segmentation results to the user. In this case, the user can delete and add PRs. The latter can be done manually or by choosing PRs from a visualization of the tree T (defined in Section 3.2.2).

The books can be reorganized by changing their orientation if they are not aligned and perpendicular to the shelf. The books can also be moved around, according to some criterion such as height, width, title or author (if OCR is applicable). Such reorganization schemes are useful when the goal is image aesthetics, or to suggest a new ordering of the books to the user. The objective here is to reorganize books while preserving their correct size and shape with respect to the perspective view of the scene. Note that preserving the size is important for suggesting rearrangement of bookshelves, in particular

when moving book spines between shelves to verify that there will be enough space to accommodate them.

To do so, we apply Tolba $et\ al.$'s [33] method for perspective-aware repositioning of planar objects to each of the spines.

Translation: Let $\hat{n} \cdot \vec{P} = d$ be the equation of a plane with normal $\hat{n}$ and distance d to the center of projection (COP). A family of homographies that correspond projections of points under translation on this 3D plane is given by:

$$H \simeq I + \frac{\delta}{d}\hat{t}\hat{n}^T, \tag{4.1}$$

where $a \simeq b$ denotes that $a = \lambda b$ where λ is an arbitrary scale factor, I is the identity matrix, $\hat{t}$ is the translation direction and δ is the translation distance, all in 3D space. Since in a single image setup there is no knowledge of the distance d, or the translation distance δ, we deduce the ratio $\beta = \delta/d$ instead:

$$H \simeq I + \beta\hat{t}\hat{n}^T. \tag{4.2}$$

The ratio β is computed using a projection of a point m on the 3D plane before translation, and the desired location m' for the projection of this point after translation. The following derivation is illustrated geometrically in Figure 4.1 for convenience. The ratio β is given by:

$$\begin{aligned} \beta &= \pm\frac{||m' - \lambda m||}{||\lambda m \cdot \hat{n}||} (= \pm\tfrac{\delta}{d}) \\ sign(\beta) &= sign(\hat{t} \cdot (m' - \lambda m)), \end{aligned} \tag{4.3}$$

where λ is the scale factor that needs to be determined. Using the translation homography (Equation 4.2) applied to point m to get point m':

$$m' = \lambda(I + \beta t n^T)m = \lambda m + \lambda\beta t n^T m. \tag{4.4}$$

First, we eliminate β from Equation 4.4 by taking the cross product of t:

$$t \times m' = \lambda t \times m.$$

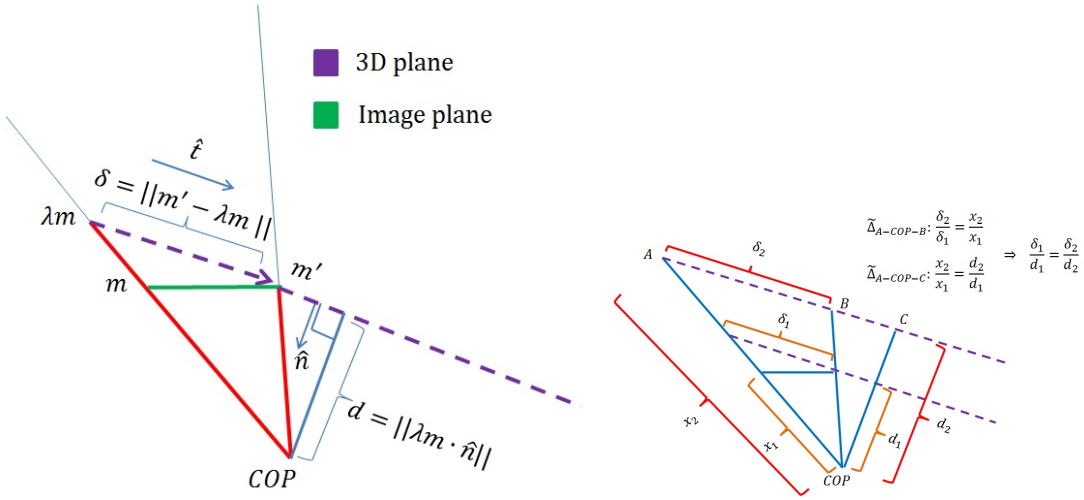


Figure 4.1: (a) Illustration of ratio β of Equiation 4.2 using a setup in which point m' and its projection are identical in 3D. (b) Justification and proof using triangle similarity, for the specific setup.

This is equation of the form $a = \lambda b$ for which the solution is:

$$\lambda = \pm \frac{||t \times m'||}{||t \times m||}$$
$$sign(\lambda) = sign((t \times m')(t \times m)).$$

Finally, one can substitute λ in Equation 4.3 to get the ratio β.

Rotation: The image of a plane under rotation about a pivot p is calculated in 2 steps. First, the plane is rotated about the normal to the plane, $\hat{n}$, with the origin of the world (COP) as a pivot. As a result, the pivot p is transformed to an intermediate location p'. To rectify this, the previously mentioned translation homography 4.1 is applied with $t \simeq p - p'$ to bring the plane back to its original location.

To employ Tolba *et al.*'s [33] method, we provide the following input: the normal to the plane (of spines) is $\hat{n}$ (see Section 3.1.3); this is also the rotation axis, since we consider book rotation only in the spine's plane. The rotation angle and translation vector are determined using the user-chosen reposition criterion and the bookshelves' location. This is done, for example, by first rotating all books to an upright position, sorting them by height, computing the translation for the longest one, and the translations for the rest accordingly. To this end, we use OpenGL to render a clean bookshelf, position

the spines as described above, and render an additional synthetic dimension for the books (the pages). Their direction is given by $\hat{n}$, and their length is arbitrarily chosen. We demonstrate reorganization results in the next chapter (see examples in Figure 5.4 to Figure 5.16).

Chapter 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 (e.g., Figures 5.5 and 5.12). The second contained 27 images (1235 book spines), of arbitrary oriented book spines and multiple shelves, under perspective projection view (e.g., Figures 5.4 and 5.7).

Segmentation and Reorganization Examples: A few segmentation results are presented in Figures 5.4c to 5.16c. 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., multiple spines segmented as a single spine, an overly extended spine, and a partially segmented book spine, are marked in Figures 5.5c to 5.7c. Some of these failures are due to the choice of parameters, as we discuss in the "Parameters" paragraph below.

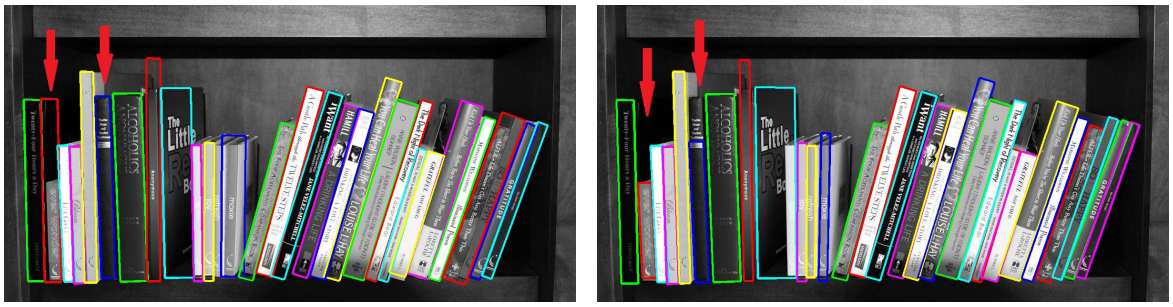
Quantitative Evaluation: For quantitative evaluation, the ground truth segmentation was manually generated. We considered only fully visible book spines and discarded those that intersect with image boundaries. We used the *Hoover Index (HI)* [17], in which the computed segments are labeled as Correct Detection (CD), Oversegmentation (OS), Undersegmentation (US), Missed (M) or Noise (N). 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, $min\left(|R(c) \cap R(g)|/|R(c)|, |R(g) \cap R(c)|/|R(g)|\right) \geq \gamma$,

	HI		SC		PRI		VI	
	Ours	**Arbelaez[3] (ODS/OIS)**	**Ours**	**Arbelaez[3] (ODS/OIS)**	**Ours**	**Arbelaez [3]**	**Ours**	**Arbelaez [3]**
Set 1	83.87%	32.68% / 34.72%	77.12%	47.81% / 51.54%	91.62%	86.91%	1.3782	2.1523
Set 2	71.47%	17.59% / 19.32%	78.53%	49.11% / 56.68%	84.02%	72.03%	1.3973	2.4090

Table 5.1: Segmentation results of our method and the OWT-UCM (Arbelaez *et al.*) method. The thresholds used for ODS under HI criterion are 0.3 and 0.2 for dataset1 and dataset 2, respectively, and under SC criterion, 0.15 and 0.05 for dataset1 and dataset2, respectively.

where γ is a predefined threshold (in our implementation we use $\gamma = 0.8$). The precision and recall were calculated using only the correct detections and combined as their harmonic mean. Dataset 1 reached a precision-recall of 83.87% (precision 80.74%, recall 87.27%) and dataset 2 reached a precision-recall of 71.47% (precision 70.44%, recall 72.55%).

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 (95.10% in dataset 1 and 82.19% in dataset 2) and, as expected, a low precision rate ($\sim$38%). The goal of the second phase is to increase the precision rate and produce a "legal" segmentation (without repetitions and significant overlapping), while preserving most of the recall rate. The contribution to the increase of precision of the different constraints in the second phase is presented in Table 5.2. The location and size (Section 3.2.1), as well as cliques and selection cases in the spatial relation tree (Section 3.2.2), were considered. According to our analysis, the contribution to the increase in precision is mostly due to the size and selection cases in the spatial relation tree. Note that the decrease in precision due to the clique constraint is expected



$t_G = 17°$ $t_G = 20°$

Figure 5.1: Different segmentation results with different parameters. The red arrows indicate segmentation errors.

	Precision					Recall				
	Initial	**Location**	**Size**	**Clique**	**Tree**	**Initial**	**Location**	**Size**	**Clique**	**Tree**
Set1	38.04%	41.63%	64.99%	42.54%	80.74%	95.10%	95.10%	94.49%	93.29%	87.27%
Set2	38.06%	43.88%	67.24%	44.54%	70.44%	82.19%	82.10%	80.97%	80.16%	72.55%

Table 5.2: Effectiveness analysis of the second phase. The parameters used for this analysis (as well as the results and generated images in this thesis) : $t_G = 17°$, size filtering above the mean $= 2.5\sigma$, and below $= 1\sigma$.

since we keep only one PR of every clique and most cliques contain true positive PRs, thus lowering the ratio of true positive PRs to other PRs. As a result of the increase in precision, the recall was affected mostly by the selection rules in the spatial relation tree ($\sim 7\%$ decrease of recall).

Parameters: For quantitative evaluation, described below, the same set of parameters was used in all experiments, despite the large variability in image resolution, number of books, etc. The general performance was not sensitive to parameter changes in the following ranges: t_G – $[15°, 23°]$, and the parameters for filtering by size – [1std, 3std] and [0.5std, 1.5std] 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 Figure 5.1. When tuning the parameters, two rules of thumb help reach the desirable results: as t_G increases, the edges of a PR are more likely to enter into the inactive mode. Increasing t_G can be beneficial, for example, when too many book spines are misdetected in the first phase due to weak edges. As the size filtering parameters increase, i.e., the filtering margin around the mean increases, fewer PRs are discarded. Increasing the size filtering parameters can be beneficial, for example, when very large or very small book spines are misdetected as a result of a large variance in size.

Comparison: Previous book spine recognition methods ([7, 28, 35, 31]) use assumptions that are not applicable to our data (see our discussion in Chapter 2). Furthermore, their goal is to identify book spines for the automation of library related tasks, which does not require precise book spine segmentation, contrary to our goal. Hence, we could not apply their methods to our data for comparison.

We 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) [4]: the

Figure 5.2: Segmentation results of the OWT-UCM method by Arbelaez *et al*. [3]. Our results for the same images are presented in Figures 5.4 to 5.16.

OWT-UCM segmentation algorithm suggested by Arbelaez *et al*. [3]. We used the available code from the Web. An example of a result (Figure 5.2) shows that the the OWT-UCM segmentation method does not perform well on the task of book spine segmentation.

A quantitative comparison of our and the OWT-UCM methods is presented in table 5.1. Our method significantly outperforms the OWT-UCM algorithm in every evaluation criterion, e.g., we reach 83.87% 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. In addition to the Hoover Index, we also used the evaluation schemes used in the BSDS500: *segmentation covering (SC)*, *variation of information*

(VI), and *probabilistic Rand index (PRI)* [3]. Segmentation covering is a continuous measure of the maximum area covered by the computed segmentation for each segment in the ground truth. Variation of information is a measure for 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 belong 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)*, which sets a uniform scale for all images and its *optimal image scale (OIS)*, which sets different and optimal scales for each image. To conclude, we demonstrated that a general segmentation method is not applicable for book spine segmentation. This is due to the absence of a critical and fundamental assumption – a book spine is always composed of straight lines.

(a) (b)

Figure 5.3: Bookshelf line detection results. a) Our method. b) Long line extraction.

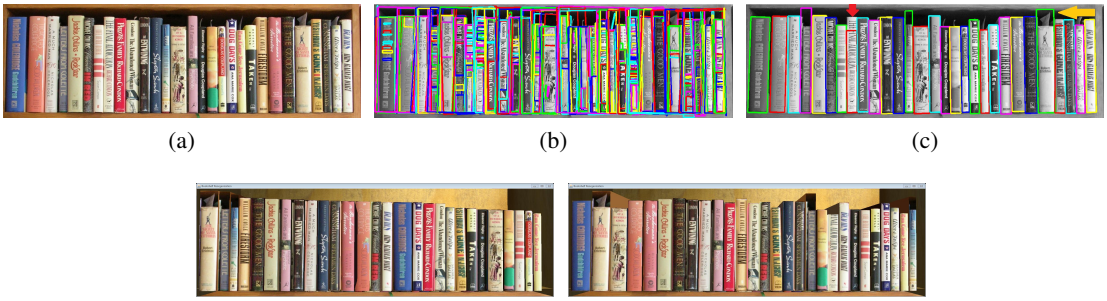
Bookshelf Line Detection: We next evaluate our algorithm for bookshelf line detection. The results were evaluated manually by us, i.e., by manually examining the detected shelves and classifying them into 3 categories: true positive, false positive and false negative. Here, the recall is more important than the precision, because of its impact on the end results of the book spine segmentation. That is, 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, and/or misaligned book spines, which do not form straight lines.

As a comparison to our bookshelf line detection method, line segments longer than 30% of the image height or width (the shortest) were extracted using [20]. Nonhorizontal line segments (up to 30 degrees difference) and, as in our method, line segments whose corresponding line is not incident to a common intersection point (up to a small difference), were discarded. Some results of the two methods are presented in Figure 5.3. It can be seen that our method significantly outperforms the line extraction method, since the latter often produces false positives and misdetects shelf lines.

(a) (b) (c)

(d) (e) (f)

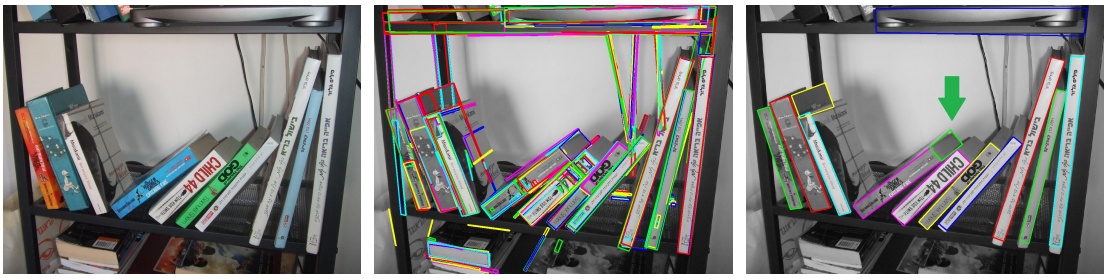
Figure 5.4: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height per shelf, e) reorganization by height for all shelves together, f) reorganization by width per shelf.



(a) (b) (c)

(d) (e)

Figure 5.5: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height, e) reorganization by width.



(a) (b) (c)

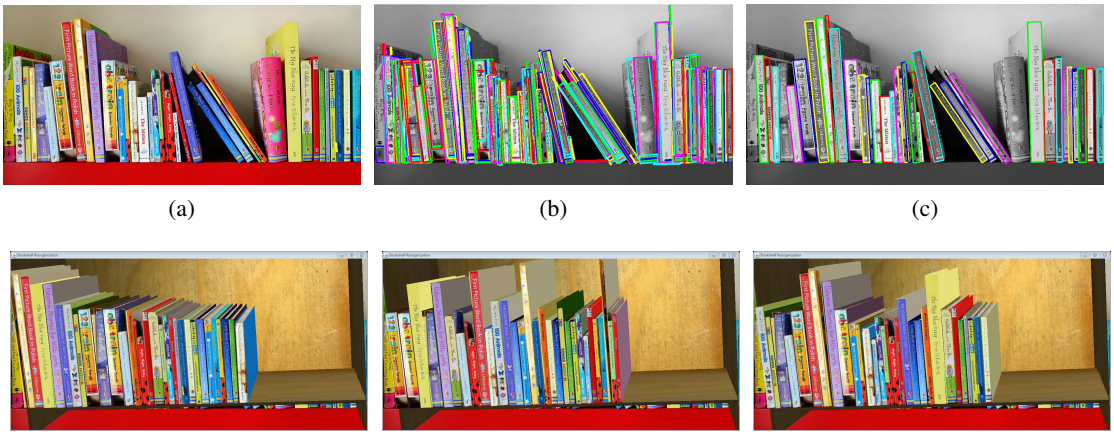
(d) (e) (f)

Figure 5.6: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height per shelf, e) reorganization by width per shelf, f) reorganization by only aligning the spines.

(a) (b) (c)

(d) (e) (f)

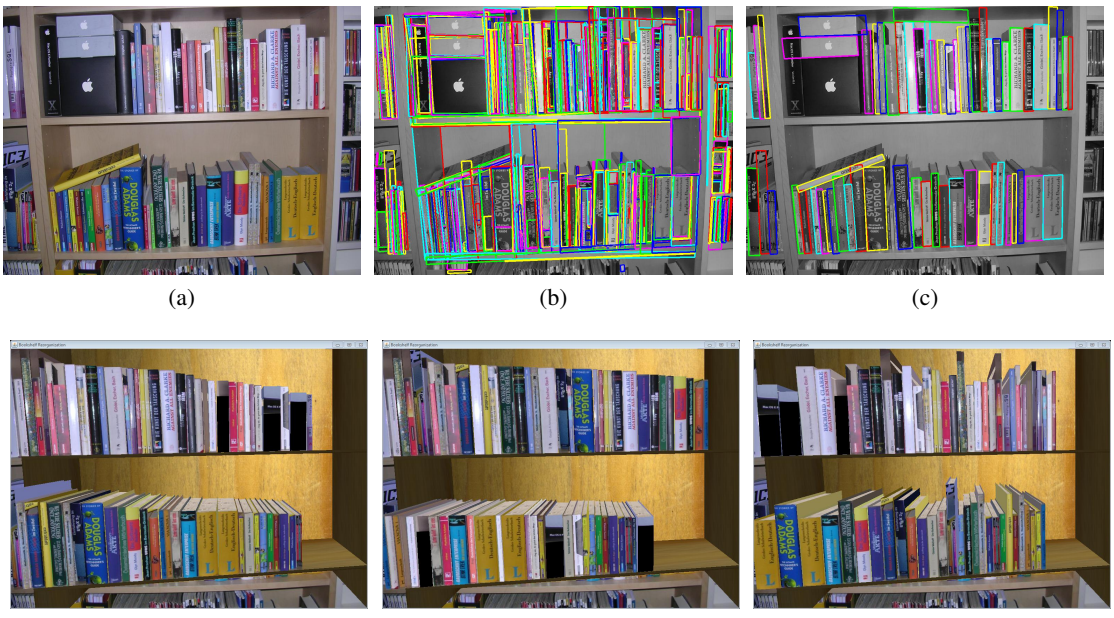
Figure 5.7: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height per shelf, e) reorganization by height for all shelves together, f) reorganization by width per shelf.



(a) (b) (c)

(d) (e) (f)

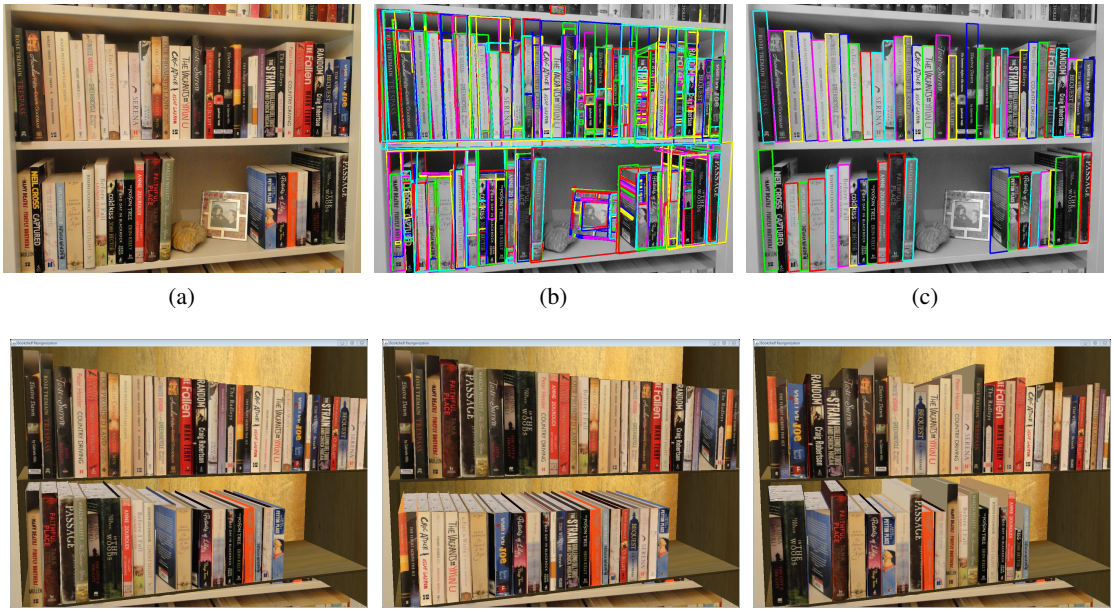
Figure 5.8: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height per shelf, e) reorganization by width per shelf, f) reorganization by only aligning the spines.



(a) (b) (c)

(d) (e) (f)

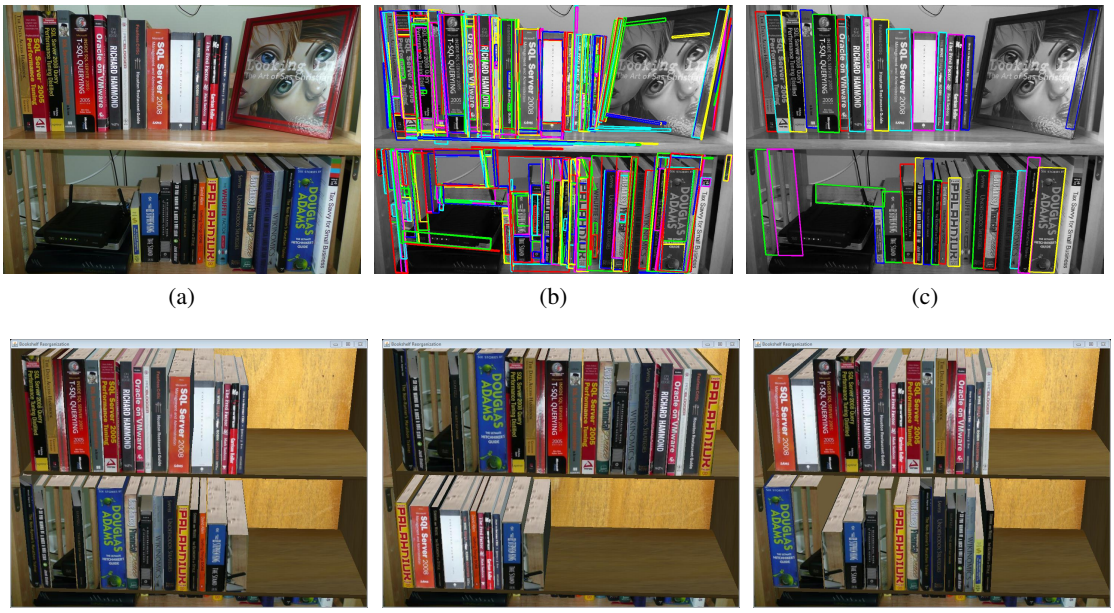
Figure 5.9: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height per shelf, e) reorganization by height for all shelves together, f) reorganization by width per shelf.



(a) (b) (c)

(d) (e) (f)

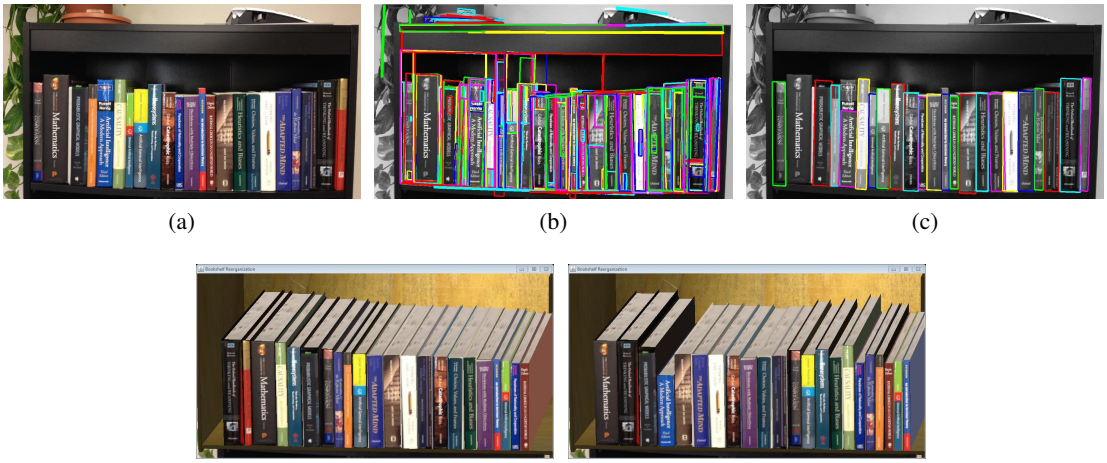
Figure 5.10: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height per shelf, e) reorganization by height for all shelves together, f) reorganization by width per shelf.



(a) (b) (c)

(d) (e) (f)

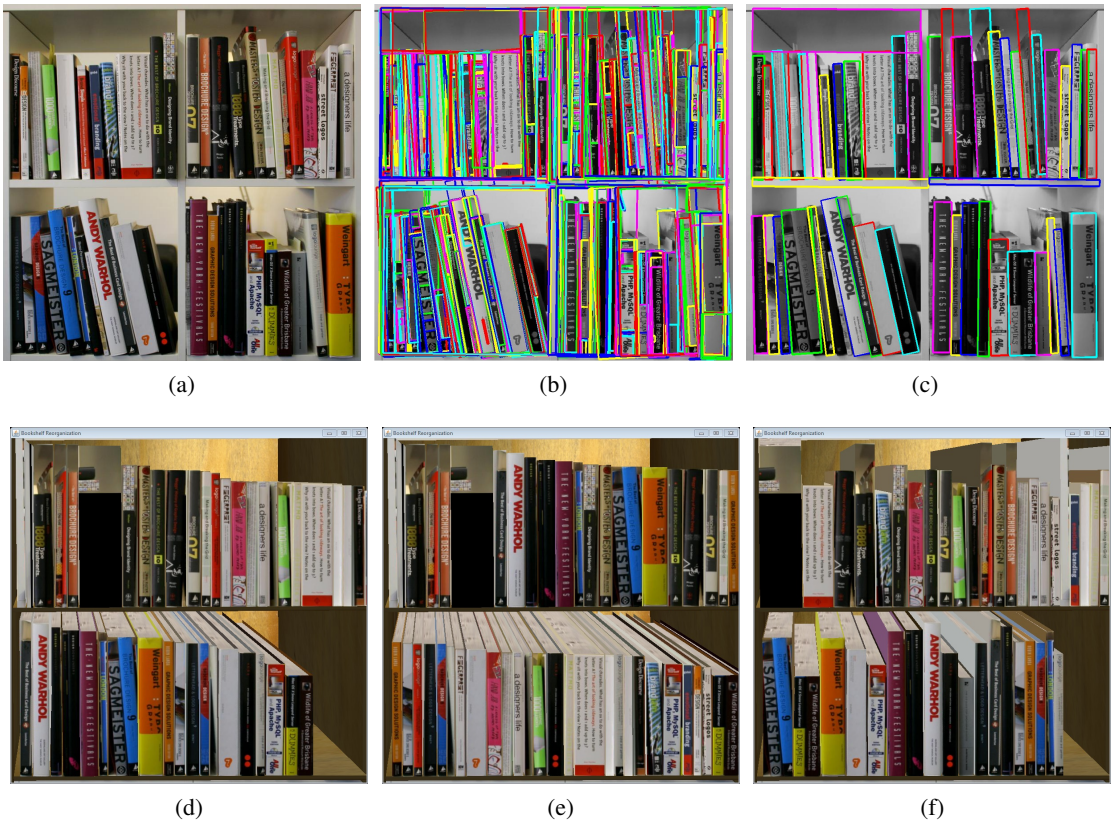
Figure 5.11: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height per shelf, e) reorganization by height for all shelves together, f) reorganization by width per shelf.



(a) (b) (c)

(d) (e)

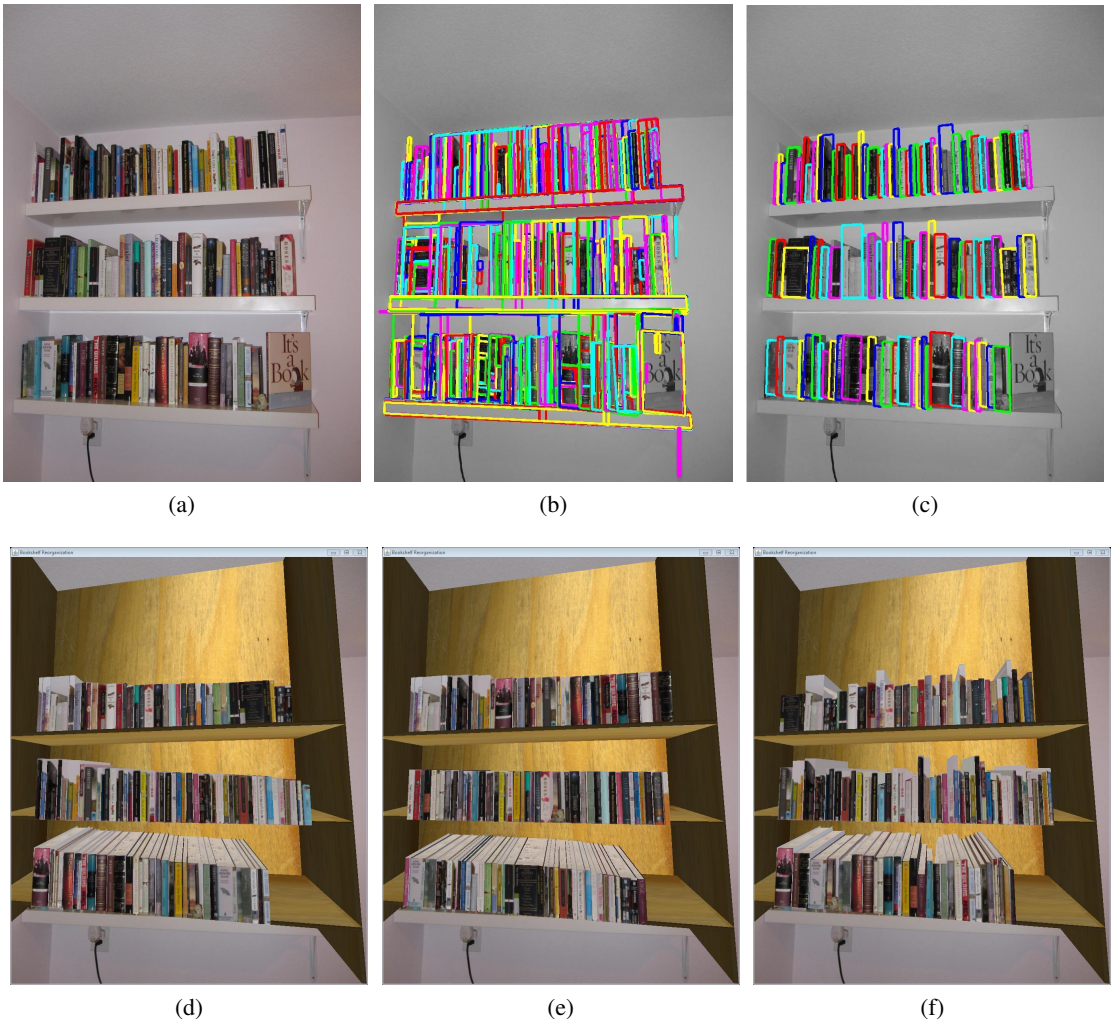
Figure 5.12: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height, e) reorganization by width.



(a) (b) (c)

(d) (e) (f)

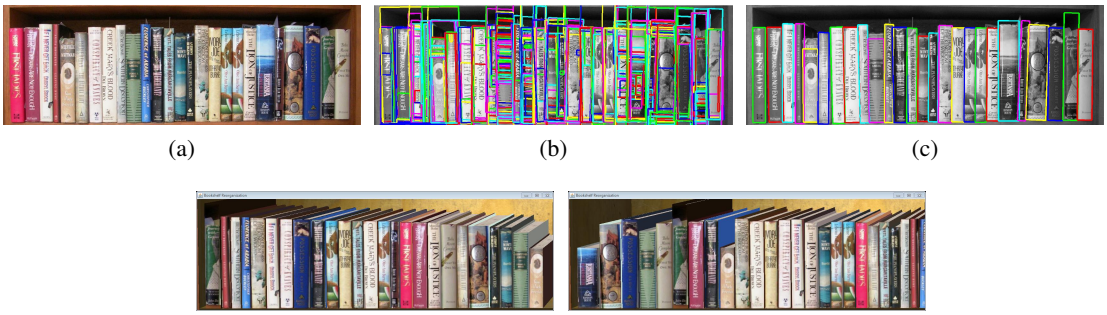
Figure 5.13: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height per shelf, e) reorganization by height for all shelves together, f) reorganization by width per shelf.



(a) (b) (c)

(d) (e) (f)

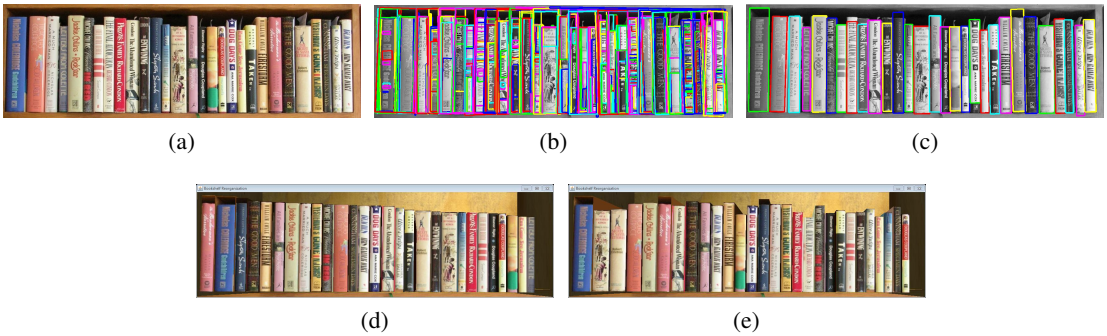
Figure 5.14: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height per shelf, e) reorganization by height for all shelves together, f) reorganization by width per shelf.



(a) (b) (c)

(d) (e)

Figure 5.15: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height, e) reorganization by width.



(a) (b) (c)

(d) (e)

Figure 5.16: Book spine segmentation and reorganization results. a) Original image, b) PR candidates, c) final segmentation results, d) reorganization by height, e) reorganization by width.

Chapter 6

Conclusion and Future Work

We proposed a solution to the challenging problem of book spine segmentation. The two phase solution allows us to obtain a large set of spine candidates using a bottom-up computation, and then use a top-down method to filter the results. The high recall of the first phase is obtained by using the PR as a basic template, which is able to deform in ways that correspond to projections of 3D rectangles that lie on a specific plane. This PR model allows us to use each individual edge of the PR to both detect book spine edges in the image and support the adjacent edges in their search. In the "traditional" approach to quadrilateral detection (e.g., [7, 28, 35, 25, 30]), lines are extracted and assembled to quadrilaterals. Although such formulation has the advantage of forming quadrilaterals globally (instead of locally), in a textured and dense scene it lacks reliable low level data, i.e., it uses lines instead of PRs. The success of the second phase comes from enforcing real physical constraints on the set of PR candidates. Because we use the bookshelf lines to discard all PRs that are not supported by the shelf, and the size and spatial constraints to discard PRs with unlikely size and PR repetitions, we are left with a small set of PRs with few false positives and, due to the first step, few false negatives as well.

Future Work As short-term future research, it is worth examining the possibility of adapting our method to similar segmentation problems of projections of rectangles, e.g., windows, pictures on walls, and building segmentation. For this purpose, a few changes in our assumptions are required:

- The assumption of different oriented PRs can be dropped, since most scenarios contain objects

of a *Manhattan world* scene (e.g., windows).

- Our method considers only one plane, on which all PRs lie. This needs to be generalized to multiple nonparallel planes (e.g., to three mutually orthogonal planes, in the case of *Manhattan world* scenes).

- The generation of seed points was designed to overcome the high density of books on a shelf. In some cases, fewer seed points would suffice, and they could be generated more accurately. For example, when segmenting pictures on walls, one can avoid generating seed points on walls (i.e., by assuming that a uniform texture-less surface is a wall).

- The physical assumptions that are used in our method should be fitted to the scenario, e.g., when segmenting pictures on walls, there should be no distinction between height and width (in contrast with our selection cases).

Moreover, our approach to segmentation in a cluttered environment of similar objects can be applied to other segmentation tasks with similar conditions, e.g., segmentation of marbles in a pile. A step towards this goal would be to define a *perspective circle* (i.e., a family of ellipses), instead of a PR, to fit the shapes of marbles in the image.

Although book segmentation is a non-trivial problem, a desired reorganization of a bookshelf is relatively easy to define. In long-term future work, our proposed application for tidying bookshelves might be extended to more complex challenges such as organizing other parts of a room directly in the image: objects placed on a desk or furniture in a room, for example. This could be done by obtaining a set of surfaces in a room, using a geometric room parsing method (e.g., [22]). Then, different segmentation approaches could be matched to each such surface, as well as different reorganization criteria.

Bibliography

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

[2] A. V. Aho, M. R. Garey, and J. D. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *Computer Vision and Pattern Recognition*, 2009.

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

[5] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG*, 28(3):24, 2009.

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

[7] D. Chen, S. Tsai, C. H. Hsu, J. Singh, and B. Girod. Mobile augmented reality for books on a shelf. In *International Conference on Multimedia and Expo*, 2011.

[8] D. M. Chen, S. S. Tsai, B. Girod, C. H. Hsu, K. H. Kim, and J. P. Singh. Building book inventories using smartphones. In *International Conference on Multimedia*, 2010.

[9] Y. Chen, H. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K. Gopinath, R. Briggs, and E. Geiser. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, 2002.

[10] T. Cho, M. Butman, S. Avidan, and W. Freeman. The patch transform and its applications to image editing. In *Computer Vision and Pattern Recognition*, 2008.

[11] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[12] D. Crasto, A. Kale, and C. Jaynes. The smart bookshelf: A study of camera projector scene augmentation of an everyday environment. In *Workshops on Application of Computer Vision/Motion*, 2005.

[13] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[14] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, 100(1):67–92, 1973.

[15] 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.

[16] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.

[17] 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. *Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996.

[18] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[19] S. Keren, I. Shimshoni, and A. Tal. Placing three-dimensional models in an uncalibrated single image of an architectural scene. *Presence: Teleoperators & Virtual Environments*, 13(6):692–707, 2004.

[20] J. Košecká and W. Zhang. Video compass. In *European Conference on Computer Vision*. 2006.

[21] D. Lee, Y. Chang, J. Archibald, and C. Pitzak. Matching book-spine images for library shelf-reading process automation. In *International Conference on Automation Science and Engineering*, 2008.

[22] D. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *Computer Vision and Pattern Recognition*, 2009.

[23] K. Matsushita, D. Iwai, and K. Sato. Interactive bookshelf surface for in situ book searching and storing support. In *Augmented Human International Conference*, 2011.

[24] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1:2, 1996.

[25] B. Micusik, H. Wildenauer, and J. Košecká. Detection and matching of rectilinear structures. In *Computer Vision and Pattern Recognition*, 2008.

[26] A. Mishra, Y. Aloimonos, and C. L. Fah. Active segmentation with fixation. In *International Conference on Computer Vision*, 2009.

[27] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *International Conference on Computer Vision*, 2009.

[28] N. H. Quoc, K. H. Woo, and W. H. Choi. Segmentation of books and characters for book recognition system of robot intelligence. In *ICROS-SICE International Joint Conference*, 2009.

[29] M. Rousson and N. Paragios. Shape priors for level set representations. In *European Conference on Computer Vision*. 2002.

[30] D. Shaw and N. Barnes. Perspective rectangle detection. In *Workshop of the Application of Computer Vision in the European Conference on Computer Vision*, 2006.

[31] E. Taira, S. Uchida, and H. Sakoe. Book boundary detection from bookshelf image based on model fitting. In *International Symposium on Information Science and Electrical Engineering*, 2003.

[32] A. Tamrakar and B. Kimia. No grouping left behind: From edges to curve fragments. In *International Conference on Computer Vision*, 2007.

[33] O. Tolba, J. Dorsey, and L. McMillan. A projective drawing system. In *Symposium on Interactive 3D Graphics*, 2001.

[34] E. Tretyak, O. Barinova, P. Kohli, and V. Lempitsky. Geometric image parsing in man-made environments. *International Journal of Computer Vision*, 97(3):305–321, 2012.

[35] 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 *International Conference on Multimedia*, 2011.

תקציר

מדף מסודר הינו נעים לעין ומשרה אווירה נעימה על החדר. למרות זאת, מדפי ספרים רבים אינם מסודרים וסידורם דורש מאמץ לא מבוטל. המוטיבציה שלנו במחקר זה היא לסדר את מדף הספרים ישירות בתמונה. האתגר העיקרי הינו לבצע סגמנטציה מדויקת של הספרים. אנחנו מציעים אלגוריתם סגמנטציה שמתגבר על הקשיים המרכזיים שנובעים מטקסטורה וטקסט על גבי הספרים, ספרים באוריינטציות שונות תחת פרספקטיבה ודחיסות הספרים על גבי המדף. במרכז האלגוריתם שלנו מוצג קונטור אקטיבי (active contour) תלוי צורה ושימושו על מנת להשיג קבוצת מועמדים לסגמנטציה של הספרים. תת-קבוצה מתוך קבוצת המועמדים נבחרת ע״י החלת כללים מרחביים על תצורת הספרים שעל המדף. אנו משתמשים בתוצאת הסגמנטציה על מנת לייצר תמונה חדשה של מדף הספרים, בה הספרים מסודרים, למשל ע״פ יישור הספרים למנח אנכי, או ע״פ קריטריונים אחרים כמו גובה, רוחב או צבע. גודלם של הספרים בתמונה החדשה נשמר כתלות בפרספקטיבה של הסצנה, ללא שחזור המבנה התלת ממדי.

המרכז הבינתחומי, הרצליה

בית ספר אפי ארזי למדעי המחשב

סגמנטציה של ספרים למטרת סידור מדפי ספרים

מוגש כחיבור מסכם לפרויקט מחקר לתואר M.Sc.

מגיש: ליאור טלקר

מנחה: דר' יעל מוזס

יוני 2013