Finding Optimal Non-Overlapping Subset of Extracted Image Objects

Filip Korč and Wolfgang Förstner

University of Bonn, Department of Photogrammetry
Nussallee 15, 53115 Bonn, Germany
filip.korc@uni-bonn.de,wf@ipb.uni-bonn.de
http://www.ipb.uni-bonn.de/

Abstract. We present a solution to the following discrete optimization problem. Given a set of independent, possibly overlapping image regions and a non-negative likeliness of the individual regions, we select a non-overlapping subset that is optimal with respect to the following requirements: First, every region is either part of the solution or has an overlap with it. Second, the degree of overlap of the solution with the rest of the regions is maximized together with the likeliness of the solution. Third, the likeliness of the individual regions influences the overall solution proportionally to the degree of overlap with neighboring regions. We represent the problem as a graph and solve the task by reduction to a constrained binary integer programming problem. The problem involves minimizing a linear objective function subject to linear inequality constraints. Both the objective function and the constraints exploit the structure of the graph. We illustrate the validity and the relevance of the proposed formulation by applying the method to the problem of facade window extraction. We generalize our formulation to the case where a set of hypotheses is given together with a binary similarity relation and similarity measure. Our formulation then exploits combination of degree and structure of hypothesis similarity and likeliness of individual hypotheses. In this case, we present a solution with non-similar hypotheses which can be viewed as a non-redundant representation.

Key words: object detection, non-overlapping solution, binary integer programming

1 Introduction

Object detection is one of the fundamental problems in computer vision. In this context, identification of the extent of an image object, that we more specifically refer to as object extraction, is a challenging and computationally demanding task.

Running single or multiple object extraction algorithms over an image at different locations and scales is a common approach. In this case, a number of independent, possibly overlapping regions are identified as representing an object of interest. We refer to these regions as candidates. See Fig. 1a for an

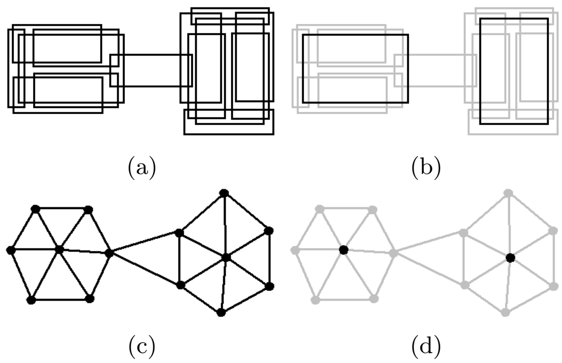


Fig. 1. (a) Image object candidate regions. (b) Selected subset of non-overlapping image object regions. (c) Image object candidate graph. Vertices denote object candidate regions. Links denote overlap between two candidate regions. (d) Image object candidate graph with selected candidates denoted by dark nodes. We note that both links and vertices are respectively weighted by the degree of overlap of two candidate regions and likeliness of individual candidates.

example. In the considered scenario, all regions are candidates of the same category. In addition, an extraction algorithm may yield some non-negative likeliness of individual regions.

We first turn candidate regions into a graphical representation, where graph vertices denote the candidates and links an overlap between two candidates, see Fig. 1c. We assign links with weights that reflect the degree of overlap between two candidates. In addition, every vertex is assigned with a weight corresponding to the likeliness of the individual candidate.

We consider every candidate as a plausible solution. Our objective is to select a non-overlapping subset that we call a solution and that is optimal with respect to the following requirements: (i) Every region is either part of the solution or has an overlap with it. (ii) The degree of overlap of the solution with the rest of the regions is maximized together with the likeliness of the overall solution. We note that the degree of overlap is a function of candidate pairs and the overall likeliness is a function of individual candidates. (iii) The likeliness of the individual regions, if provided by the extraction algorithm, influences the overall solution proportionally to the degree of its overlap with neighboring regions. The likeliness is specified in terms of a non-negative cost or a probability of a candidate representing an object of interest. Altogether, we look for a representative subset of a group of possibly overlapping candidates that is optimal with respect to the above mentioned requirements. An illustration for such a solution is given in Fig. 1b.

To find an optimal solution we reduce the problem to a constrained binary integer programming problem. The problem involves minimizing a linear objective function subject to linear inequality constraints. Both the objective function and the constraints exploit structure of the graph.

Extraction of non-overlapping objects, such as windows in facade images, serves as a natural example of the task under consideration. We note that the proposed method is not restricted to regularly shaped objects and allows a region to be a non-convex irregular image patch with one or more connectivity components.

Our formulation allows a generalization where we view candidates as hypotheses assigned with likeliness and region overlap as a binary relation reflecting the similarity of two hypotheses. We further view the degree of the overlap as a similarity measure of two hypotheses or, in other words, a degree of mutual support. In this view, the graphical representation encodes hypothesis similarity structure. Further, our formulation exploits combination of both degree and structure of hypothesis similarity and likeliness of individual hypotheses. In this case, we present a solution with non-similar hypotheses which can be viewed as a non-redundant representation. This generalization increases the applicability of the proposed method.

2 Related Work

The transition from Fig. 1c to Fig. 1d suggests similarity of the task to the formation of an image pyramid as proposed by [7][9]. This formation process is controlled by selecting surviving and non-surviving vertices of a graph. In this view, dark nodes in Fig. 1d would represent survivors possibly retained for the next level of the hierarchy. An alternative approach in [6] shows the graph decimation controlled by the image data. Our formulation may also be viewed as related to graph contraction introduced in [2] for building irregular pyramids.

Our motivation, however, is to use the graphical representation to set up a binary integer program that exploits the combination of the degree and structure of candidate overlap together with the likeliness of the individual candidates. Our formulation is used to select a subset of candidates. The selected subset is neither represented as a graph nor as a hierarchy level for further hierarchy building. In addition, as opposed to a graph theoretical approach, we use the graphical representation to encode the structure of a problem that we formulate within binary integer programming framework.

Our generalization is related to [8] where optimal feature groupings consistent with constraints is searched. Our generalization further relates to [11] where a set of hypotheses is generated from detected image features such as lines and a selection process is then used to choose among the generated hypotheses and to eliminate those without sufficient evidence.

Our approach is closely related to the selection of canonical subsets of image regions [3]. Further, a similar and more general approach employing a fully connected graph is presented in [4]. We are currently investigating similarities of our approach and the methods presented in the two above works.

3 Object Candidate Regions

We start with the result of a region extraction procedure. The result is a set of candidate regions V. A candidate region, or simply a candidate, is a region that we want to assign with an object category. Further, a result of object extraction is also a non-negative likeliness f, a quality measure of extracted candidates.

We represent candidates as a list V of N elements:

$$V = \{V_i\}_{i \in D} \tag{1}$$

where $V_i \subset I$ is a set of pixels in the image I associated with the ith extracted candidate region. In Eq. (1) index set $D = \{1, 2, \ldots, N\}$ denotes indices of the extracted candidates.

A likeliness $f : V \to \mathrm{R}$ of candidates V is denoted as

$$f(V_i) = f_i \tag{2}$$

In other words, likeliness is a function that assigns every candidate V_i with the likeliness value f_i. The likeliness f may correspond to a criterion that has been optimized during the object extraction process. In such case, we assume the optimization to be formulated as minimization – that is, the smaller the value f_i is, the more we value the ith candidate. If the quality measure is specified in terms of probabilities, we set the likeliness f to $f(V) = \{1 - P_i\}_{i \in D}$, where P_i is the probability of the ith candidate region being correctly assigned with the considered category. We set the likeliness f to $f : V \to 1$ if no candidate quality measure is given. There is no preference of one candidate over the other in this case.

We conclude with several comments that will become aparent in the next sections and that we wish to mention here for completeness and greater clarity. Later, we scale the likeliness by the degree of overlap with neighboring candidates and, hence, we require likeliness f to take on non-negative values. Further, we will see that if no candidate quality measure is given, the choice of the constant does not influence the result. Only the values relative to each other play a role in the overall cost that is being minimized. Last, we note that $f_i = 0$ implies that the ith candidate does not add any cost to the overall criterion. In other words, this only means that the candidate is more likely to be selected.

3.1 Candidate Region Model

We consider a graph

$$\mathcal{G} = (V, E, f, w) \tag{3}$$

where V is a finite set of candidates introduced in Eq. (1) and f is the function introduced in Eq. (2). We recall that f associates every candidate V_i with the likeliness value f_i. The set E is a subset of the set of all unordered pairs $\{V_i, V_j\}$ of distinct candidates V_i, V_j. A pair $\{V_i, V_j\}$ is an element of E if there is an overlap between the candidates V_i and V_j. Further, we introduce an overlap-based weight function $w : E \to (0, 1]$, where

$$w(\{V_i, V_j\}) = \frac{|V_i \cap V_j|}{|V_i \cup V_j|} \tag{4}$$

$|V_i|$ means the number of pixels of the candidate i. To conclude, our candidate region model $\mathcal{G}$ is a weighted undirected graph with no multiple edges and no loops. We maintain that both vertices V and edges E are respectively weighted by likeliness f and overlap w.

4 Binary Programming Optimization

To find the optimal non-overlapping subset of extracted candidate regions we use reduction to binary integer programming problem, see [12][10][5][1] for reference. We solve an optimization problem of the following form:

$$\begin{aligned} &\min && c^T x \\ &\text{subject to} && Ax \le b \end{aligned} \tag{5}$$

The problem in Eq. (5) involves minimizing a linear objective function $c^T x$ subject to linear inequality constraints $Ax \le b$. In Eq. (5), c is a real valued vector, b is an integer valued vector, A is an integer valued matrix, and the optimization variable x is a binary integer vector – that is, its entries can only take on the values 0 or 1. We denote by $x^\star$ the solution of the problem in Eq. (5).

4.1 Formalization of the Problem

Each element $x_i \in x$ of our solution x represents a candidate V_i being selected as part of the solution or, in other words, being assigned with the considered object category. If the candidate is selected, the element has value 1. If the candidate is rejected or, equivalently, assigned with the background category, the element has value 0.

In Eq. (1), we index the set V of candidates with the index set D. We consider candidates in this order as we need x to be a vector:

$$x = [x_1, \ldots, x_N]^T$$

Then $x_1, \ldots, x_N$ correspond to candidates $V_1, \ldots, V_N$ being or being not assigned with the category of interest. In all, our vector x has N elements, since we have N candidates to assign.

4.2 Measuring Overlap

We weight the likeliness of individual candidates based on overlap with other candidates. The greater the overlap is, the more we value the candidate. The weight o'_i of a candidate V_i is computed as a sum of overlaps with neighboring candidates V_j:

$$o'_i = \sum_{V_j \in N_i} w(\{V_i, V_j\}) \tag{6}$$

where N_i denotes the set of neighbors of the candidate V_i. w refers to the weight function in Eq. (4). We create a normalized weight vector $o = (o_1, \ldots, o_N)^T$, where $o_i = \frac{o'_i}{\sum_j o'_j}$. If overlap maximization was our only objective, this would be a linear function $o^T x$. Equivalently, we could minimize $-o^T x$.

4.3 Objective Function

We wish to not only take into account the overlap of the overall solution with the rest of the candidates but also its likeliness, recall requirements (ii) and (iii) in Sec. 1. Hence, we multiply elements f_i of the likeliness vector with respective elements o_i of the normalized weight vector and scale thus the likeliness of the individual candidates by the degree of its overlap with neighboring candidate regions. This represents an utility that is assigned to every candidate. Our objective is to maximize the utility of the overall solution or, equivalently, minimize the negative utility. This is a linear objective function $c^T x$ of the problem in Eq. (5) where c is a $N \times 1$ vector with coefficients

$$c_i = -f_i o_i$$

This formulation favors the most likely solution of minimal size and maximal neighborhood overlap.

4.4 Constraints

We demand that the solution found is consistent with all suggested object candidates, recall requirement (i) in Sec. 1.

Hence, the first set of constraints requires that every candidate V_i is either an element of the solution $V^\star = \{V_j | V_j \in V, j \in D, x^\star_j = 1, x^\star_j \in x^\star\}$ or is adjacent to a vertex in the solution $V^\star$. See Fig. 1d for illustration. Every vertex V_i in Fig. 1d is either part of solution denoted by dark nodes or is adjacent to these nodes.

We build a matrix A_1 and a vector b_1 such that $A_1 x \leq b_1$ to capture these linear constraints:

$$\begin{aligned} A_1 &= -(B_a + I_N) \\ b_1 &= -[1, \ldots, 1]^T \end{aligned}$$

B_a denotes the $N \times N$ adjacency matrix of the graph $\mathcal{G}$ and I_N is the $N \times N$ identity matrix. b_1 is a $N \times 1$ vector with elements equal -1. N denotes number of vertices V.

We wish to find a subset of non-overlapping objects. Hence, the second set of constraints specify that no pair of vertices in the solution $V^\star$ form an edge of

E. We represent these linear constraints with a matrix A_2 and a vector b_2:

$$\begin{aligned} A_2 &= B_i^T \\ b_2 &= [1, \ldots, 1]^T \end{aligned}$$

where $A_2 x \leq b_2$. B_i denotes the $N \times M$ incidence matrix of the graph $\mathcal{G}$ and b_2 is a $M \times 1$ vector with elements equal 1. M refers to number of edges E.

To represent the linear inequality constraints of the problem in Eq. (5) we combine both sets of constraints A_1, b_1 and A_2, b_2 in the matrix A and the vector b:

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

4.5 Complexity

In our problem, we assign every of N candidates with object or background category. The resulting complexity is thus $\mathrm{O}(2^N)$. Even though we face a hard problem, we can still aim for a globally optimal solution if the problem that we wish to solve is small.

We propose decreasing the size of the problem by first finding all connectivity components of the graph $\mathcal{G}$ in Eq. (3) and then solving the optimization problem in Eq. (5) separately for every individual component of the graph $\mathcal{G}$. In experiment shown in Fig. 5b, this approach produces 4 subproblems that are solved separately. This illustrates that the number of objects present in the image may still be large.

5 Experiments

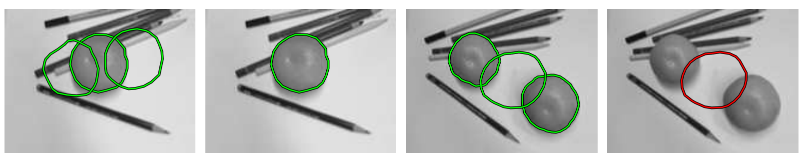
5.1 Graph Component of Size 2

Let us first consider solution for graph components of size 1 and 2. Component of size 1 represents trivial solution. We define symmetrical overlap. In this particular case, two overlapping candidates are equivalent in terms of weights o_i introduced in Sec. 4.2. It is the likeliness of individual candidates provided by extraction algorithm that eventually favors one of the candidates. In case no additional quality measure is provided, a candidate is selected at random.

5.2 Graph Component of Size Greater than 2

Let us now continue with the investigation of the case where three candidates overlap. Not distinguishing between isomorphic graphs, we face two possible situations. We discuss these two cases next. Afterwards, we present a situation with more than three candidates.

Case 1 In this case, three candidates form a graph with only two links as can be seen in Fig. 2ac. If no likeliness of individual candidates is provided, the method favors solution with greater neighborhood support or, equivalently, smaller weight o_i introduced in Sec. 4.2. For illustration see Fig. 2bd. This inevitably leads to an undesired outcome in the case shown in Fig. 2d. However, we note that if correct candidate likeliness is given, correct answer in (a) will be favored both by likeliness and strong neighborhood support. In case (c), correct solution will be supported by likeliness and weakly contradicted by neighborhood. Hence, correct likeliness will more likely outweigh weak neighborhood contradiction in (c) and, as a result, the method will favor correct solutions in both cases (a) and (c).



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

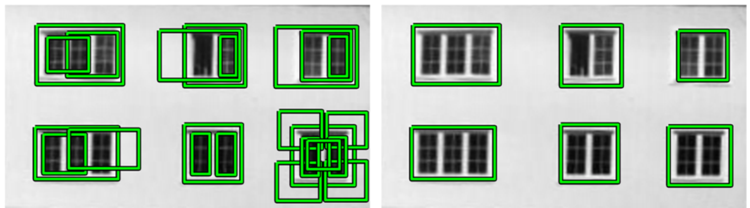
Fig. 2. (a)(c) Test data created manually to illustrate a situation with undesired outcome. No candidate likeliness is provided. (b)(d) Subset of extracted candidate regions selected automatically by the proposed method. See text for details.

Case 2 Let us now turn our attention to the case, where three candidates make up a triangle. Candidates in Fig. 3a have manually been created for illustration purposes and form a graph of six components. Every component, except for bottom right, illustrates an example of a triangle under consideration. No additional likeliness of individual candidates is provided in this case. In every triangle, single representative with greater neighborhood support or, equivalently, minimal weight o_i is favored during the optimization.

We emphasize, that it is not our objective to cover the area of the proposed candidates maximally. Closer examination of the middle window in the top row and the top right window in Fig. 3ab reveals that this is indeed not the case. The candidates selected in this experiment represent a solution that is optimal with respect to the requirements (i) and (ii) introduced in Sec. 1. Loosely speaking, our formulation aims for the solution that is the most consistent with the rest of the candidates.

To provide further insight, we point out that the candidate representing the top left window in Fig. 3 contains fully the rest of the unselected candidates. On the contrary, the solution found for the top right window is fully contained in an unselected candidate. Further, the degree of overlap with the unselected candidates varies as can be seen in the top row. In addition, the degree of overlap between the unselected candidates is variable, see the left and middle column.

Results in Fig. 3b represent candidates that have in the optimization process been identified as having the greatest neighborhood support.



(a) (b)

Fig. 3. (a) Test data created manually where only candidate region spatial extent is known, i.e., no additional quality measure is given. (b) Subset of extracted candidate regions selected automatically by the proposed method.

Should two or all links in a candidate triangle graph be assigned with the same weight, then the two or all vertices are equivalent. It is then the likeliness that favors a particular candidate case. If no likeliness is given and multiple nodes appear equivalent then a solution is picked at random.

Graph Component of Size Greater Than 3 Closer inspection of the bottom right component in Fig. 3 reveals that there are 8 competing candidates. 2 of these have an overlap with all other candidate regions. Candidate with greater neighborhood support is selected.

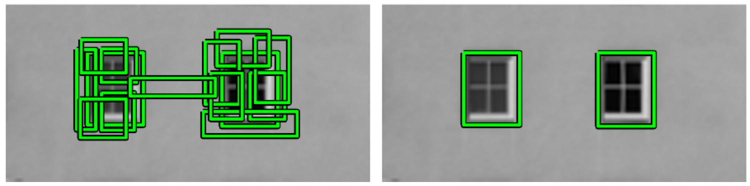
5.3 Multiple Representatives

Experiment shown in Fig. 4 shows a situation where 14 candidates make up a single candidate graph component. Again, data in Fig. 4a have manually been created to illustrate that multiple candidates may be selected to represent single connectivity component. We note that the situation in Fig. 4a corresponds to the graph in Fig. 1c. Again, it is only candidate region spatial extent that is provided. Subset of extracted candidates selected automatically by the proposed method is shown in Fig. 4b. This result corresponds to the candidate graph in Fig. 1d.

5.4 Application to Facade Window Extraction

In the last experiment we applied our approach to the problem of facade window extraction. We represent foreground objects as axis-aligned rectangles described in terms of its symmetry, homogeneity and darkness, which yields a 3-dimensional appearance feature vector.

We adopt a supervised approach and learn the appearance model of the foreground category from training data. We hand-annotated 160 facade windows



(a) (b)

Fig. 4. (a) Test data created manually to illustrate that multiple candidates may be selected to represent single connectivity component. This situation corresponds to the graph in Fig. 1c. Again, it is only candidate region spatial extent that is known. (b) Subset of extracted candidate regions selected automatically by the proposed method. This situation corresponds to the graph in Fig. 1d.

Table 1. Facade window extraction. Confusion matrix of (a) independent regionwise classifier and (b) the proposed method. Columns refer to the predicted positive (foreground) or negative (background) category and rows refer to the actual categories.

	Positive	Negative
True Positive	100	8
True Negative	309	452

(a)

	Positive	Negative
True Positive	94	14
True Negative	5	756

(b)

and randomly generated the same number of background objects. We model the appearance of the foreground category as a single Gaussian and classify the appearance feature vectors based on the probability density of the estimated normal distribution. We adopt a density threshold that yields the greatest accuracy on the training data.

During the object extraction process we first use an iterative mean shift based constrained optimization procedure to propose large number of regions that are locally optimal with respect to the appearance model, i.e., to propose large number of symmetrical non-homogeneous dark rectangles, see Fig. 5a. In our experiments, we test our approach on 18 images of both single and multiple facade windows. In total 869 regions are proposed by the iterative mean shift optimization procedure. These regions represent an input of the proposed method.

We now compare two approaches to the classification of the proposed regions. (a) We use an independent regionwise classifier or, in other words, the estimated appearance model, to make category assignment. (b) We select an optimal non-overlapping subset of the extracted objects or, equivalently, we select a subset of the regions that have been classified as foreground. Results of the approaches (a) and (b) are respectively given in Fig. 5bc. Both the object spatial extent and its quality measure (probability density) are exploited in this case.

The confusion matrices for both methods are given in Tab. 1. Note that the number of regions falsely classified as foreground by the independent regionwise classifier is reduced by the the factor of 61.80 when the proposed method is employed. Several standard terms are further given in Tab. 2. We note that the accuracy (AC) is improved by 53.98% and the precision (P) is improved by the factor of 3.88.

Table 2. Proposed method compared with independent regionwise classifier in the case of automatic extraction of facade windows. Comparison is given in terms of accuracy (AC), true positive rate (TP), false positive rate (FP), true negative rate (TN), false negative rate (FN) and precision (P).

	AC (%)	TP (%)	FP (%)	TN (%)	FN (%)	P (%)
classifier	63.52	92.59	40.60	59.40	7.41	24.45
our method	97.81	87.04	0.66	99.34	12.96	94.95

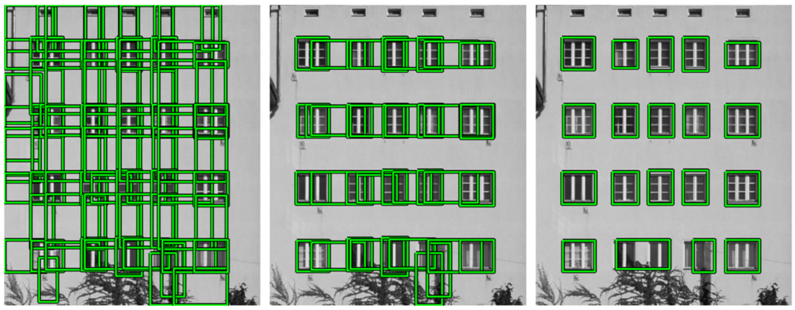
5.5 Computational Complexity

We conclude with comment on computational aspect. Es explained in Sec. 4.5, we reduce computational complexity by solving binary integer programming problem in Eq. (5) separately for individual graph components. Solving the problem for the bottom right component of 8 candidates presented in Fig. 4 took 0.48 seconds on Intel Pentium CPU, 3 GHz, 2 GB RAM and Matlab implementation. Solving the problem for example presented in Fig. 4 involving assignment of 14 candidates in a single component took 0.50 seconds.

To solve the binary integer programming problem in Eq. (5) in the presented experiments, we used the function `bintprog` from Matlab Optimization Toolbox Version 3.1. A linear programming (LP)-based branch-and-bound algorithm is used. The algorithm searches for an optimal solution by solving a series of LP-relaxation problems, in which the binary integer requirement on the variables is replaced by the weaker constraint $0 \leq x \leq 1$. The algorithm searches for a feasible solution, updates the best feasible point found as the search tree grows and verifies that no better solution is possible by solving a series of LP problems.

6 Conclusion

We present a solution to the following discrete optimization problem. Given a set of independent, possibly overlapping image regions and a non-negative likeliness of the individual regions, we select an optimal non-overlapping subset representing the original set. Regions are provided by a single or multiple object extraction algorithms. We illustrate the validity and the relevance of the formulation by applying the proposed method to the problem of window extraction. Last, we generalize our formulation to the case where a set of hypotheses assigned with likeliness is given together with a binary similarity relation and similarity measure. This generalization increases the applicability of the proposed method. In the future, we aim at a problem formulation with multiple objectives and decreased computational complexity.



(a) (b) (c)

Fig. 5. Automatic facade windows extraction. (a) Regions proposed by iterative mean shift optimization procedure. (b) Result of independent regionwise classification. (c) Optimal non-overlapping subset of extracted objects selected automatically by the proposed method. Both the object spatial extent and quality measure are exploited.

References

1. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, 2004
2. Brun, L., Kropatsch, W.: *Digital and image geometry: advanced lectures*, chapter Introduction to combinatorial pyramids, pages 108–128. Springer-Verlag New York, Inc., 2001
3. Denton, T., Abrahamson, J., Shokoufandeh, A.: Approximation of canonical sets and their applications to 2d view simplification. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 550–557, 2004
4. Denton, T., Demirci, M.F., Abrahamson, J., Shokoufandeh, A., Dickinson, S.: Selecting canonical views for view-based 3-d object recognition. In *Proc. of the 17th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 273–276, 2004
5. Hillier, F.S., Lieberman, G.J.: *Introduction to Operations Research* McGraw-Hill, 2001.
6. Jolion, J.-M., Montanvert, A.: The adaptive pyramid: a framework for 2d image analysis. *CVGIP: Image Understanding*, 55(3):339–348, May 1992
7. Meer, P.: Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, 45(3):269–294, March 1989
8. Mohan, R., Nevatia, R.: Using perceptual organization to extract 3d structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(11):1121–1139, 1989
9. Montanvert, A., Meer, P., Rosenfeld, A.: Hierarchical image analysis using irregular tessellations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(4):307–316, April 1991
10. Nemhauser, G.L., Wolsey, L.A.: *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988
11. Noronha, S., Nevatia, R.: Detection and modeling of buildings from multiple aerial images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(5):501–518, 2001
12. Wolsey, L.A.: *Integer Programming*. John Wiley & Sons, 1998