Constraints for Object Recognition in Aerial Images — Handling of Unobserved Features

Thomas H. Kolbe

Institute for Environmental Sciences
University of Vechta
P.O. Box 1553, 49364 Vechta, Germany
Thomas.Kolbe@uni-vechta.de

Abstract. In this paper we will show how constraint solving methods can be applied for the recognition of buildings in aerial images. Object models are transformed to constraint representations which are matched against extracted image features. To cope with disturbances caused by occlusions and noise, we distinguish between the unobservability of a) relations between object parts and b) object parts themselves. Whereas other approaches for solving over-constrained problems suggest to reduce the relaxation of a variable to the relaxation of its incident constraints, we argue that both cases have to be treated separately. Information theory is applied to derive constraint weights on a probabilistic basis. We extend constraints and variables in a way which provides for an adequate integration of constraint violation and variable elimination on the one hand, and allows the determination of the maximum likelihood estimation for the matching between model and image on the other hand.

1 Introduction

The automation of 3D object extraction, esp. buildings, is an issue of high importance due to the increasing demand for 3D city models. 3D data are needed for various applications including geo-information systems, transmitter placement, urban planning, cartography, and environmental related investigations.

The complexity and variability of buildings makes the use of strict models and low level image matching techniques like template matching and pattern classification unfeasible [24]. Relational matching was identified to be the appropriate scheme, where model and image are decomposed into graph structures and matched against each other [23, 26]. In section 2 we explain the employed models in detail.

To cope with the huge number of involved image features and the high complexity of building models, we have to apply efficient techniques in order to solve the underlying subgraph isomorphism problem. The application of constraint solving techniques [18], which has a long tradition in computer vision [28, 10, 23], proved to achieve good efficiency on large images [16]. However, they rely on the fact that all variables can be assigned values and all constraints can be fulfilled. Unfortunately this is often not the case, because of disturbances in the image like occlusions, noise and segmentation errors. Thus the problem often is over-constrained.

In recent years different methods for solving over-constrained systems have been developed [7, 19, 30]. Although several frameworks were proposed which have these methods as special cases [8, 15], the specific techniques concentrate either on the relaxation of constraints or on the elimination of variables. Unobservability of objects in aerial images occurs rather often, and sometimes it can even be predicted (for example, when we know about low contrast). Clearly in these cases the unobservability of an object may not be punished as hard as the violation of constraints between observed objects. Therefore a more natural model is required which makes this distinction explicit.

We developed a modeling scheme, which distinguishes between the unobservability of objects and the violation of relations between objects. Above, it integrates both effects into one evaluation function. Our concept is based on information theory and is motivated by work on relational matching [1, 26]. It is explained in section 4. The evaluation function gives the best matching a probabilistic semantic, namely the maximum likelihood estimation. The relaxation of constraints is achieved by program transformation, namely augmentation of the constraints by an additional variable which rates the violation resp. satisfaction of that constraint. Our concept can further be used in conjunction with standard consistency techniques and thus can be implemented i.e. on top of a CLP(FD) system. This is explicated in section 5.

At the end of the article (section 6) we finally show that our proposed modeling scheme also builds a natural link between MaxCSP and Dynamic CSP.

2 Building Recognition as a CSP

For the recognition of buildings we basically apply two models. On the one hand we have a (3D) object model, describing the shape and geometry of buildings. The model primitives consist of volumetric building parts that can be combined to more or less complex building structures. On the other hand we have a (2D) image model, which describes the objects and their relations that can be observed in the image. Here the primitives consist of points, lines and faces. Relations include (among others) line parallelism, neighbourhood, and collinearity.

The gap between the 3D object model and the 2D image model is bridged using aspect graphs. Aspect graphs enumerate all topologically invariant views on an object model, and can be efficiently computed for polyhedral 3D objects. Every aspect is represented in terms of the image model. Relations are propagated from the object model to the 2D aspects.

The modeling is presented in detail in [2]. The strategy, esp. how building hypotheses are generated and selected, is described in [5].

To identify an aspect and thus the underlying building in an image, the aerial raster image is segmented to derive a symbolic image description [6]. The extracted image features are also represented in terms of the image model. The features and their interrelationships form the feature relation graph FRG.

Now, the model and the extracted image features both are given as relational structures (graphs), where the nodes represent objects and the edges relations. The aim is to find the (small) model graph in the (huge) feature relation graph

FRG. To solve this subgraph isomorphism problem, we transform the model graph into a constraint satisfaction problem CSP(V, D, C), where the variables V represent the model primitives, and the constraints C the model relations. The variable domains D consist of the set of extracted image features. The task then is to find a valid assignment of extracted image features to the model variables, which satisfies all model constraints. This combinatorial problem is also known as the consistent labeling problem [10, 11].

As we have shown in [16], constraint logic programming over finite domains CLP(FD) [25, 13] has proved to be an appropriate platform for representing the different models and performing the search. The application of consistency techniques (forward checking and look-ahead) [17, 9, 18] provided an acceptable efficiency.

Applied constraints. For the task of building recognition we employ four different types of constraints. Fig. 1 shows the geometric constraints and fig. 2 the topological constraints.

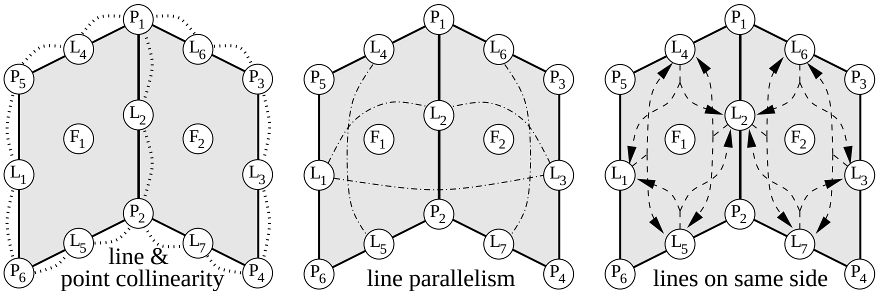

Fig. 1. Geometric constraints being used to describe a building roof consisting of points $P_1, \ldots, P_6$, lines $L_1, \ldots, L_7$, and faces F_1, F_2.

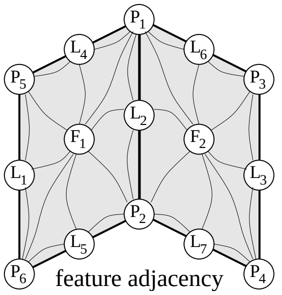

Fig. 2. Topological constraints used for the roof model.

The collinear(L, P) constraint states that a line segment L and a point P have to be collinear. This is realized by measuring the distance between the point and the straight line that goes through the line segment. The constraint now checks whether the distance is below a given threshold.

The line_parallel(L_1, L_2) constraint states that two line segments L_1 and L_2 have to be parallel. It is realized by computing the angle difference between both lines and the test whether it is smaller than a given threshold value.

same_side_line(L_{ref}, L_1, L_2) is a ternary constraint, stating that two line segments L_1 and L_2 have to lie in the same halfplane which is defined by the straight line going through a third line segment L_{ref}.

feature_adjacent(F_1, F_2) is a topological constraint, demanding that two image features F_1 and F_2 have to be neighboured, where features can be points, lines and faces. Since the feature adjacency graph is derived during the image segmentation and all neighboured feature pairs are enumerated, the test of this constraints reduces to a simple look-up.

Unobserved features and violation of constraints. The application of standard constraint solving methods demands that every variable can be assigned a value and all constraints can be satisfied. However, occlusions, low contrast, noise, and image segmentation errors often cause disturbances, which in the last consequence have two different effects: 1) relations that are expected in the model do not hold between the corresponding image features, and 2) expected model features are not observed in the image and therefore objects are missing (see fig. 3 for an example). Thus the given CSPs are over-constrained and the employed techniques have to be adapted to reflect these problems.

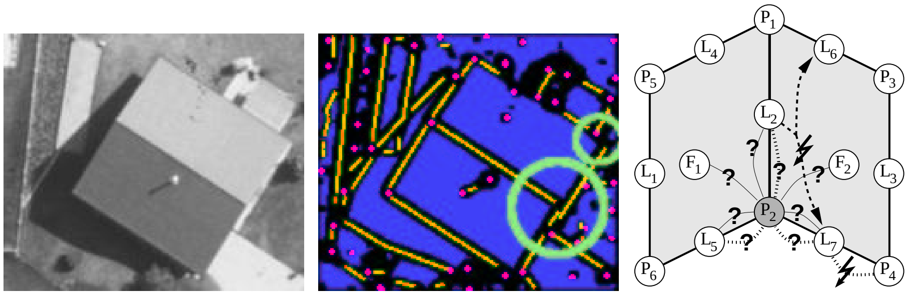

Fig. 3. Portion of an aerial image showing a saddleroof building (on the left) and the segmentation result with typical disturbances (in the middle). From the segmentation process the ridge point P_2 on the right of the roof was not observed. Therefore the incident relations cannot be observed either, which is indicated by question marks in the right image. Furthermore the same_side_line(L_2, L_6, L_7) constraint and the collinear(L_7, P_4) constraint is violated. This is indicated by the two flash symbols in the right image.

3 The Problem is Over-Constrained – What Literature Offers

An over-constrained system OCS is a CSP with no solution, because some constraints contradict others [14]. Nevertheless, to allow for the computation of (a somewhat sub-optimal) result, there are four possibilities of weakening the problem [7]: 1) enlarging a variable domain, 2) enlarging a constraint domain, 3) removing a variable, and 4) removing a constraint. Since there are generally different possibilities to weaken one CSP, they have to be rated by an evaluation function, allowing the definition of an ordering on them.

In literature several methods with specific evaluation functions for solving OCS have been proposed. They can be basically classified into two categories:

1. Relaxation of constraints (HCLP [30], MaxCSP [8])
2. Elimination of variables (Dynamic CSP [19])

Above, different frameworks (PCSP [8], GOCS [15]) have been suggested. These frameworks abstract from concrete evaluation functions and the way of weakening the original problem. As we will explain below, they have HCLP and/or

MaxCSP as instances. To clarify the relation between the mentioned methods, we have arranged them in a taxonomy as shown in fig. 4. In the following we will give a brief overview of each method, explain their position in the taxonomy, and discuss their use for our task of object recognition.

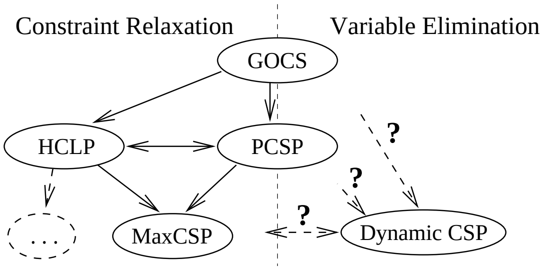

Fig. 4. Taxonomy of OCS methods. The only method focussing on variable elimination, DCSP, has not been cast in a general framework yet.

Maximum Constraint Satisfaction (MaxCSP) is the most simple and yet best-known method. It has been investigated in detail by Freuder and Wallace in [8]. The underlying (natural) metric simply counts the number of satisfied constraints. The best solutions are those with the smallest number of constraint violations.

Hierarchical Constraint Logic Programming (HCLP) was developed by Wilson, Borning et al. ([29, 30]), and expresses preferences by employing a constraint hierarchy. Constraints on the highest level of the hierarchy always have to be satisfied. Constraints on lower levels may be violated. The lexicographic ordering of the hierarchy levels ensures that the satisfaction of a more important constraint is always rated higher than alternatively satisfying an arbitrary number of constraints from lower levels.

HCLP can be considered as a framework for the relaxation of constraints, because it allows different ways to construct the evaluation function. For example, in [15] MaxCSP is shown to be an instance of HCLP.

Partial Constraint Satisfaction (PCSP) is a framework developed by Freuder and Wallace [7, 8] that regards any modification of the original CSP as a change of the problem space PS. It is general in the sense that any distance function that is a metric can be used to measure the difference of a modified problem space PS' to the original PS.

In [8] several concrete metrics are proposed, with MaxCSP being investigated in detail. However, none of them dealt explicitly with the relaxation of variables.

General framework for Over-Constrained Systems (GOCS) has been proposed by Jampel in his PhD thesis [15] as an abstraction from HCLP and PCSP. First it is shown that (most) problems expressed in HCLP can be transformed to PCSP and vice versa, when the evaluation function is expressed in terms of constraint augmentations (weakening method 2) from above). The main focus of GOCS lies on the compositionality of evaluation functions and solving schemes. Unfortunately, the relation to Dynamic CSP was only raised as a question of future research in the conclusion of the thesis.

Dynamic CSP (DCSP) was developed by Mittal and Falkenhainer [19] and is, to the best of our knowledge, the only scheme that explicitly handles the elimination of variables. In DCSP variables can be activated or switched off by special activity constraints. The circumstances under which variables are activated can depend on variable valuations in the form $P(v_1, \ldots, v_n) \rightarrow \mathit{active}\!: v_j$ or on the activity status from other variables $\mathit{active}\!: v_1 \wedge \ldots \wedge \mathit{active}\!: v_n \rightarrow \mathit{active}\!: v_j$. When a variable is deactivated, all its incident constraints also become deactivated. A subset of the variables, called the initial set, denotes the variables which have always to be active and therefore will be present in any solution.

Discussion. The main problem of the methods that map constraint violation and variable elimination onto the weakening of constraints (MaxCSP, HCLP) is that the elimination of a variable can only be simulated by the relaxation of its incident constraints. Since every constraint also has to be relaxable due to a simple violation of the underlying relation, what costs should be assigned to each constraint? These two cases have to be distinguished, because otherwise the cost for variable elimination would be equal to the violation of all its incident constraints, which does not necessarily reflect the importance of the variable and thus will often be over-estimated.

The same applies to the frameworks PCSP and GOCS. Although they are not restricted to the weakening of constraints, the authors state that any of the four weakening possibilities enumerated above may be reduced to the augmentation of constraints by new, compatible relation tuples. Whereas logically correct, this leads to the problems mentioned when defining an evaluation function. Moreover, augmenting constraint or variable domains to resolve the inconsistencies is not feasible, if we further want to use consistency techniques. They rely on the fact that 1) variable domains become monotonically smaller as search proceeds, and 2) the set of relation tuples defining compatible value combinations remains constant.

The problem of DCSP as the only available method for handling variable elimination is, that if variables are active, all constraints between them have to be satisfied. This clearly is too restrictive for our application domain (cf. fig. 3).

We will therefore need a scheme which explicitly distinguishes between the unobservability of objects (realised by variable elimination + relaxation of incident constraints) and relations between objects (by single constraint relaxations). Starting point for the integration of both effects into one evaluation function is the work of Vosselman on relational matching [26]. He was the first who succeeded to provide for a sound integration of unobservability of objects and their relations in an evaluation function for relational matching.

4 An Evaluation Function Based on Information Theory

In literature several evaluation functions for measuring the similarity of two relational descriptions (graph structures) $D_1 = (V_1, E_1)$ and $D_2 = (V_2, E_2)$ have been proposed. We will briefly review the crucial aspects of three important established schemes in chronological order, and explain how they overcome the deficencies (which are closely related to those of the reviewed OCS methods) of their ancestors.

Shapiro and Haralick [22] suggested a (still popular) metric to measure the structural error $\mathcal{E}$ of a mapping between two relational descriptions D_1 and D_2. They simply count the number of relation tuples that are not mapped by the mapping function $h : V_1 \mapsto V_2$ from E_1 to E_2 and vice versa: $\mathcal{E}(h) = |E_1 \circ h - E_2| + |E_2 \circ h^{-1} - E_1|$. Later they extended their metric by allowing normalized, weighted relation attributes (cf. [11]). There are two main problems with this metric: 1) Graph nodes in V_1 that have no mapping partner in V_2 are mapped to a dummy element (the so-called *wildcard*, symbolized by $*$). As we have discussed above with using constraint relaxation methods for variable elimination, it is difficult to assign costs for such wildcard mappings. Costs cannot be 0, because then the best mapping would only consists of wildcard mappings. 2) It is difficult to determine constraint weights, because relations can have attributes of different types (real, integer, symbolic).

Boyer and Kak [1] proposed to regard relational matching as a communication problem, where the first description D_1 is transmitted over a discrete, noisy communication channel and is received as a somewhat distorted description D_2 (see fig. 5). In 1948 Shannon introduced a quantitative measure for the information contained in a transmitted message [21]. The relation between the information contained in a symbol a and the probability that a will be chosen from an alphabet is defined by the equation $I(a) = -\log P(a)$. It can be regarded as a measure of surprise that a was observed: the higher the probability, the lower the information content we get. The conditional information $I(a|b) = -\log P(a|b)$ measures the surprise if we know that a was sent whereas we receive b. It is a measure for the uncertainty of the transmission.

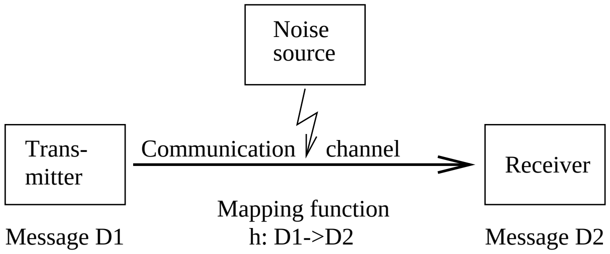

Fig. 5. Information theoretic modeling of a discrete communication channel

Boyer and Kak showed that the information content of a relational description is $I(D) = I(V) + I(E)$, with $I(V) = \sum_{v \in V} I(v)$ being the sum of the information contained in the nodes, and $I(E) = \sum_{e \in E}$ the information contained in the relations. The information of a node v_i itself consists of the sum[1] of the information contained in each attribute a_j: $I(v_i) = \sum_{a_j \in v_i} I(a_j)$. The information contributed by an attribute a depends on the probability of a taking a certain $value_l$: $I(a = value_l) = -\log P(a = value_l)$. The information of a relation e_i is defined as the sum over the information of all relation tuples $I(e_i) = \sum_{tup_j \in e_i} I(tup_j)$. Finally, the information contained in a relation tuple is the sum over the relation tuple attributes $I(tup_j) = \sum_{a_j \in tup_j} I(a_j)$, where $I(a_j)$ has the same definition as above.

[1] Assuming that all attributes are statistically independent.

Now the conditional information between two description $I(D_2|D_1)$ is defined in an analogous manner by replacing the probabilities by conditional probabilities. Since the conditional information measures the uncertainty of a transmission (and here also the quality of a channel), the task of finding the best mapping $\hat{h}$ from D_1 to D_2 can be reduced to the minimization $\hat{h} : \min_h I_h(D_2|D_1)$. The authors showed that $\hat{h}$ maximizes $P(h|D_1 \cap D_2)$, expressing that $\hat{h}$ is the most likely mapping under the given descriptions, the maximum likelihood estimation (cf. [4]) between D_1 and D_2.

The most important contribution of the proposed scheme for our purposes is that (in contrast to the first method presented above) weights now reflect the probabilistic nature of attribute values, and that they easily combine to an evaluation function having a probabilistic semantic. However, it does not solve the problem of wildcard mappings, because it is not possible to define the conditional information $I(*|a)$ between an attribute a and a wildcard.

Here the work of Vosselman [26] begins, who switched from using the conditional information to the mutual information, which is a symmetrical measure of the information that a symbol a gives about another b (and vice versa). It is defined as [12]:

$$
\begin{aligned}
I(a\,;\,b) &= \log_2 \frac{P(a \cap b)}{P(a) \cdot P(b)} && (1)\\
&= \log_2 \frac{P(a|b)}{P(a)} = \log_2 \frac{P(b|a)}{P(b)} && (2)
\end{aligned}
$$

Since the mutual information measures the similarity rather than the difference between relational structures, it has to be maximized in order to find the best mapping $\hat{h}$. Vosselman showed that maximizing $I_h(D_1; D_2)$ is equivalent to maximizing $P(h|D_1 \cap D_2)$. Still, $\hat{h}$ is the maximum likelihood estimation between the relational descriptions D_1 and D_2.

The crucial improvement wrt. the model of Boyer and Kak consists in the evaluation of wildcard mappings. Since a relation attribute from an object model r_m and a wildcard are statistically independent, following eqn. 1 this leads to

$$
\begin{aligned}
I(r_m\,;\,*) &= \log \frac{P(r_m) \cdot P(*)}{P(r_m) \cdot P(*)} && \text{(because } r_m \text{ and } * \text{ are independent)}\\
&= 0 && (3)
\end{aligned}
$$

This means that wildcard mappings neither support nor contradict to a mapping. A matching consisting only of wildcard mappings therefore would contain no (mutual) information.

To see how the modeling scheme can be applied to the constraint representation of object models we have presented in section 2, suppose that our model relations are denoted by r_m, and the relations that can be observed in the segmented image by r_i. Then the mutual information between every pair (r_m, r_i) can be calculated from eqn. 2 by applying *Jeffrey's Rule* [20]:

$$
I(r_i\,;\,r_m) = \log_2 \frac{P(r_i|r_m)}{P(r_i)} = \log_2 \frac{P(r_i|r_m)}{\sum_{r'_m} P(r_i|r'_m) \cdot P(r'_m)} \qquad (4)
$$

The mutual information has to be computed for every value combination of r_m and r_i, which in this case are the four tuples $(true, true)$, $(true, false)$, $(false, true)$, and $(false, false)$. Eq. 5 shows the calculation for the first tuple:

$$I(r_i = t\ ;\ r_m = t)\ =\ \log_2 \frac{P(r_i = t | r_m = t)}{P(r_i{=}t|r_m{=}t)P(r_m{=}t)+P(r_i{=}t|r_m{=}f)P(r_m{=}f)} \tag{5}$$

The following example (tab. 1 and tab. 2) demonstrates the computation of the mutual information, when the a priori probabilities for the model relation and the conditional probabilities for the image relation wrt. the model relation are given:

Table 1. Example for a priori probabilities for model relation r_m (left) and conditional probabilities of image relation r_i wrt. model relation r_m (right).

r	$P(r)$
$true$	0.17
$false$	0.83

$P(r_i\|r_m)$	$r_m = true$	$r_m = false$
$r_i = true$	0.95	0.00
$r_i = false$	0.05	1.00

Table 2. Resulting mutual information calculated from table 1 using equation 4. The upper left value was computed using eqn. 5 which is derived from eqn. 4 for the case $(r_i = true\ ,\ r_m = true)$.

$I(r_i\ ;\ r_m)$ [bits]	$r_m = true$	$r_m = false$
$r_i = true$	2.56	$-\infty$
$r_i = false$	-4.07	0.25

From this example we can see that if a predicted model relation can be observed in the image, it supports the matching by 2.56 bits. Otherwise, if the same relation would not hold for the image, it contradicts the matching by 4.07 bits. If the relation could not be observed, because of a wildcard mapping of an incident variable, it would be rated 0.

Before we now proceed with the definition of our CSP modeling scheme we will summarize the four main points from this section, because we will explicitly refer to them later:

1. The mutual information between a (relation or object) attribute and a wildcard is 0.
2. The combination of the ratings of objects and relations is done by simply building the sum over the mutual information of all attributes.
3. For relations having no other attributes except for $true/false$ the mutual information is given as a 2×2-table.
4. The mapping $\hat{h}$ with the highest mutual information corresponds to the maximum likelihood estimation.

5 Combining Variable Elimination and Constraint Relaxation

We assume that the model which should be matched with the extracted image features is given as a CSP(V, D, C) with variables V, associated domains D, and constraints C. Now, to distinguish between the relaxation of a constraint due to a simple violation or due to the unobservability of an incident variable every constraint $c(v_1, \ldots, v_n) \in C$ is extended by a three-valued domain variable $b \in \{-1, 0, 1\}$ to a constraint $c'(v_1, \ldots, v_n, b)$ with

$$\begin{aligned} c'(v_1, \ldots, v_n, b) \Leftrightarrow\ & (b = 1 \wedge c(v_1, \ldots, v_n)) \ \vee \\ & (b = -1 \wedge \neg c(v_1, \ldots, v_n)) \ \vee \\ & (b = 0) \end{aligned} \tag{6}$$

This variable can be seen both as an indicator and a control switch for the constraint. If on the one hand the original constraint c becomes entailed, b will be set to one. On the other hand, if b is set to 1, c has to be satisfied in order to satisfy c', thus the original constraint c will be enforced. Analogously, if c becomes contradictory[2], b will be set to -1, and if b is set to -1, the negation of c is enforced. The third case ($b = 0$) allows for an unconditional relaxation of c' wrt. the original constraint c.

Each variable domain $d \in D$ is extended by the wildcard value '$*$' (comparable to an explicit null value in record fields of databases) to a domain $d' = d \cup \{*\}$ with

$$\forall v_i \in V : \quad (v_i = * \quad \Leftrightarrow \quad \forall c'_j(\ldots, v_i, \ldots, b_j) \in C' : \ b_j = 0) \tag{7}$$

This condition relates the indicator variables to the original CSP variables. It expresses that if a wildcard is assigned to a variable, the b variables of all incident constraints have to be 0, to satisfy the constraints c'. The other way around, if the b variables of all constraints which are incident to a variable are set to 0, the variable has to be assigned the wildcard value.

As one might already suspect the three-valued variable b is closely related to the wildcard mapping and the mutual information. In fact it covers main point 1) from the end of the last section. The values -1 and 1 are inspired from point 3), esp. from the mutual information in the left column of tab. 2. A satisfied constraint supports a matching ($b = 1$), a violated constraint gives a malus ($b = -1$).

According to point 2) from last section, the evaluation function is defined as the sum over the b variables of the transformed constraints:

$$f(C') = \sum_{i=1}^{|C'|} b_i \ \text{ with } c'_i(\ldots, b_i) \in C' \tag{8}$$

Maximization of this function leads to the best matching (cf. point 4) from last section).

[2] Contradiction here has the same definition as in [15]: there does not exist any model (in the logical sense) in which all constraints without c are true, and c is also true.

Up to now, constraints have been unweighted. If a maximum likelihood estimation is desired, two things have to be done:

1. The values 1 and -1 of the b variables have to be replaced by the mutual information $I(r_m = \mathit{true}; r_i = \mathit{true})$ resp. $I(r_m = \mathit{true}; r_i = \mathit{false})$ (cf. left column of tab. 2).
2. The set of complementary constraints $\bar{C}$ has to be determined. When the set of all possible (binary) constraints is $\mathcal{C} = \{c(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$, the set of complementary constraints then is $\bar{C} = \{\neg c \mid c \in \mathcal{C} \wedge c \notin C\}$. These are the constraints which state that relations which are not true in the object model must also be false in the image. The values 1 and -1 of their indicator variables have to be replaced by the mutual information $I(r_m = \mathit{false}; r_i = \mathit{false})$ resp. $I(r_m = \mathit{false}; r_i = \mathit{true})$ (shown in the right column of tab. 2).

The problem here is that we get a quadratic number of constraints wrt. the number of variables, because $|C \dot{\cup} \bar{C}| = |\mathcal{C}|$. We therefore omit the complementary constraints in the following, knowing that the best matching is related to but not guaranteed to be the maximum likelihood estimation. It is an issue of future research to estimate and limit the error.

Application to building recognition. Adding the b variables significantly enlarges the search space (by a factor of $3^{|C|}$). However, a careful modeling in the context of an application domain allows a priori reductions of the search space. We will exemplify this in the following for the definition of geometric constraints and the knowledge about the observability of image features.

The geometric constraints `line_parallel` and `collinear` play an important role wrt. the quality of a reconstructed building. Violations of these constraints are not tolerated, because it can be observed that in all of these cases one of the participating image features is not correctly extracted from the image. As explained in section 2 the constraints were defined using thresholds. If we now set the threshold to the maximum difference we find in a (large) set of training matches, the constraint will be true for all training examples[3]. Thus the conditional probability $P(r_i = \mathit{true} | r_m = \mathit{true})$ that can be derived from the training data set will be 1 and therefore $P(r_i = \mathit{false} | r_m = \mathit{true}) = 0$. This means the constraint will not be violated in any acceptable matching. It can only be relaxed due to the unobservability of an incident variable. Therefore we can remove the -1 from the domains of the b variables of `line_parallel` and `collinear` constraints.

If one has knowledge about the observability of certain model parts, this can be exploited by the a priori elimination of wildcards from the resp. variable domains. In the system we have described in [5], the generation of building models is indexed by previously reconstructed 3D corners. Since the reconstruction of the corners uses the same image(s) as the final recognition of the complete building, the observability of the resp. building corners is propagated via the building model to the resulting CSP.

[3] We assume that the training matches were done by an expert, who can decide in every case, wether a certain deviation is acceptable.

Although a detailed explanation of the implementation is out of the scope of this paper, we will drop a few words on this topic. The modeling scheme is implemented by extending the CLP(FD) solver of the ECLIPSE system [27]. We adapted the inference rules for forward checking (FCIR) and look ahead (LAIR) as defined by van Hentenryck in [25], and provide a language interface similar to the constraint declarations of CHIP [3]. At the moment we only use the values -1 and 1 for the b variables, but we are currently evaluating test data sets to gain the probability distributions needed to compute the mutual information. Fig. 6 shows three example matchings that were determined using the proposed scheme.

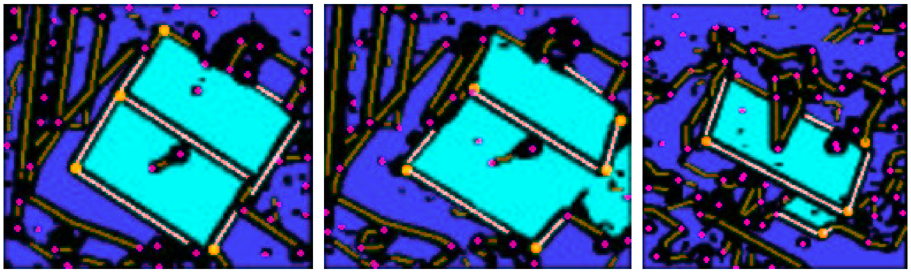

Fig. 6. Matching results for three buildings. The left building already was shown in fig. 3. The proposed evaluation function (correctly) decides that most violations of geometric constraints are best reflected by assuming the unobservability of incident variables. On the left one corner point and the right ridge point were mapped to a wildcard. Note that the right edge of the upper roof face is mapped correctly although it violates a `same_side_line` constraint (cf. fig. 3). In the middle one roof edge and the incident corner point were mapped to a wildcard, because the `line_parallel` and `collinearity` constraint could not be satisfied. Finally, on the right two corner points and the left edge of the lower roof face have a wildcard mapping.

6 Related Work

In section 3 we have pointed out that MaxCSP only considers the relaxation of constraints whereas Dynamic CSP is restricted to the elimination of variables. Here we show that both MaxCSP and Dynamic CSP may be regarded as special cases of our evaluation function. We show this by simulating MaxCSP and Dynamic CSP in terms of our model.

To implement the MaxCSP metric, one simply has to remove the 0 from the domains of the b variables and the wildcard $*$ from the variable domains. Thus constraints can only be relaxed, if the underlying relation is violated. Clearly, maximizing eqn. 8 then maximizes the number of satisfied constraints.

Dynamic CSP forbids the violation of constraints between active variables. Therefore we remove the -1 from the domains of the b variables. The initial variable set is always active and thus present in any solution. This is ensured by removing the wildcard $*$ from the variables in this set. Activity constraints of the form $P(v_1, \ldots, v_j) \rightarrow \mathit{active} : v_k$ are transformed to $P'(v_1, \ldots, v_j, b) \Leftrightarrow (b = 0) \vee (b = 1 \wedge (P(v_1, \ldots, v_j) \rightarrow v_k \neq *))$. The other activity constraints can be transformed in a analogous way. This simulation of Dynamic CSP is similar to the one mentioned in [19].

Finally, we demonstrate the use of the proposed scheme with an example given by Freuder and Wallace for PCSP in [8] (which can be seen as a simple version of a configuration problem). The problem is that we have a minimal wardrobe and some restrictions which clothes can be worn together with others. It is shown in fig. 7 and has under the given conditions no solution. When applying the MaxCSP metric to solve the over-constrained problem, every variable has to be assigned a clothing article, because MaxCSP is not capable of variable relaxation[4]. In this example this leads to the two ”best” solutions which suggest to wear Cordovans, a grey shirt, a grey tie, and dress blue slacks or Cordovans, a green shirt, a grey tie, and dress grey slacks. Both solutions obviously are inacceptable. Allowing also for the elimination of variables, we instead get a solution that suggests to drop the tie, which is not only the most comfortable but also the least eye-offending proposal.

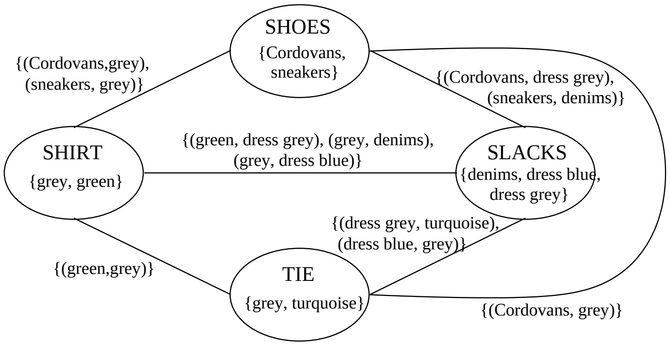
SHOES
{Cordovans, sneakers}
{(Cordovans,grey), (sneakers, grey)}
{(Cordovans, dress grey), (sneakers, denims)}
SHIRT
{grey, green}
{(green, dress grey), (grey, denims), (grey, dress blue)}
SLACKS
{denims, dress blue, dress grey}
{(dress grey, turquoise), (dress blue, grey)}
{(green,grey)}
TIE
{grey, turquoise}
{(Cordovans, grey)}

Fig. 7. Robot clothing example adopted from [8] and extended by a tie. The MaxCSP approach finds two best-rated solutions with two constraint violations each: 1) Cordovans, grey shirt, grey tie, dress blue slacks (yuck!), and 2) Cordovans, green shirt, grey tie, dress grey slacks (which is not much better). Our evaluation function in contrast decides that it is best to wear sneakers, denims, and a grey shirt and to drop the tie.

7 Conclusion and Future Work

We have presented a modeling scheme for CSPs which provides a smart integration of the unobservability of object parts and their interrelationships in the context of object recognition. The proposed evaluation function has a probabilistic basis. Information theory is applied to derive constraint weights from probability distributions of relations. These probabilities can be empirically derived from training matchings. This concept allows the definition of the best matching in a probabilistic sense, namely the maximum likelihood estimation between model and image data. Over-constrained systems which are modeled using this concept can be solved by the application of standard constraint solving methods.

We have implemented the scheme in CLP(FD) by extending the solver of the Eclipse system and have succesfully applied it for the recognition of buildings in aerial images. We have shown how domain specific restrictions can be used to

[4] The same applies to HCLP, which also cannot handle the elimination of variables.

sharpen the modeling of constraints and how a priori knowledge can be used for initial pruning of the search space.

Finally, by demonstrating that MaxCSP and DCSP are special cases of our concept we have established an up to now missing (natural) link between them.

Future work has two main focuses, the first concentrating on the evaluation function and the second on operational aspects:

1) The proposed evaluation scheme is close to maximum likelihood estimation. However, if the latter has to be ensured, not only the constraints for the relations that are true in the object model but also constraints for the complementary negated relations have to be posted. Clearly, the quadratic number of required constraints is too big. Therefore further investigation has to be done on how this can be avoided and what the probabilistic interpretation of such a reduced model will be.

2) The proximity to MaxCSP suggests the examination of its sophisticated heuristics in the context of our model. Furthermore it would be interesting to cast the proposed modeling scheme in the general frameworks of PCSP and GOCS to gain further insights into the evaluation function (i.e. wrt. compositionality).

Acknowledgements. This work was mainly done within the project "Semantic Modeling and Extraction of Spatial Objects from Images and Maps", especially in the subproject "Building Extraction" funded by the German Research Council (DFG). Thanks go to Lutz Plümer, Armin B. Cremers, Wolfgang Förstner, Gerd Gröger and Ingo Petzold who helped with many discussions and critical comments to develop these ideas. We thank the DFG for supporting our work.

References

[1] K. L. Boyer and A. C. Kak. Structural Stereopsis for 3D-Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):144–166, 1988.

[2] C. Braun, T. H. Kolbe, F. Lang, W. Schickler, V. Steinhage, A. B. Cremers, W. Förstner, and L. Plümer. Models for Photogrammetric Building Reconstruction. *Computer & Graphics*, 19(1):109–118, 1995.

[3] M. Dincbas, P. van Hentenryck, H. Simonis, A. Aggoun, T. Graf, and F. Berthier. The Constraint Logic Programming Language CHIP. In ICOT, editor, *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 693–702, 1988.

[4] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley & Sons, New York, 1973.

[5] A. Fischer, T. H. Kolbe, and F. Lang. Integration of 2D and 3D Reasoning for Building Reconstruction Using a Generic Hierarchical Model. In W. Förstner and L. Plümer, editors, *Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*. Birkhäuser Verlag, Basel, 1997.

[6] W. Förstner. A Framework for Low Level Feature Extraction. In J.-O. Eklundh, editor, *Computer Vision, ECCV '94, Vol. II*, number 801 in Lecture Notes in Computer Science, pages 383–394. Springer-Verlag, 1994.

[7] E. C. Freuder. Partial Constraint Satisfaction. In N. Sridharan, editor, *Proc. of the 11. Int. Joint Conf. on Artificial Intelligence IJCAI'89 in Detroit, MI, USA*. Morgan Kaufmann, 1989.

[8] E. C. Freuder and R. J. Wallace. Partial Constraint Satisfaction. In M. Jampel, E. Freuder, and M. Maher, editors, *Over-Constrained Systems*, number 1106 in LNCS, pages 63–110. Springer-Verlag, Berlin, 1996.

[9] R. M. Haralick and G. L. Elliott. Increasing Tree Search Efficiency for Constraint Satisfaction Problems. *Artificial Intelligence*, 14:263–313, 1980.

[10] R. M. Haralick and L. G. Shapiro. The Consistent Labeling Problem: Part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:173–184, 1979.

[11] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume II. Addison-Wesley Publishing Company, 1993.

[12] F. M. Ingels. *Information and Coding Theory.* Intext Educational Publishers, San Francisco, Toronto, London, 1971.

[13] J. Jaffar and M. J. Maher. Constraint Logic Programming: A Survey. *Journal of Logic Programming*, 19/20:503–581, 1994.

[14] M. Jampel. A Brief Overview of Over-Constrained Systems. In M. Jampel, E. Freuder, and M. Maher, editors, *Over-Constrained Systems*, number 1106 in LNCS, pages 1–22. Springer-Verlag, Berlin, 1996.

[15] M. B. Jampel. *Over-Constrained Systems in CLP and CSP.* PhD thesis, Dep. of Computer Science, City University London, UK, September 1996.

[16] T. H. Kolbe, L. Plümer, and A. B. Cremers. Using Constraints for the Identification of Buildings in Aerial Images. In *Proceedings of the 2. Int. Conf. on Practical Applications of Constraint Technology PACT'96 in London*, pages 143–154. The Practical Application Company Ltd., 1996.

[17] A. K. Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8:99–118, 1977.

[18] P. Meseguer. Constraint Satisfaction Problems: An Overview. *AICOM*, 2(1):3–17, March 1989.

[19] S. Mittal and B. Falkenhainer. Dynamic Constraint Satisfaction Problems. In *Proc. of AAAI 1990 in Boston*, pages 25–31, 1990.

[20] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, California, 2. edition, 1988.

[21] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Journal*, 27, 1948.

[22] L. G. Shapiro and R. M. Haralick. A Metric for Comparing Relational Descriptions. *IEEE Trans. on Pattern Analysis and Machine Intell.*, 7(1):90–94, 1985.

[23] L. G. Shapiro and R. M. Haralick. Relational Matching. *Applied Optics*, 26(10):1845–1851, 1987.

[24] P. Suetens, P. Fua, and A. J. Hanson. Computational Strategies for Object Recognition. *ACM Computing Surveys*, 24(1):5–61, 1992.

[25] P. van Hentenryck. *Constraint Satisfaction in Logic Programming.* Logic Programming Series. MIT Press, Cambridge, MA, 1989.

[26] G. Vosselman. *Relational Matching.* Number 628 in LNCS. Springer-Verlag, Berlin, 1992.

[27] M. Wallace, S. Novello, and J. Schimpf. ECLiPSe: A Platform for Constraint Logic Programming. Technical report, IC-Parc London, UK, August 1997.

[28] D. L. Waltz. Understanding Line Drawings of Scenes with Shadows. In P. H. Winston, editor, *Psychology of Computer Vision.* McGraw-Hill, New York, 1975.

[29] M. Wilson and A. Borning. Extending Hierarchical Constraint Logic Programming: Nonmonotonicity and Inter-Hierarchy Comparison. In *Proceedings of the 1989 North American Conference on Logic Programming in Cleveland, Ohio*, 1989.

[30] M. A. Wilson. *Hierarchical Constraint Logic Programming.* PhD thesis, Dept. of Computer Science, University of Washington, May 1993.