Properties of Local Geometric Constraints

Alistair J Bray
School of Cognitive and Computing Sciences
University of Sussex
Vaclav Hlavac
Faculty of Electrical Engineering
Czech Technical University (Prague)

Abstract

Formal analysis shows that the relationship between 2D line segments which is rotationally and translationally invariant can be expressed using four independent variables (five if scale is included). The formalism leads to a language for defining local geometric constraints and describing their properties. This terminology allows us to establish a criterion for evaluating the quality of a constraint set. Finally, a 2D constraint set is described that is deemed useful in light of the criterion proposed.

Model-based vision makes extensive use of the *local constraint* paradigm for matching models to images [1, 2, 3, 4, 5, 6, 7]; in this paradigm unary and binary constraints between linear segments are often used to prune an *interpretation tree* (describing the correspondence between model and image features) that is potentially combinatorial. The efficiency of this search is largely determined by the *constraint set*, and how capable it is of meeting the *ideal* criterion: that it should prune the tree at any node representing an incorrect correspondence, and grow the tree at any node representing a correct one. Various sets of constraints have been described for either matching in 3-space or 2-space. Such constraint sets are largely proposed *ad hoc* and justified by either practical results or intuitive plausibility (with some exceptions e.g. [8]). In this paper we examine the concept of a local constraint and its properties. We formalise the vocabulary for talking about constraints that is commonly adopted in the literature, and propose *qualitative* criteria for assessing a constraint set. This vocabulary and criterion allow us to say of a constraint set: "It is *bad* because ...".

The paper restricts itself to the relationship between a pair of 2D segments, although much of the content lends itself to 3D constraints also. In as much as the analysis is two dimensional, it has especial relevance to recognition systems that either recognise 2D objects, or recognise 3D objects using 2D features.

1 A Formalism

Consider the pair of non-parallel line segments $(\vec{s}_1, \vec{s}_2)$ from the set of segments $\mathbf{S} \equiv \{\vec{s}_1, \vec{s}_2, \ldots\}$. The relationship between $\vec{s}_1$ and $\vec{s}_2$ which is invariant to translation and rotation can be expressed using five mutually independent variables. One possibility is the 5-tuple $(\theta, a_1, a_2, a_3, a_4)$ [See Figure 1] . In this representation θ is the angle between $\vec{s}_1$ and $\vec{s}_2$; a_1 and a_3 are the distances

BMVC 1991 doi:10.5244/C.5.13

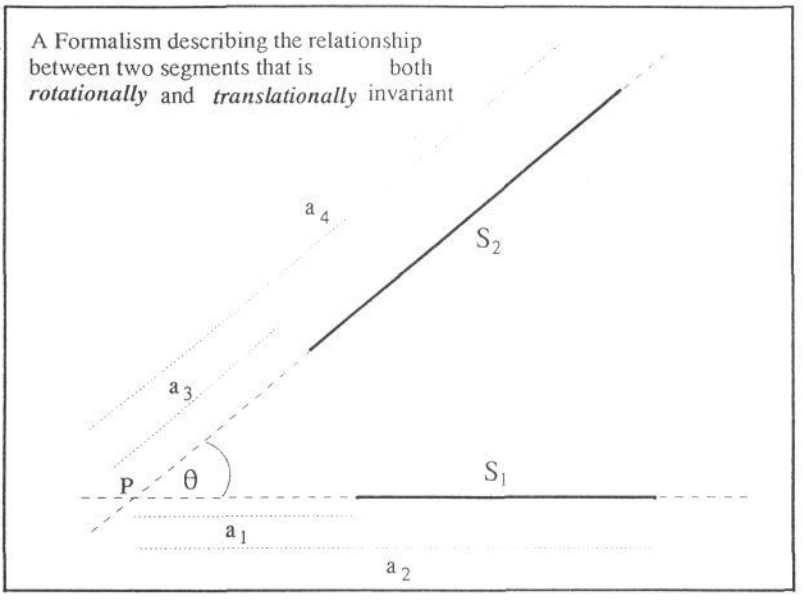

Figure 1: The Relationship between Segments

between the point of intersection (P) of the extended segment lines and the *start* of segments $\vec{s}_1$, $\vec{s}_2$; a_2 and a_4 are likewise the distances between P and the *ends* of $\vec{s}_1$ and $\vec{s}_2$.

If in addition, arbitrary scaling is to be allowed then all distances $a_1, \ldots, a_4$ can be normalised with respect to a_4 so that the five-tuple is $(\theta, a'_1, a'_2, a'_3, 1)$. This depends on four independent variables alone. An alternative quadruple that is invariant to scale, rotation and translation was formulated by Goad [1]; there are infinitely many such parameterisations.

1.1 Feature Definition

A correspondence between model and image segments can be checked by matching values of *features* evaluated for model segments (or tuples of model segments) against similar feature values obtained from image segments (or tuples of image segments). For our purposes a feature is defined as a function that maps a line segment or a tuple of line segments to the set of real numbers. Only binary and unary features are usually considered due to combinatorial problems. Given the set of segments $\mathbf{S} \equiv \{\vec{s}_1, \vec{s}_2, \ldots\}$, the set of model segments $\mathbf{M}$ ($\mathbf{M} \subset \mathbf{S}$), and the set of image segments $\mathbf{I}$ ($\mathbf{I} \subset \mathbf{S}$) we define the notation such that:

$$\mathbf{S}^2 \equiv \{\vec{s}^2 : \vec{s}^2 = (\vec{s}_p, \vec{s}_q) \ \textit{where} \ \vec{s}_p, \vec{s}_q \in \mathbf{S}; p \neq q\}$$

$$\mathbf{M}^2 \equiv \{\vec{m}^2 : \vec{m}^2 = (\vec{m}_p, \vec{m}_q) \ \textit{where} \ \vec{m}_p, \vec{m}_q \in \mathbf{M}; p \neq q\}$$

$$\mathbf{I}^2 \equiv \{\vec{i}^2 : \vec{i}^2 = (\vec{i}_p, \vec{i}_q) \ \textit{where} \ \vec{i}_p, \vec{i}_q \in \mathbf{I}; p \neq q\}$$

then we can define unary and binary functions as:

$$Unary\ Feature:\ g_1 : \vec{s} \mapsto R$$

$$Binary\ Feature:\ g_2 : \vec{s}^2 \mapsto R^5$$

The only unary feature is the length of the segment. A *strong* possibility for a translationally and rotationally invariant binary feature is $(\theta, a_1, a_2, a_3, a_4)$. It is strong because it gives a unique description of the relative position of the two lines. However, in practice it is reasonable to introduce several *weak* binary features. A weak feature doesn't determine the position of two line segments *uniquely*:

$$Weak\ Binary\ Feature:\ f_2 : \vec{s}^2 \mapsto R^x,\ where\ x < 5$$

1.2 Constraint Definition

A *Binary Constraint* C can be defined as anything that constrains the relationship between two segments or more precisely, a function that takes a feature function f_2, a matching function h, a pair of image lines $\vec{i}^2$, and a pair of model lines $\vec{m}^2$ and maps into the boolean set $\{true, false\}$ depending upon whether the feature generated from the image lines is consistent with the feature generated from the model lines. This may be expressed formally as:

$$Binary\ Constraint:\ C : T \mapsto \{true, false\}$$

$$T \equiv \{(f_2, h, \vec{i}^2, \vec{m}^2) : f_2 \in F; h \in H; \vec{i}^2 \in \mathbf{I}^2; \vec{m}^2 \in \mathbf{M}^2\}$$

$$F \equiv \{f : f\ is\ a\ feature\ function\}$$

$$H \equiv \{h : h\ is\ a\ matching\ function\}$$

In order to evaluate a constraint a feature value is computed from both image and model segment pairs and the result is matched one against the other using the matching function. There are various possibilities for the matching function; one is to evaluate a measure of distance between image and model features (many metrics are possible) and to compare this with a predefined threshold. Another possibility is based around computing bounds on the features: a minimal and maximal bound is precompiled for each feature for each possible pair of model line segments; matching succeeds if the value of the feature for a pair of image line segments lies within bounds on the model feature.

It is the nature of a constraint based upon a weak feature that it might fail to detect an incorrect model–image assignment; however, it is more important that they are *guaranteed* not to rule out a *correct* assignment. Such constraints based upon weak features are extremely useful given that there are sufficient image and model features to counteract their inherent inconclusiveness, and checking such constraints can usually be done very efficiently.

2 Constraint Properties

Given the above definition of a constraint C, this section outlines seven properties — five properties of a constraint, and two of a constraint set. These properties provide a language for talking about constraints and in which to formulate criteria for judging constraint sets.

1. **Reliability**
   A constraint C is *Reliable* to the extent that given a correct assignment $(\vec{i}^2, \vec{m}^2)$:

   $$C(f, h, \vec{i}^2, \vec{m}^2) \mapsto true$$

   If this is always the case then the constraint C has a reliability of 1. The reliability of a constraint can either be shown theoretically to be 1 or derived statistically from an image set[1].

2. **Plausibility**
   A constraint C is *Plausible* to the extent that given an incorrect assignment $(\vec{i}^2, \vec{m}^2)$:

   $$C(f, h, \vec{i}^2, \vec{m}^2) \mapsto false$$

   If this is always the case then the constraint has a plausibility of 1. The plausibility of a constraint can be derived statistically from an image set.

   Note: If a set of constraints is *independent* [See below] then reliability and plausibility can be computed for the set from its members using the Bayesian rule. Both reliability and plausibility of a *complete set* [See below] is 1. If the set is not independent then reliability and plausibility can be generated statistically from an image set.

3. **Noise Sensitivity**
   A constraint C is *Noise Sensitive* to the extent that, due to fragmentation (segments being shorter than they should be) it is unreliable i.e. given a correct assignment $(\vec{i}^2, \vec{m}^2)$:

   $$C(f, h, \vec{i}^2, \vec{m}^2) \mapsto false$$

4. **Scale Independence**
   A constraint C is *Scale Independent* iff for any scaling factors x_1 and x_2:

   $$C(f, h, x_1.\vec{i}^2, x_2.\vec{m}^2) \equiv C(f, h, \vec{i}^2, \vec{m}^2)$$

   where $x.\vec{s}^2$ represents a uniform scaling of both segments in the tuple $\vec{s}^2$ by x.

[1] A theoretical approach would be to demonstrate that given possible types of noise the constraint *must* always accept correct matches; the alternative is to use the constraint in practice and measure statistically how often the constraint *actually* accepts an incorrect assignment.

5. **Symmetry**
   A constraint C is *Symmetrical* iff:

$$C(f, h, (\vec{i}_n, \vec{i}_o), (\vec{m}_p, \vec{m}_q)) \equiv C(f, h, (\vec{i}_o, \vec{i}_n), (\vec{m}_q, \vec{m}_p))$$

   The advantage of symmetrical constraints is that only half of the constraint checks needs to be performed, and only half of the constraint information need be stored (if this was precompiled).

6. **Independence**
   A set of constraints is *Independent* iff, for any two constraints (C_a, C_b) in the set, C_a and C_b are uncorrelated over $\mathbf{M}^2, \mathbf{I}^2$.

7. **Completeness**
   A set of constraints is *Complete* iff the set of features that underlie it is complete i.e. assuming no noise, either of the segments can be uniquely reconstructed from the other segment and the set of feature values. If a set of constraints is complete then no further independent constraint can be added to the set.

The *intuitiveness* of a constraint may be considered another less important property. An intuitive constraint has a geometric simplicity that makes it simple to conceive. For example, the feature function f may correspond to a *perceptual* feature. This notion is obviously imprecise, but if constraints are intuitive then in a practical application the variation due to parameter settings (to cater with noise etc.) is meaningful and easier to determine.

3 An Evaluation Criterion

In the no-noise analysis it is sufficient to define a complete set of independent constraints — any such set is equivalent to any other. By making sure that the set is complete it is guaranteed that no spurious matches will pass the constraints; by making sure that they are independent, it guarantees that a minimal amount of constraint checking needs to be done (i.e. no redundant checks are being made). In the more practical noise-analysis it is necessary to extend the constraint bounds such that a larger proportion of the input set map to *true*. As a result *completeness* is lost and uncertainty is introduced. It is therefore the case that not all complete, independent constraint sets will be equivalent when noise is catered for, and extra constraints that might be redundant in the no-noise analysis may well now have pruning potential. It therefore makes sense to talk of a *good* constraint set, where *good* may be qualified in terms of the properties described above. We propose that an optimal constraint set would be as follows:

- The set would have reliability of 1. Anything less would mean that correct interpretations might not be found.

- The set would have a plausibility as close to 1 as possible. The lower the plausibility, the larger the search space.

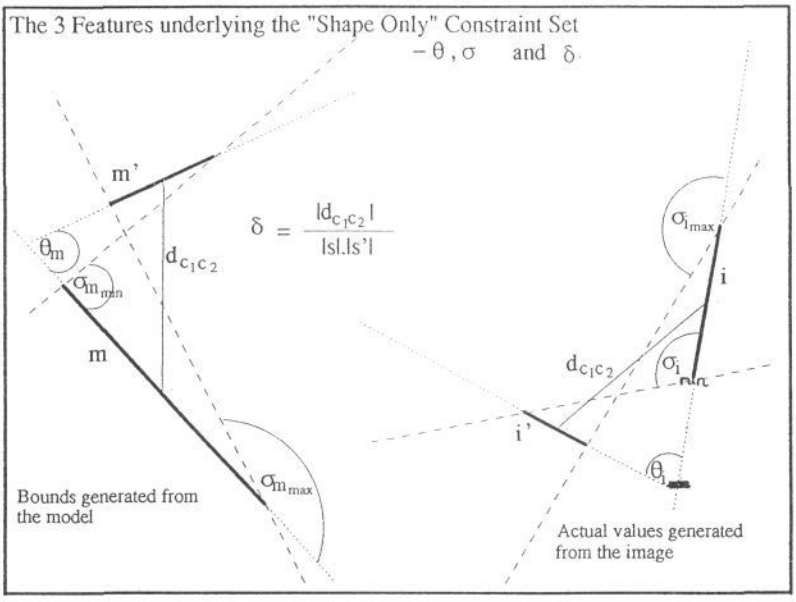

Figure 2: The Pruning Constraints

- The set would be as noise insensitive as possible. This would allow it to cope with occlusion, poor lighting and poor edge-detection.
- A maximum of the constraints in the set would be symmetrical. For each constraint that is symmetrical only half the checking needs to be performed and only half the model data need be stored compared to an asymmetrical one.
- The set would contain constraints that are largely independent. This means that minimal checking of constraints would be necessary (the more independent the constraint set is, the fewer constraints it contains).
- The set would be complete. This means that no further constraints could be incorporated without redundancy.

4 A "Shape Only" Constraint Set

In this section a constraint set is proposed that consists of three constraints alone. These constraints are illustrated in Figure 2 and rely solely upon the *shape* of the object.

The Angle Constraint

The angle constraint exploits the angle feature (θ), which measures the angle between two directed segments[2]. Formally, the angle feature function and the angle constraint are defined as:

$$f_\theta : \vec{s}^2 \mapsto R^1$$

[2]It is assumed that all segments are assigned a direction in the matching process.

$$C_\theta : \{(f_\theta, h_\theta, \vec{i}^2, \vec{m}^2)\} \mapsto \{true, false\}$$

The angle constraint states that for two image lines to correspond to two model lines it is necessary that they generate the identical θ value i.e. h_θ is the strict equality function. However, when matching over a range of viewpoints θ must fall within the bounds of possible values determined by that range. By measuring θ as the direction of one directed segment relative to the other, problems of discontinuities in absolute orientation values can be easily avoided. In this case h_θ is described by the inequality:

$$0° \leq min(f_\theta(\vec{m}, \vec{m'})) \leq f_\theta(\vec{i}, \vec{i'}) \leq max(f_\theta(\vec{m}, \vec{m'})) < 360°$$

The Direction Constraint

The direction constraint exploits the direction feature (σ), which measures the minimum and maximum angles of a vector connecting any point on segment $\vec{s}_1$ to any point on $\vec{s}_2$, relative to the direction of $\vec{s}_1$. Formally, the direction feature function and the direction constraint are defined as:

$$f_\sigma : \vec{s}^2 \mapsto R^2$$

$$C_\sigma : \{(f_\sigma, h_\sigma, \vec{i}^2, \vec{m}^2)\} \mapsto \{true, false\}$$

Given two directed segments $(\vec{s}_1, \vec{s}_2)$ and some vector $\vec{d}_{i,j}$ leading from $\vec{s}_1$ to $\vec{s}_2$ then σ is defined as the angle between $\vec{s}_1$ and $\vec{d}_{i,j}$. The bounds on σ over all possible $\vec{d}_{i,j}$ for the model segments $\vec{m}_1$, $\vec{m}_2$ define a *sector*. Therefore the bounds on σ may be computed from $\vec{m}_1$ and $\vec{m}_2$ simply by examining $\vec{d}_{i,j}$ values connecting the extrema of the two segments. Similar σ values may be computed from $\vec{i}_1$ and $\vec{i}_2$ in a like manner.

The direction constraint states that for two image lines to correspond to two model lines it is necessary that the bounds generated from the image lines *fall within* the bounds generated from the model lines i.e. h_σ is described by the two inequalities:

$$0° \leq min(f_\sigma(\vec{m}, \vec{m'})) \leq min(f_\sigma(\vec{i}, \vec{i'})) \leq max(f_\sigma(\vec{i}, \vec{i'})) \leq max(f_\sigma(\vec{m}, \vec{m'})) < 360°$$

$$max(f_\sigma(\vec{m}, \vec{m'})) - min(f_\sigma(\vec{m}, \vec{m'})) < 180°$$

The Distance Constraint

The distance constraint uses the distance feature (δ), which measures the ratio of the distance between the midpoints of the segments to the sum of the length of the segments. Formally, the distance feature function and the distance constraint are defined:

$$f_\delta : \vec{s}^2 \mapsto R^1$$

$$C_\delta : \{(f_\delta, h_\delta, \vec{i}^2, \vec{m}^2)\} \mapsto \{true, false\}$$

Given two directed segments $(\vec{s}_1, \vec{s}_2)$, and the vector $\vec{d}_{c_1,c_2}$ connecting the centres of $\vec{s}_1$ and $\vec{s}_2$, then f_δ is defined as:

$$f_\delta = \frac{|\vec{d}_{c_1,c_2}|}{|\vec{s}_1|.|\vec{s}_2|}$$

The distance constraint states that for two image lines to correspond to two model lines it is necessary that they generate the identical δ value i.e. h_θ is the strict equality function. However, when matching over a range of viewpoints, δ must fall within the bounds of possible values determined by that range. In this case h_δ is described by the inequality:

$$0 < min(f_\delta(\vec{m}, \vec{m'})) \leq f_\delta(\vec{i}, \vec{i'}) \leq max(f_\delta(\vec{m}, \vec{m'})) < \infty$$

5 Evaluation of the Constraint Set

The proposed set has the following *good* properties:

- **Reliability/Plausibility**
  The Angle and Direction constraints have a theoretical reliability of one despite fragmentation noise and together have a high plausibility (the plausibility of the Angle constraint being higher than that of the Direction constraint). The Distance constraint is susceptible to fragmentation noise; to maintain a reliability of one, bounds must be weakened that inevitably reduce its plausibility.

- **Noise**
  Only the Distance constraint is sensitive to fragmentation noise. However, the distance constraint it *robust* to this noise since it uses both the midpoints of the lines, and normalises with respect to the combined length. Both of these are robust to fragmentation.

- **Scale**
  The set is Scale Independent. That is, it can detect an object at any scale in the image, without having to consider a *scaling factor* or *locus of scales*[3]. However, only the Distance constraint is scale-related, so a scale dependent set is easily obtained by simply altering the distance constraint: the normalisation can simply be eradicated.

- **Symmetry**
  All three constraints are symmetrical. Constraints need only be applied in a one direction.

- **Independent**
  The set of constraints is independent. The three feature functions provide four independent parameters.

- **Complete**
  The set of constraints is complete. In the case of no fragmentation, the values of the Angle, Direction and Distance features are sufficient to reconstruct one segment from the other without uncertainty.

- **Intuitive**
  The Angle and Direction constraints are extremely intuitive. The Distance constraint is also intuitive, maybe to a lesser extent.

[3] Any interpretation generated using these constraints can therefore be *globally* inconsistent with regard to scale — however, this is unlikely and can be checked in the *model-test*.

Conclusions

We have performed a simple analysis of the relationship between a pair of 2D line segments, and provided a formalism to describe this relationship. We have used this formalism to define the concept of *features* and *constraints*. We have outlined *properties* of constraints that we consider useful in terms of practical algorithms, and tried to define these properties formally. We have proposed a *criterion* in light of these properties that allows a qualitative evaluation of a constraint set. Finally we have defined a small set of 2D constraints that we consider to come out quite favourably in light of the criterion; a more practical evaluation of two of these constraints, running on simulated data, can be found in [6, Chapter 3, 66–74]. Further work would be useful to determine practical *quantitative* methods of analysing constraint sets[4], given specific models and noisy data[5].

References

[1] **Goad C**. Fast 3D model-based vision. In Pentland A P, editor, *From Pixels to Predicates*, pages 371–391. 1984.

[2] **Murray D W**. Algebraic polyhedral constraints and 3D structure from motion. *Proceedings of the 5th Alvey Vision Conference, Reading*, pages 215–220, 1989.

[3] **Grimson W E L**. The combinatorics of object recognition in cluttered environments using constrained search. *Second International Conference on Computer Vision,IEEE*, pages 218–227, 1988.

[4] **Grimson W E L**. On the recognition of curved objects. *IEEE Pattern Analysis and Machine Intelligence PAMI*, 11(6):632–643, 1989.

[5] **Bray A J**. Object recognition using local geometic constraints: A robust alternative to tree search. *The First European Conference of Computer Vision ECCV*, 1990.

[6] **Bray A J**. *Recognising and Tracking Polyhedral Objects*. PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, UK, 1990.

[7] **Lamdan Y and Wolfson H J**. Geometric hashing: A general and efficient model-based recognition system. *Second International Conference on Computer Vision,IEEE*, pages 238–249, 1988.

[8] **Grimson W E L**. On the verification of hypothesised matches in model-based recognition. *The First European Conference of Computer Vision ECCV*, pages 489–498, 1990.

[4] Thanks to a reviewer for the idea that analysis *similar to* Principal Component Analysis might provide a statistical breakdown of which constraints in the set are doing the work.

[5] We wish to thank Dave Hogg, Kelvin Yuen, and Peter North for much helpful discussion of the topics covered in this paper.