Matching of Line Segments Across Multiple Views: Implementation Description (memo)

Tomas Werner

Visual Geometry Group
Department of Engineering Science
University of Oxford, U.K.

2002

1 Introduction

This memo describes a MATLAB implementation of matching of line segments accross multiple calibrated views. It extends the three-view matching described in [3, 4] to multiple views.

The input of the algorithm is a set of gray-scale images, a set of line segments (each represented by two endpoints) for each image, and the camera projection matrix for each image. The output is a set of matches (a match is a set of line segments, each from a different image) and reconstructed 3D line segments.

The text is organised as follows. After giving notation and terminology in §2, the main matching algorithm is described in §3. Supported types of camera calibration are described in §4. The subsequent sections then discuss non-trivial or interesting parts of the algorithm in detail.

2 Notation and terminology

Symbols related to images are indexed by $k = 1, \ldots, K$ in subscript, where K is the number of images. $\mathbf{P}_k$ are 3×4 camera matrices. I_k are image intensity matrices. The line segments detected in image k form the set $\{ s_k^n \mid n = 1, \ldots, N_k \}$. The image to which a segment s belongs is denoted by $k(s)$.

A *match* S is a list of image line segments, each from a different image. S need not contain segments from all K images. The set of images in which S has segments is denoted by $k(S) = \{ k(s) \mid s \in S \}$. E.g., if S contains three line segment from images 1, 3 and 7, then $k(S) = \{1, 3, 7\}$. A *set of matches* is denoted by $\mathcal{S}$ (caligraphic letter, note the difference to italic S).

The *pairwise photommetric score* c represents photommetric similarity of two line segments, each from a different image. It is based on normalised cross-correlation of the segments' neighborhoods.

The *score* C of a match S is a positive scalar. A higher score means a larger chance that S is a match. The score of a two-view match S is $C = -\log(1 - c)$, where c is the pairwise photommetric score. The score of a multiview match S is a sum of scores between selected pairs of images from $k(S)$. Roughly, scores can be interpreted as log-likelihoods.

The *epipolar beam constraint* is a geometrical constraint on corresponding line segments in two images [4] induced by the epipolar geometry.

The *reprojection constraint* is a geometrical constraint on corresponding line segments in three or more images, obtained by reconstructing the 3D segment and projecting it back to images.

View distance function $D : \{1, \ldots, K\}^2 \to \mathbb{R}_+$ says how 'far' image k is from image l. It is a heuristic to estimate the number of correspondences between two images, used to plan the matching strategy. If metric calibration is available, a possible choice for $D(k,l)$ is the distance between the camera centers of images k and l (even if this ignores the samera rotation).

3 The algorithm

The matching algorithm consists of two steps:

1. **Generating tentative matches**

   Input: Images I_k. Image line segments s_k^n. Camera matrices $\mathbf{P}_k$.
   Output: Set $\mathcal{S}$ of tentative matches. Score of each match $S \in \mathcal{S}$.

2. **Resolving ambiguities in tentative matches**

   Input: Set $\mathcal{S}$ of tentative matches and their scores.
   Output: Set $\mathcal{S}^* \subseteq \mathcal{S}$ of final matches and reconstructed 3D line segments.

The output from Step 1, the set $\mathcal{S}$ of tentative matches, is generally inconsistent. That means, two matches $S, S' \in \mathcal{S}$ can, for example, share a line segment in a certain image, violating thus the uniqueness constraint. It is the aim of Step 2 to find a subset of $\mathcal{S}$ which is consistent and (sub)optimal with respect to the scores.

The two steps are described in the following two sections.

3.1 Generating tentative matches

Tentative matches are generated as follows. First, initial two-view matches are generated from a chosen *base pair* of images. Then, the remaining images are searched for line segments consistent with the existing matches. These two steps are repeated for more base pairs. The strategy of choosing the base pairs and the order in which the remaining images are searched is planned using the view distance function.

Here is the complete algorithm:

1. Choose a set of base image pairs that evenly covers the image set. Set $\mathcal{S} := \emptyset$.
2. For each base image pair (k,l) do:

   (a) *Generate two-view matches from the base pair.*
   All tentative matches from the view pair (k,l) are added to $\mathcal{S}$. A pair S of line segments, one from image k and the other from image l, is a match iff it satisfies the epipolar beam constraint and its pairwise photommetric score c is higher than 0.6. The score of S is computed as $C = -\log(1-c)$.

   (b) *Search other images for support.*
   Each existing match S has image line segments in a subset $k(S)$ of all views. We seek to find segments in other views corresponding to S. Denote by $\bar{k}(S)$ the set of views in which S has no segments and that are not flagged as 'missing'. Denote by

   $$(k, \bar{k}) = \operatorname*{argmin}_{(k',\bar{k}') \in k(S) \times \bar{k}(S)} D(k', \bar{k}')$$

   the nearest (in terms of the view distance function D) views in the two sets. If $D(k,\bar{k})$ is smaller than a threshold, we search for a line segment s in view $\bar{k}$ that corresponds to S. That means, such that the match $S \cup \{s\}$ satisfies the reprojection constraint and the pairwise photommetric score c between segment s (in view $\bar{k}$) and the segment of S in view k is greater than a threshold (typically 0.6). The score of S is then updated as $C := C - \log(1-c)$.

   Three situations can occur:

i. A single segment s corresponding to a match $S \in \mathcal{S}$ is found in an image. Then the existing match S is extended by the new segment, $S := S \cup \{s\}$.

ii. More than one segment corresponding to S is found, e.g., s_1 and s_2. Then the original match S is removed from $\mathcal{S}$ and the new matches, $S \cup \{s_1\}$ and $S \cup \{s_2\}$, are added to $\mathcal{S}$.

iii. No segment corresponding to S is found. The view $\bar{k}$ of match S is flagged as 'missing'.

Step (b) is repeated until no existing match can be extended and no new match generated.

3.2 Matching

The task of the matching step is to select a subset $\mathcal{S}^*$ of the set $\mathcal{S}$ of tentative matches that is consistent and (sub)optimal in the sense of scores. A set $\mathcal{S}$ of matches is *consistent* iff each pair of its elements is compatible. Two given matches S and S' are compatible if one of the following constraints is satisfied:

- *Uniqueness constraint.* In every image from $k(S) \cap k(S')$, the segment of S and S' are different.

  During detection of image line segments, a single segment is often fragmented into several shorter ones. Using the multiple-view information, these fragmented segments can be merged back to the underlying larger segment [3, 4]. In this case, the uniqueness constraint becomes more complex as follows: S and S' may share a common segment in some image, but in the remaining images their segments must be mergeable.

- *Ordering constraint*, known from stereo (§8).

Ordering constraint and deframgentation can be switched off.

To find $\mathcal{S}^*$, we use the formalism from [2], which allows to treat uniqueness, ordering constraint, and merging of fragmented segments in a uniform manner. Let $Z(\mathcal{S}, S) \subseteq \mathcal{S}$ denote the *inhibition zone*, being the set of matches from $\mathcal{S}$ incompatible with S. Selecting $\mathcal{S}^*$ from $\mathcal{S}$ *quasi-stable matching* as follows:

```
1. Initialize $\mathcal{S}^* := \emptyset$.
2. While $\mathcal{S}$ is non-empty do
    (a) Find $S \in \mathcal{S}$ with the highest score.
    (b) $\mathcal{S}^* := \mathcal{S}^* \cup \{S\}$.
    (c) $\mathcal{S} := \mathcal{S} \setminus (Z(\mathcal{S}, S) \cup \{S\})$.
```

This algorithm uses a simple greedy strategy which nevertheless has nice statistical properties [2].

4 Calibration and orientation

The software distinguishes three levels of calibration:

- *Oriented projective* [1, 7]. Matrices $\mathbf{P}_k$ have such overall signs that for each image point $\mathbf{x}_{nk}$ (with a positive third component) and scene point $\mathbf{X}_n$ visible in each camera $\mathbf{P}_k$ we have $\mathbf{x}_{nk} \overset{+}{\sim} \mathbf{P}_k \mathbf{X}_n$, where $\overset{+}{\sim}$ denotes equality up to a positive scale. A projective reconstruction can be upgraded to an oriented projective one simply by swapping signs of $\mathbf{P}_k$ and $\mathbf{X}_n$; for this, we need to known some reconstructed points $\mathbf{X}_n$.

- *Quasi-affine* [1]. In addition, there exists a scene plane $\mathbf{A}$ such that $\mathbf{AC} > 0$ and $\mathbf{AX} > 0$ for each scene point $\mathbf{X}$ and camera $\mathbf{P}$. Here, $\mathbf{C}$ denotes the wedge product of the rows of $\mathbf{P}$, i.e., the oriented camera center. For the particular case of affine reconstruction, $\mathbf{A}$ is the plane at infinity. We assume w.l.o.g. that $\mathbf{A} = [0, 0, 0, 1]$.

- *Metric* (also known as Euclidean). In addition, the reconstruction differs from the true scene by a similarity transformation.

Quasi-affine and metric information is explicitly used by the software if available. Consequently, the results are better for a higher level of calibration.

The software often makes use of oriented projective geometry [5], which results in simpler and faster algorithms. Therefore, the overall signs of scene and image entities and multiview tensors *matter*. Correct signs of camera matrices $\mathbf{P}_k$ are required from the user.

Corrrect signs of fundamental matrices, epipoles, and inter-image homographies are internally ensured by using Grassmann algebra expressions for computing them. Thus, epipoles are computed as $\mathbf{e} = \mathbf{P}'\mathbf{C}$, where $\mathbf{C}$ is the oriented center $\mathbf{P}$. In contrast to [1], the inter-image homography $\mathbf{H}$ induced by a scene plane $\mathbf{A}$ and cameras $\mathbf{P}$ and $\mathbf{P}'$ is $\mathbf{H} = \mathbf{P}'\mathbf{G}$ where $\mathbf{G}$ is a 4×3 matrix mapping points in the first image onto $\mathbf{A}$. In tensor notation used in [6], $\mathbf{G}$ is given by

$$\mathbf{G}^a_A = \varepsilon_{ABC}\mathbf{P}^B_b\mathbf{P}^C_c\mathbf{A}_d\varepsilon^{abcd} \ .$$

5 Epipolar beam constraint

The epipolar beam constraint is a geometrical constraint on a pair (s, s') of corresponding line segments induced by the epipolar geometry. It is efficiently implemented using the oriented projective geometry.

Let $\mathbf{x}, \mathbf{y}$ be homogeneous 3-vectors representing end points of segment s. Similarly, $\mathbf{x}', \mathbf{y}'$ are end points of s'. Prior to matching, each image line segment is oriented according to the sign of intensity gradient such that its brighter side is always e.g. on the right. The segments s and s' satisfy the epipolar beam constraint iff the two following conditions hold:

1. $\det[\mathbf{e}, \mathbf{x}, \mathbf{y}]\ \det[\mathbf{e}', \mathbf{x}', \mathbf{y}'] < 0$ (see [7]).

2. $(\mathbf{x}', \mathbf{y}')$ intersects the beam delimited by the epipolar lines $\mathbf{Fx}$ and $\mathbf{Fy}$. This translates to two tests

   (a) $\det[\mathbf{e}, \mathbf{x}, \mathbf{y}]\,(\mathbf{Fy})^\top\mathbf{x}' < 0$,

   (b) $\det[\mathbf{e}, \mathbf{x}, \mathbf{y}]\,(\mathbf{Fx})^\top\mathbf{y}' > 0$.

Doing these tests for each pair (s, s') results in an algorithm with complexity $O(NN')$, where N, N' are numbers of segments in the first and second image, respectively. This can matter for huge images. The complexity could be descreased by first sorting the segments (e.g. vertically) and then using search in a sorted list.

The segment that passes the tests is clipped by the epipolar lines. This is also done conveniently using oriented projective geometry, but we skip the details.

This implementation of the epipolar beam constraint works not only for cameras with planar retinas (conventional cameras) but also for central panoramic cameras. For planar retinas, the third coordinates of the homogeneous 3-vectors representing image points must be positive.

5.1 Clipping by scene planes

If $\mathbf{P}$ and $\mathbf{P}'$ are quasi-affine [1, 7], we have an additional knowledge that all 3D segments must be in front of a single scene plane $\mathbf{A}$. The clipping by a plane can be easily incorporated into Test 2 above. Straight line $\mathbf{l} = \mathbf{x} \vee \mathbf{y}$ of the segment in image 1 is mapped into image 2 via the homography $\mathbf{H}$ induced by $\mathbf{A}$ as $\mathbf{l}' = (\mathbf{le})\mathbf{l}\mathbf{H}^{-1}$. Then, the segments in image 2 are discarded that have any end point ‘behind’ this line, $\mathbf{l}'\mathbf{x}' < 0$ or $\mathbf{l}'\mathbf{y}' < 0$.

6 Reprojection constraint

It is a geometrical constraint on a match consisting of three or more segments. It is used to find a line segment s in image $k(s)$ corresponding to an existing match S. It requires three tests:

1. The reprojection residuals of the 3D line reconstructed from $S \cup \{s\}$ are small.
2. Reprojection of S into image $k(s)$ has a non-empty overlap with s.
3. Reprojection of S into image $k(s)$ has equal orientation with s.

Test 1 is speeded up by doing first linear reconstruction of the 3D segment using a large reprojection threshold, and only if this is passed, non-linear reconstruction (maximum-likelihood estimate) is done.

7 Pairwise photommetric score

Pairwise photommetric score is critical for the performance of the matching algorithm, therefore great care has been devoted to its design. After experimenting with the score described in [4] and several other scores, the algorithm described below was observed to perform very well.

Let $(\mathbf{X}, \mathbf{Y})$ be a 3D line segment and $\mathbf{P}, \mathbf{P}'$ two camera matrices. The pairwise photommetric score c between image line segments $s = (\mathbf{x}, \mathbf{y}) = (\mathbf{PX}, \mathbf{PY})$ and $s' = (\mathbf{x}', \mathbf{y}') = (\mathbf{P}'\mathbf{X}, \mathbf{P}'\mathbf{Y})$ is computed as follows:

1. For each point $\mathbf{z}_i$ of s do:
   (a) Three windows are taken in the first image: centered in $\mathbf{z}_i$, to the left from $\mathbf{z}_i$, to the right from $\mathbf{z}_i$.
   (b) Compute normalized cross-correlation of these windows with the same windows mapped to the second image by the homography $\mathbf{H}$ described below.
   (c) Set the score c_i of $\mathbf{z}_i$ to the highest of these three scores.
2. If the number of segment points $\mathbf{z}_i$ with scores higher than 0.6 is smaller than 10, set $c := 0$. Otherwise, set c to the average of those c_i that are higher than 0.6.

The motivation behind using three rather than one window in Step 1a is as follows. Photometric similarity of the neighbourhoods of s and s' are caused by existence of a (nearly) planar surface with tangent plane $\mathbf{A}$, visible from both images, on which the 3D segment $(\mathbf{X}, \mathbf{Y})$ lies. Three cases can occur: (i) this surface is a single plane, (ii) this surface are two different planes adjacent to the 3D segment from either side, and (iii) the visible surface is a single plane adjacent to the 3D segment from a single side (it means, the segment is an occluding edge).

Ideally, $\mathbf{H}$ should be the inter-image homography induced by $\mathbf{A}$. We know only two of the three parameters of $\mathbf{A}$, the unknown parameter being the rotation around $(\mathbf{X}, \mathbf{Y})$. The next two sections show how $\mathbf{H}$ is computed if metric calibration is or is not available.

7.1 Metric calibration

If metric calibration is available, we assume that $\mathbf{A}$ is as orthogonal to the viewing direction as possible. This is formalized by taking the normal of $\mathbf{A}$ to be orthogonal both to the viewing direction and to $(\mathbf{X}, \mathbf{Y})$. As the viewing direction, the line joining the middle point between the camera centers with $\mathbf{X}$ is considered.

7.2 Non-metric calibration

If metric calibration is not available, the missing parameter of $\mathbf{A}$ must be determined in terms of image entities rather than scene entities. Let row 3-vectors $\mathbf{l} = \mathbf{x} \vee \mathbf{y}$ and $\mathbf{l}' = \mathbf{x}' \vee \mathbf{y}'$ be image lines of the segments s and s'. The family of homographies consistent with $\mathbf{P}$ and $\mathbf{P}'$ and mapping $\mathbf{l}$ to $\mathbf{l}'$ is given by [4]

$$\mathbf{H}_\mu = \mathbf{H}_0 + \mu \mathbf{e}' \mathbf{l} \tag{1}$$

where

$$\mathbf{H}_0 = [\mathbf{l}']_\times \mathbf{F} \tag{2}$$

and μ is the parameter corresponding to rotation of the plane $\mathbf{A}$ around the 3D line.

If the segment $(\mathbf{X}, \mathbf{Y})$ lies in the epipolar plane (or nearly so), $\mathbf{H}_\mu$ is inaccurate which results in a wrong pairwise photommetric score. The instability is removed if the segment is measured in another image(s) where it is not parallel with the epipolar lines. This stabilization may not prevent from rejecting an initial two-view match which is nearly parallel to epipolars, however it allows to obtain more accurate scores for this match later when segments from more images are added to it.

We will show how to use the information from other views in (1). First, the corresponding 3D segment $(\mathbf{X}, \mathbf{Y})$ is reconstructed from *all* the images and reprojected to our two images as segments $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$. The homography $\mathbf{H}_\mu$ satisfies

$$\mathbf{H}_\mu \mathbf{x} \overset{+}{\sim} \mathbf{x}', \quad \mathbf{H}_\mu \mathbf{y} \overset{+}{\sim} \mathbf{y}' \ . \tag{3}$$

In this case, (1) still holds but (2) can be replaced by

$$\mathbf{H}_0 = [\mathbf{e}']_\times \mathbf{F} + \mathbf{e}' \mathbf{m} \tag{4}$$

where the row 3-vector $\mathbf{m}$ is uniquely determined by (3).

In the case of metric calibration (§7.1), the stabilization by other images is done automatically.

To determine μ, we assume *area element invariance*, saying that area elements in corresponding points on s and s' are equal. In fact, this means that the images are assumed to have the same scale. Let $\mathbf{u}, \mathbf{u}'$ denote the non-homogeneous 2-vectors representing the homogeneous points $\mathbf{x}, \mathbf{x}'$, and $\mathbf{u}' = h_\mu(\mathbf{u})$ denote the homography mapping expressed by matrix $\mathbf{H}_\mu$. Area invariance means that the Jacobian of $h(\mathbf{u})$ equals identity,

$$\det \frac{\mathrm{d} h_\mu(\mathbf{u})}{\mathrm{d} \mathbf{u}} = 1 \ . \tag{5}$$

While for general $\mathbf{u}$ the Jacobian is a complicated rational function of μ, it turns out to be a linear function of μ if $\mathbf{u}$ lies on $\mathbf{l}$. Therefore, μ can be easily computed from (5).

If the signs of $\mathbf{F}$ and $\mathbf{e}'$ and orientations of the segments are correct, setting the Jacobian to 1 ensures that $\mathbf{H}_\mu$ maps correctly the left and right sides of the line.

8 Ordering constraint

Imposing the *ordering constraint* in the resolving phase of the matching algorithm (§3.2) can reduce the number of false positives and mismatches. However, it excludes some configurations of reconstructed 3D line segments, typically thin lines in the foreground.

The constraint is well-known from stereo vision, where it is applied on two point correspondences lying on a common epipolar. We can restate this in 3D as follows. The ordering relation is defined for two scene points $\mathbf{X}, \mathbf{Y}$ visible in two images that are coplanar with the camera centers $\mathbf{C}, \mathbf{C}'$ (in other words, $\mathbf{X}$ and $\mathbf{Y}$ lie in a common epipolar plane). The relation is satisfied if the line joining $\mathbf{X}$ and $\mathbf{Y}$ does not separate $\mathbf{C}$ from $\mathbf{C}'$ and violated otherwise. The concept of separation is meaningful here since oriented projective calibration is assumed, i.e., the overall signs of $\mathbf{X}, \mathbf{Y}, \mathbf{C}, \mathbf{C}'$ matter. The relation can be restated in image entities, in terms of (oriented) epipoles $\mathbf{e}, \mathbf{e}'$ and image projections $\mathbf{x}, \mathbf{y}, \mathbf{x}', \mathbf{y}'$.

Given two 3D line segments, the ordering relation is defined for their points of intersection with the epipolar plane, if these exist. In other words, the constraint is applied pointwise.

Extension to multiple images is possible by performing the test for all pairs of camera centers. Note, this is an $O(K^2)$ algorithm, therefore quite time consuming for a large number of images.

9 Example results

After matching image segments, 3D segments are reconstructed using non-linear method (maximum-likelihood estimation). Optionally, this reconstruction can be constrained by one or three mutually orthogonal principal directions: segments are first classified whether their intersection with the plane at infinity goes through a principal point or line, and then reconstructed with this constraint.

The algorithm has been tested on a number of image sets using both metric and projective calibration. 3D line segments obtained by matching three example image sets are in Figure 1. The runtime for the MATLAB implementation was several minutes on a 1 GHz PC laptop.

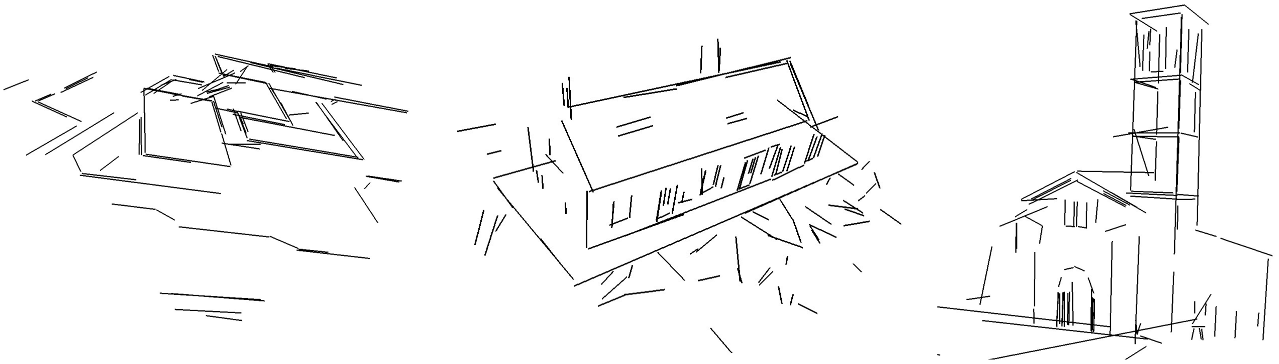

Figure 1: Reconstructed 3D lines for respectively Aerial (6 images), Dunster (10 images), and Valbonne (15 images) image sets.

References

[1] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, Cambridge, UK, 2000.

[2] Radim Šára. Sigma-delta stable matching for computational stereopsis. Technical Report CTU–CMP–2001–25, Center for Machine Perception, K333 FEE Czech Technical University, Prague, Czech Republic, September 2001.

[3] C. Schmid and A. Zisserman. Automatic line matching across views. In *CVPR97*, pages 666–671, 1997.

[4] Cordelia Schmid and Andrew Zisserman. The geometry and matching of lines and curves over multiple views. *Int. J. Comput. Vision*, 40(3):199–233, 2000.

[5] Jorge Stolfi. *Oriented Projective Geometry: A Framework for Geometric Computations.* Academic Press, Inc., 1250 Sixth Avenue, San Diego, CA 92101, 1991.

[6] B. Triggs. Matching constraints and the joint image. In *ICCV95*, pages 338–343, 1995. See also the full version on www, submitted to IJCV.

[7] Tomáš Werner and Tomáš Pajdla. Oriented matching constraints. In T Cootes and C Taylor, editors, *British Machine Vision Conference 2001*, pages 441–450, London, UK, September 2001. British Machine Vision Association.