Consensus Set Maximization with Branch and Bound

Orion Junkins

March 26, 2024

1 Introduction

Identifying a transformation model between two images, given a set of correspondences, is a classical optimization problem in Computer Vision. Naive approaches fail, due to large search spaces for model parameters, relatively large amounts of data noise, and relatively high outlier ratios. This assignment describes a solution to this problem using Consensus Set Maximization with Branch and Bound.

2 Problem Definition

Let S be a set of n correspondences between two images S. S Can be partitioned into an inlier-set $S_I \subseteq S$ and an outlier-set $S_O = S \setminus S_I$. Each correspondence in S comprises two points, $\boldsymbol{p_i} = (x_i, y_i)$ and $\boldsymbol{p_i'} = (x_i', y_i')$, denoting the point in the left and right image, respectively.

The two images are known to be related by some 2D translation modeled by $\boldsymbol{\theta} = \mathbf{T} = (T_x, T_y)$, denoting the respective translation in the x and y directions. The cost function, $\boldsymbol{f}$, is composed of two components: error on the x-axis and error on the y axis:

$$f_x(\boldsymbol{\theta}, \boldsymbol{p_i}, \boldsymbol{p_i'}) = |x_i + T_x - x_i'|$$

$$f_y(\boldsymbol{\theta}, \boldsymbol{p_i}, \boldsymbol{p_i'}) = |y_i + T_y - y_i'|$$

Under a particular model, a correspondence $(\boldsymbol{p_i}, \boldsymbol{p_i'})$ is considered an inlier if and only if both cost components are less than or equal to a given threshold δ. That is, $f_x(\boldsymbol{\theta}, \boldsymbol{p_i}, \boldsymbol{p_i'}) \leq \delta$ and $f_y(\boldsymbol{\theta}, \boldsymbol{p_i}, \boldsymbol{p_i'}) \leq \delta$. The optimization problem is to find model parameters $\boldsymbol{\theta}$ that maximize the number of inliers. Formally:

$$\max_{\boldsymbol{\theta}} \quad \operatorname{card}(S_I) \tag{1a}$$

$$\text{s.t.} \quad f_x(\boldsymbol{\theta}, \boldsymbol{p_i}, \boldsymbol{p_i'}) \leq \delta, \forall i \in S_I \subseteq S \tag{1b}$$

$$\text{and} \quad f_y(\boldsymbol{\theta}, \boldsymbol{p_i}, \boldsymbol{p_i'}) \leq \delta, \forall i \in S_I \subseteq S \tag{1c}$$

Equivalently,

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & \operatorname{card}(S_I) && \text{(2a)} \\ \text{s.t.} \quad & |x_i + T_x - x_i'| \leq \delta, \forall i \in S_I \subseteq S && \text{(2b)} \\ \text{and} \quad & |y_i + T_y - y_i'| \leq \delta, \forall i \in S_I \subseteq S && \text{(2c)} \end{aligned}$$

3 Problem Reformulation for Upper Bound

This problem is intractable to solve directly. However, reformulating the problem and applying linear programming yields a tractable solution for the upper bound.

3.1 Binary Variable Introduction

Introduce an identification binary variable z_i

$$z_i = \begin{cases} 1 & \text{if } f_x(\boldsymbol{\theta}, \boldsymbol{p_i}, \boldsymbol{p_i'}) \leq \delta \quad \text{and} \quad f_y(\boldsymbol{\theta}, \boldsymbol{p_i}, \boldsymbol{p_i'}) \leq \delta \\ 0 & \text{otherwise} \end{cases}$$

Equation (2) becomes:

$$\begin{aligned} \max_{\mathbf{z}, \boldsymbol{\theta}} \quad & \sum_{i=1}^{N} z_i && \text{(3a)} \\ \text{s.t.} \quad & z_i |x_i + T_x - x_i'| \leq z_i \delta, \forall i \in S_I \subseteq S && \text{(3b)} \\ \text{and} \quad & z_i |y_i + T_y - y_i'| \leq z_i \delta, \forall i \in S_I \subseteq S && \text{(3c)} \\ \text{and} \quad & z_i \in \{0, 1\}, \forall i = 1..N && \text{(3d)} \end{aligned}$$

To remove the binary variable, obverve that the following relaxation is equivalent:

$$\begin{aligned} \max_{\mathbf{z}, \boldsymbol{\theta}} \quad & \sum_{i=1}^{N} z_i && \text{(4a)} \\ \text{s.t.} \quad & z_i |x_i + T_x - x_i'| \leq z_i \delta, \forall i \in S_I \subseteq S && \text{(4b)} \\ \text{and} \quad & z_i |y_i + T_y - y_i'| \leq z_i \delta, \forall i \in S_I \subseteq S && \text{(4c)} \\ \text{and} \quad & z_i \in [0, 1], \forall i = 1..N && \text{(4d)} \end{aligned}$$

Conceptually, this follows because fractional values for any z_i will not occur. Consider a replacement argument as proof. Replacing any fractional z_i with 1 would not impact the inequalities while yielding a larger overall sum.

3.2 Address Bilinearity

Next, reformulate the constraint containing T_x in equation (4b) (Note that $\forall i \in S_I \subseteq S$ is assumed and omitted for clarity).

$$
\begin{aligned}
& z_i|x_i + T_x - x_i'| \leq z_i\delta && \text{(5a)}\\
\Longrightarrow \quad & -z_i\delta \leq z_i(x_i + T_x - x_i') \leq z_i\delta && \text{(5b)}\\
\Longrightarrow \quad & -z_i\delta \leq z_i(x_i) + z_i(T_x) - z_i(x_i') \leq z_i\delta && \text{(5c)}
\end{aligned}
$$

To address the bilinear term $z_i(T_x)$, introduce the auxiliary variable $w_{ix} = z_i(T_x)$:

$$\Longrightarrow \quad -z_i\delta \leq z_i(x_i) + w_{ix} - z_i(x_i') \leq z_i\delta \tag{6a}$$

Now consider a bilinear relaxation by concave and convex envelopes. The equality $w_{ix} = z_iT_x$ with bounding boxes $[\underline{z_i}, \overline{z_i}]$ and $[\underline{T_x}, \overline{T_x}]$ is relaxed by the following envelopes:

$$
\begin{aligned}
w_{ix} &\geq max(\underline{z_i}T_x + \underline{T_x}z_i - \underline{z_i}\underline{T_x}, \quad \overline{z_i}T_x + \overline{T_x}z_i - \overline{z_i}\overline{T_x}) && \text{(7a)}\\
w_{ix} &\leq min(\overline{z_i}T_x + \underline{T_x}z_i - \overline{z_i}\underline{T_x}, \quad \underline{z_i}T_x + \overline{T_x}z_i - \underline{z_i}\overline{T_x}) && \text{(7b)}
\end{aligned}
$$

Given that $[\underline{z_i}, \overline{z_i}] = [0, 1]$, this can be simplified to:

$$
\begin{aligned}
w_{ix} &\geq max(\underline{T_x}z_i, \quad T_x + \overline{T_x}z_i - \overline{T_x}) && \text{(8a)}\\
w_{ix} &\leq min(T_x + \underline{T_x}z_i - \underline{T_x}, \quad \overline{T_x}z_i) && \text{(8b)}
\end{aligned}
$$

Decomposing the *max* and *min* functions gives:

$$
\begin{aligned}
w_{ix} &\geq \underline{T_x}z_i && \text{(9a)}\\
w_{ix} &\geq T_x + \overline{T_x}z_i - \overline{T_x} && \text{(9b)}\\
w_{ix} &\leq T_x + \underline{T_x}z_i - \underline{T_x} && \text{(9c)}\\
w_{ix} &\leq \overline{T_x}z_i && \text{(9d)}
\end{aligned}
$$

Following an identical procedure for the T_y constraint in equation (4c) yields:

$$
\begin{aligned}
w_{iy} &\geq \underline{T_y}z_i && \text{(10a)}\\
w_{iy} &\geq T_y + \overline{T_y}z_i - \overline{T_y} && \text{(10b)}\\
w_{iy} &\leq T_y + \underline{T_y}z_i - \underline{T_y} && \text{(10c)}\\
w_{iy} &\leq \overline{T_y}z_i && \text{(10d)}
\end{aligned}
$$

3.3 Formulate as Linear Programming

Given fixed bounds for $\boldsymbol{T}$ of $[\underline{T_x}, \overline{T_x}]$ and $[\underline{T_y}, \overline{T_y}]$, linear programming can now solve for the upper bound of this relaxed version. To do so, rearrange the equations above in the canonical form:

$$\min_{\mathbf{x}} \mathbf{c^T x}$$
$$\text{s.t. } A\mathbf{x} \leq \mathbf{b}$$
$$\text{and } \mathbf{l_b} \leq \mathbf{x} \leq \mathbf{u_b}$$
$$\text{where } \mathbf{x} = (T_x, T_y, z_1, ..., z_n, w_{1x}, w_{1y}, ..., w_{ny})^T$$

Equation (6a) can be reformulated:

$$z_i x_i + w_{ix} - z_i x_i' \geq -z_i \delta \tag{11a}$$
$$z_i x_i + w_{ix} - z_i x_i' \leq z_i \delta \tag{11b}$$

Rearranging and grouping terms:

$$(x_i' - x_i - \delta) z_i - w_{ix} \leq 0 \impliedby (11a) \tag{12a}$$
$$(x_i - x_i' - \delta) z_i + w_{ix} \leq 0 \impliedby (11b) \tag{12b}$$

For brevity, let $r_{xi} = (x_i' - x_i - \delta)$ and $r_{xi}' = (x_i - x_i' - \delta)$:

$$r_{xi} z_i - w_{ix} \leq 0 \tag{13a}$$
$$r_{xi}' z_i + w_{ix} \leq 0 \tag{13b}$$

Equivalently for y components, it follows that:

$$r_{yi} z_i - w_{iy} \leq 0 \tag{14a}$$
$$r_{yi}' z_i + w_{iy} \leq 0 \tag{14b}$$
$$\text{where } r_{yi} = (y_i' - y_i - \delta) \tag{14c}$$
$$\text{and } r_{yi}' = (y_i - y_i' - \delta) \tag{14d}$$

These constraints can now be represented in the canonical form:

$$\left[\begin{array}{cc|ccc|ccccc} 0 & 0 & r_{x1} & \cdots & 0 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & r_{x1}' & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & r_{y1} & \cdots & 0 & 0 & -1 & \cdots & 0 & 0 \\ 0 & 0 & r_{y1}' & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & r_{x1} & -1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & r_{x1}' & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & r_{y1} & 0 & -1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & r_{y1}' & 0 & 1 & \cdots & 0 & 0 \end{array}\right] \left[\begin{array}{c} T_x \\ T_y \\ \hline z_1 \\ \vdots \\ z_n \\ \hline w_{1x} \\ w_{1y} \\ \vdots \\ w_{nx} \\ w_{ny} \end{array}\right] \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{15}$$

Next, restructure the inequalities in equation (9):

$$\underline{T_x} z_i - w_{ix} \leq 0 \Longleftarrow \text{equation (9a)} \tag{16a}$$
$$T_x + \overline{T_x} z_i - w_{ix} \leq \overline{T_x} \Longleftarrow \text{equation (9b)} \tag{16b}$$
$$-T_x - \underline{T_x} z_i + w_{ix} \leq -\underline{T_x} \Longleftarrow \text{equation (9c)} \tag{16c}$$
$$-\overline{T_x} z_i + w_{ix} \leq 0 \Longleftarrow \text{equation (9d)} \tag{16d}$$

These constraints can be represented in canonical form:

$$\left[\begin{array}{cc|ccc|ccccc}
0 & 0 & \underline{T_x} & \cdots & 0 & -1 & 0 & \cdots & 0 & 0 \\
1 & 0 & \overline{T_x} & \cdots & 0 & -1 & 0 & \cdots & 0 & 0 \\
-1 & 0 & -\underline{T_x} & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & -\overline{T_x} & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \underline{T_x} & 0 & 0 & \cdots & -1 & 0 \\
1 & 0 & 0 & \cdots & \overline{T_x} & 0 & 0 & \cdots & -1 & 0 \\
-1 & 0 & 0 & \cdots & -\underline{T_x} & 0 & 0 & \cdots & 1 & 0 \\
0 & 0 & 0 & \cdots & -\overline{T_x} & 0 & 0 & \cdots & 1 & 0
\end{array}\right]
\left[\begin{array}{c}
T_x \\ T_y \\ \hline z_1 \\ \vdots \\ z_n \\ \hline w_{1x} \\ w_{1y} \\ \vdots \\ w_{nx} \\ w_{ny}
\end{array}\right]
\leq
\left[\begin{array}{c}
0 \\ \overline{T_x} \\ -\underline{T_x} \\ 0 \\ \vdots \\ 0 \\ \overline{T_x} \\ -\underline{T_x} \\ 0
\end{array}\right] \tag{17}$$

Equivalently for the y constraints defined in equation (10):

$$\underline{T_y} z_i - w_{iy} \leq 0 \Longleftarrow \text{equation (10a)} \tag{18a}$$
$$T_y + \overline{T_y} z_i - w_{iy} \leq \overline{T_y} \Longleftarrow \text{equation (10b)} \tag{18b}$$
$$-T_y - \underline{T_y} z_i + w_{iy} \leq -\underline{T_y} \Longleftarrow \text{equation (10c)} \tag{18c}$$
$$-\overline{T_y} z_i + w_{iy} \leq 0 \Longleftarrow \text{equation (10d)} \tag{18d}$$

In canonical form:

$$\left[\begin{array}{cc|ccc|ccccc}
0 & 0 & \underline{T_y} & \cdots & 0 & 0 & -1 & \cdots & 0 & 0 \\
0 & 1 & \overline{T_y} & \cdots & 0 & 0 & -1 & \cdots & 0 & 0 \\
0 & -1 & -\underline{T_y} & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 \\
0 & 0 & -\overline{T_y} & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \underline{T_y} & 0 & 0 & \cdots & 0 & -1 \\
0 & 1 & 0 & \cdots & \overline{T_y} & 0 & 0 & \cdots & 0 & -1 \\
0 & -1 & 0 & \cdots & -\underline{T_y} & 0 & 0 & \cdots & 0 & 1 \\
0 & 0 & 0 & \cdots & -\overline{T_y} & 0 & 0 & \cdots & 0 & 1
\end{array}\right]
\left[\begin{array}{c}
T_x \\ T_y \\ \hline z_1 \\ \vdots \\ z_n \\ \hline w_{1x} \\ w_{1y} \\ \vdots \\ w_{nx} \\ w_{ny}
\end{array}\right]
\leq
\left[\begin{array}{c}
0 \\ \overline{T_y} \\ -\underline{T_y} \\ 0 \\ \vdots \\ 0 \\ \overline{T_y} \\ -\underline{T_y} \\ 0
\end{array}\right] \tag{19}$$

Next, define lower and upper bounds in the form $\boldsymbol{l_b} \leq \boldsymbol{x} \leq \boldsymbol{u_b}$:

$$\begin{bmatrix} \underline{T_x} \\ \underline{T_y} \\ \hline 0 \\ \vdots \\ 0 \\ \hline -\infty \\ -\infty \\ \vdots \\ -\infty \\ -\infty \end{bmatrix} \leq \begin{bmatrix} T_x \\ T_y \\ \hline z_1 \\ \vdots \\ z_n \\ \hline w_{1x} \\ w_{1y} \\ \vdots \\ w_{nx} \\ w_{ny} \end{bmatrix} \leq \begin{bmatrix} \overline{T_x} \\ \overline{T_y} \\ \hline 1 \\ \vdots \\ 1 \\ \hline \infty \\ \infty \\ \vdots \\ \infty \\ \infty \end{bmatrix} \tag{20}$$

Lastly, define the objective vector $\boldsymbol{c}$. The original problem is defined as a maximization over the sum of $z_i \forall i$. To formulate this as a minimization problem, set all entries corresponding to $\boldsymbol{z}$ to -1, and set all other entries to 0:

$$\boldsymbol{c} = \left[\begin{array}{cc|ccc|cccccc} 0 & 0 & -1 & \cdots & -1 & 0 & 0 & \cdots & 0 & 0 \end{array}\right] \tag{21}$$

4 Complete Solution with Branch and Bound

The above linear programming system provides an upper bound on the number of inliers given some $\underline{T_x}, \overline{T_x}, \underline{T_y}, \overline{T_y}$. However, it does not offer a global solution. A global solution can be found by solving this linear system iteratively in a branch and bound algorithm.

For images with width w and height h, consider the initial search space of $\underline{T_x} = -w$, $\overline{T_x} = w$, $\underline{T_y} = -h$ and $\overline{T_y} = h$. Initialize default bounds for the inlier set cardinality of $[0, N]$. Add these values to a priority queue that sorts items primarily by the cardinality set upper bound and secondarily by the cardinality set lower bound. Additionally, track the best-identified lower bound and corresponding model parameters.

So long as the priority queue is not empty, take the highest priority (most promising) item. If the upper bound is below the best-identified lower bound, move on to the next item in the queue. Split this search space into two smaller search spaces with a partition in the center of the larger dimension. For each of the smaller search spaces, compute the upper bound for the inlier set cardinality using the above linear programming system. Test the generated model parameters directly to compute a lower bound. If the lower bound is the best seen so far, update the best lower bound and best model accordingly. Add each newly generated search space to the priority queue if at least one of the dimensions has a size greater than 1.

Throughout this algorithm, track the identified inlier set cardinality upper and lower bounds at each iteration.

For comparison, a baseline naive upper bound is used instead of the LP solution. This naive baseline applies the upper and lower bounding models to each point in the left image and checks to determine if the corresponding point in the right image is contained within the resulting bounding box. While this approach correctly determines the upper bound inlier set cardinality, it yields a much less strict upper bound, resulting in slower convergence.

5 Results

Translational model: (Tx, Ty) = T(x=-232.0, y=-154.0)
Inlier set S_I: [2 7 8 14 15 19 25 30 31 33 34 39 41 44 50] (15 points)

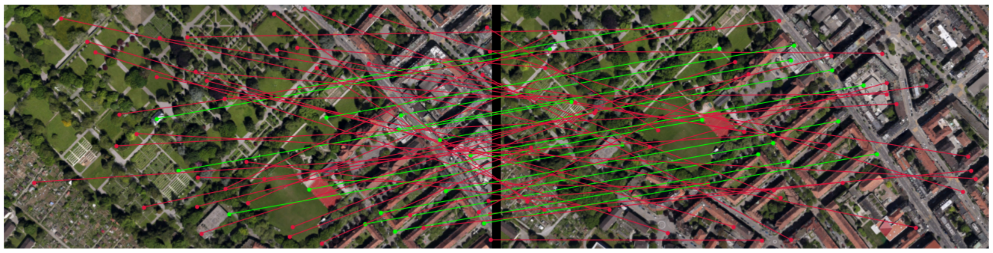

Figure 1: Inlier (green) and outlier (red) correspondences identified in both approaches

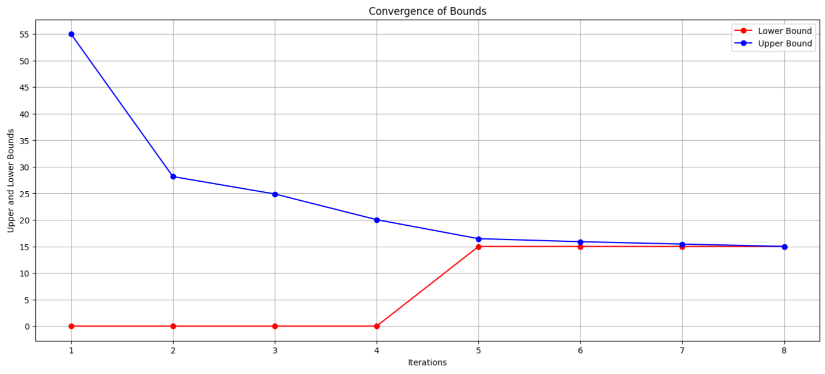

Figure 2: Convergence of bounds for the Linear Programming Upper Bound approach

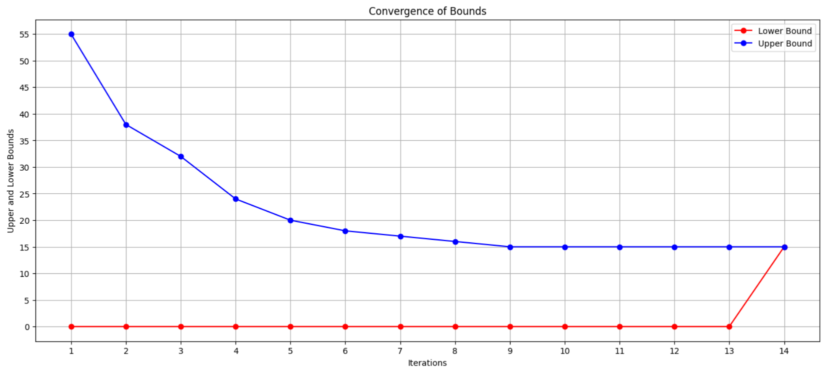

Figure 3: Convergence of bounds for the Naive Upper Bound Baseline approach

6 Discussion

Both approaches yield the same final solution regarding the translation model (up to rounding) and identified inlier set. The Linear Programming solution enables convergence in 8 iterations rather than 14 with the naive approach. However, it is worth noting that despite running in fewer iterations, the overhead associated with linear programming causes the runtimes to be very comparable. Thus, for this specific problem, Branch and Bound is potentially ill-suited. However, in other situations where an effective and efficient-to-compute naive baseline is not as readily available, the decreased iteration count may be crucial in achieving a tractable runtime.