Robust Global Translations with 1DSfM

Kyle Wilson and Noah Snavely
ECCV14

Aleksander Holynski
UW CSE Vision Seminar

Problem Statement

- Incremental SfM is expensive and error-prone.
- It requires repeated nonlinear model refinement (bundle adjustment).
- First images are weighted more heavily, can fall into local minima.
- Why not solve SfM globally?

Global SfM methods usually follow the same *structure*

1. Find $\mathbf{R_i}$, the rotation matrix that maps world coordinates to camera coordinates for all $\mathbf{i}$.
2. Calculate $\mathbf{t^l_{ij}}$, the relative translation between cameras $\mathbf{i}$ and $\mathbf{j}$ in the local coordinate system.
3. Given the estimate for $\mathbf{R_i}$, calculate $\mathbf{t_{ij} = R_i t^l_{ij}}$, the same translation in the world coordinate frame.
4. Find a consistent embedding of this set of translations using some optimization method and loss function.

Global SfM faces its own challenges

- Sequential SfM has the ability to filter out measurements after each step, something that's not applicable to global methods. Since outliers can reduce the solution quality and prevent convergence, it is important to find a way to identify and discard them.

- Most methods of solving the translations problem are non-convex and unreliable.

Related Works

- Lots of iterative SfM methods, not too many global ones.
- Of the global methods, there has been significant work done on calculating the rotation matrices $\mathbf{R}_i$. This paper does not attempt to improve solutions to this subproblem, instead employing methods from Chatterjee et and Govindu.
- Many methods in solving the translation problem are inefficient or do not explicitly remove outliers. There are two main methods mentioned: looking at loop closure in the graph cycles, or the use of a robust cost function.

Contributions

1. A method for preprocessing the problem instance to remove outlier measurements, called **1DSfM**.

2. A new approach to solving the translations problem using a nonlinear optimization method.

Where does this fall in the scope of the problem?

1. Find $\mathbf{R_i}$, the rotation matrix that maps world coordinates to camera coordinates for all **i.**

2. Calculate $\mathbf{t^l_{ij}}$, the relative translation between cameras **i** and **j** in the local coordinate system.

3. Given the estimate for $\mathbf{R_i}$, calculate $\mathbf{t_{ij} = R_i t^l_{ij}}$, the same translation in the world coordinate frame.

4. Find a consistent embedding of this set of translations using some optimization method and loss function.

Where does this fall in the scope of the problem?

1. Find $\mathbf{R_i}$, the rotation matrix that maps world coordinates to camera coordinates for all **i.**

2. Calculate $\mathbf{t^l_{ij}}$, the relative translation between cameras **i** and **j** in the local coordinate system.

3. Given the estimate for $\mathbf{R_i}$, calculate $\mathbf{t_{ij} = R_i t^l_{ij}}$, the same translation in the world coordinate frame.

- **1DSfM**

4. Find a consistent embedding of this set of translations using some optimization method and loss function.

Where does this fall in the scope of the problem?

1. Find $\mathbf{R_i}$, the rotation matrix that maps world coordinates to camera coordinates for all **i.**

2. Calculate $\mathbf{t^l_{ij}}$, the relative translation between cameras **i** and **j** in the local coordinate system.

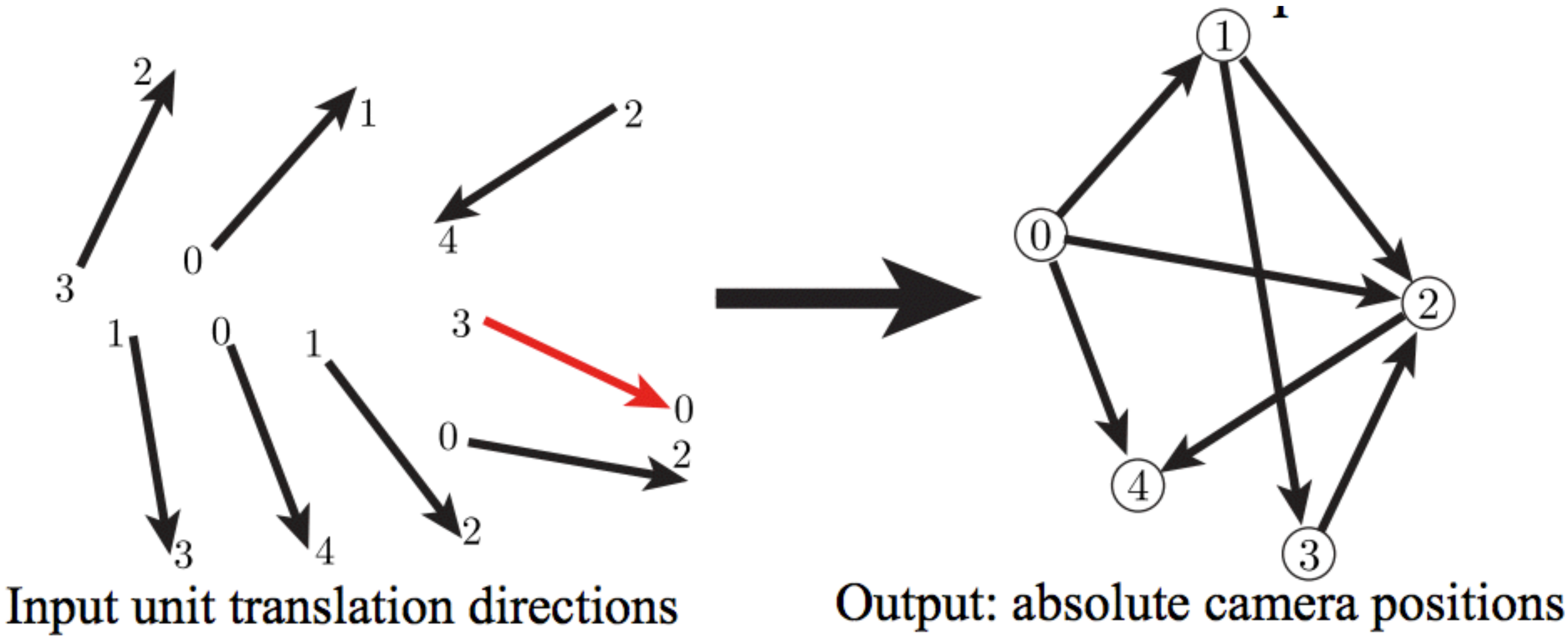
3. Given the estimate for $\mathbf{R_i}$, calculate $\mathbf{t_{ij} = R_i t^l_{ij}}$, the same translation in the world coordinate frame.

- 1DSfM

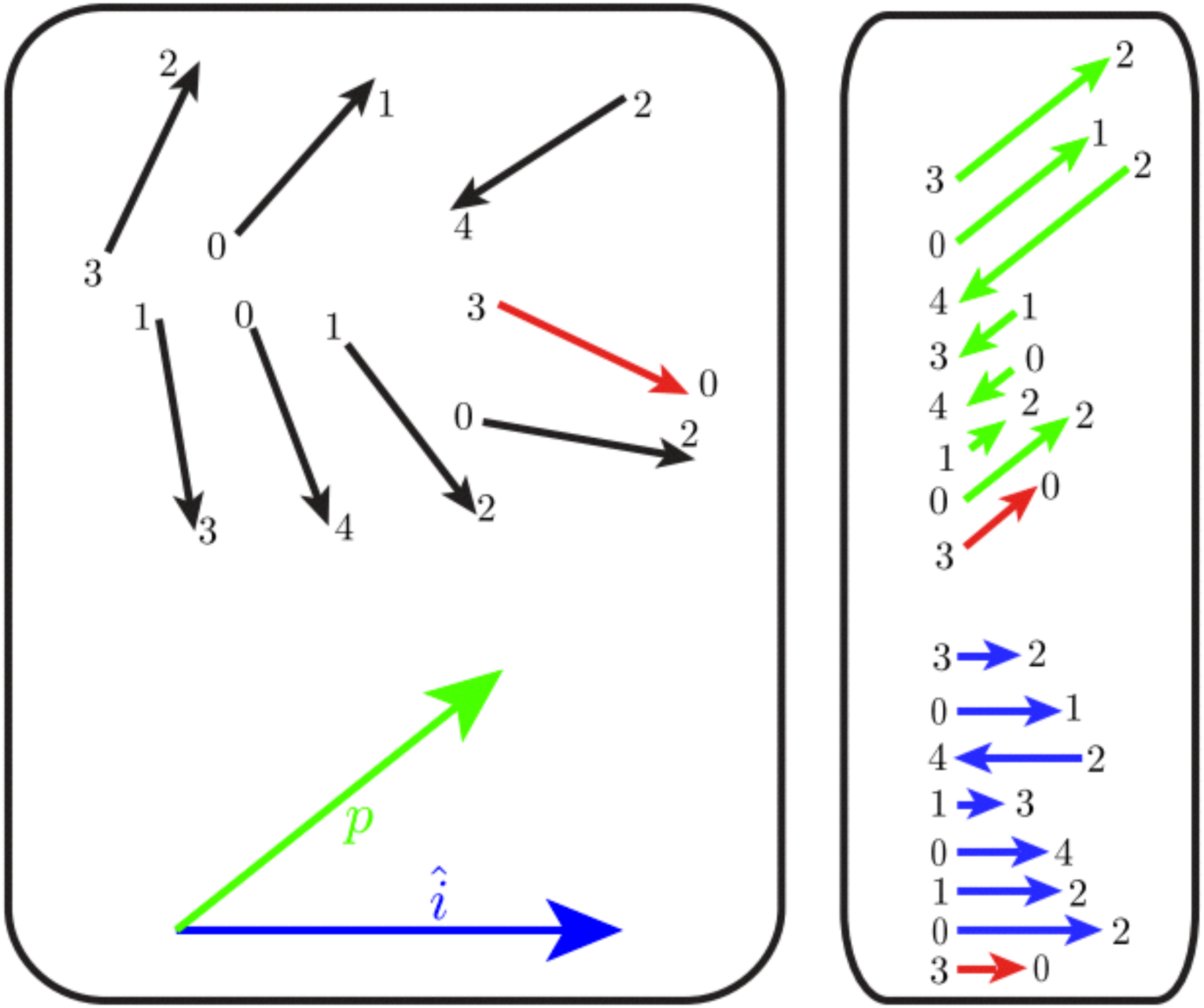
4. Translation solver

Scope of 1DSfM+Solver

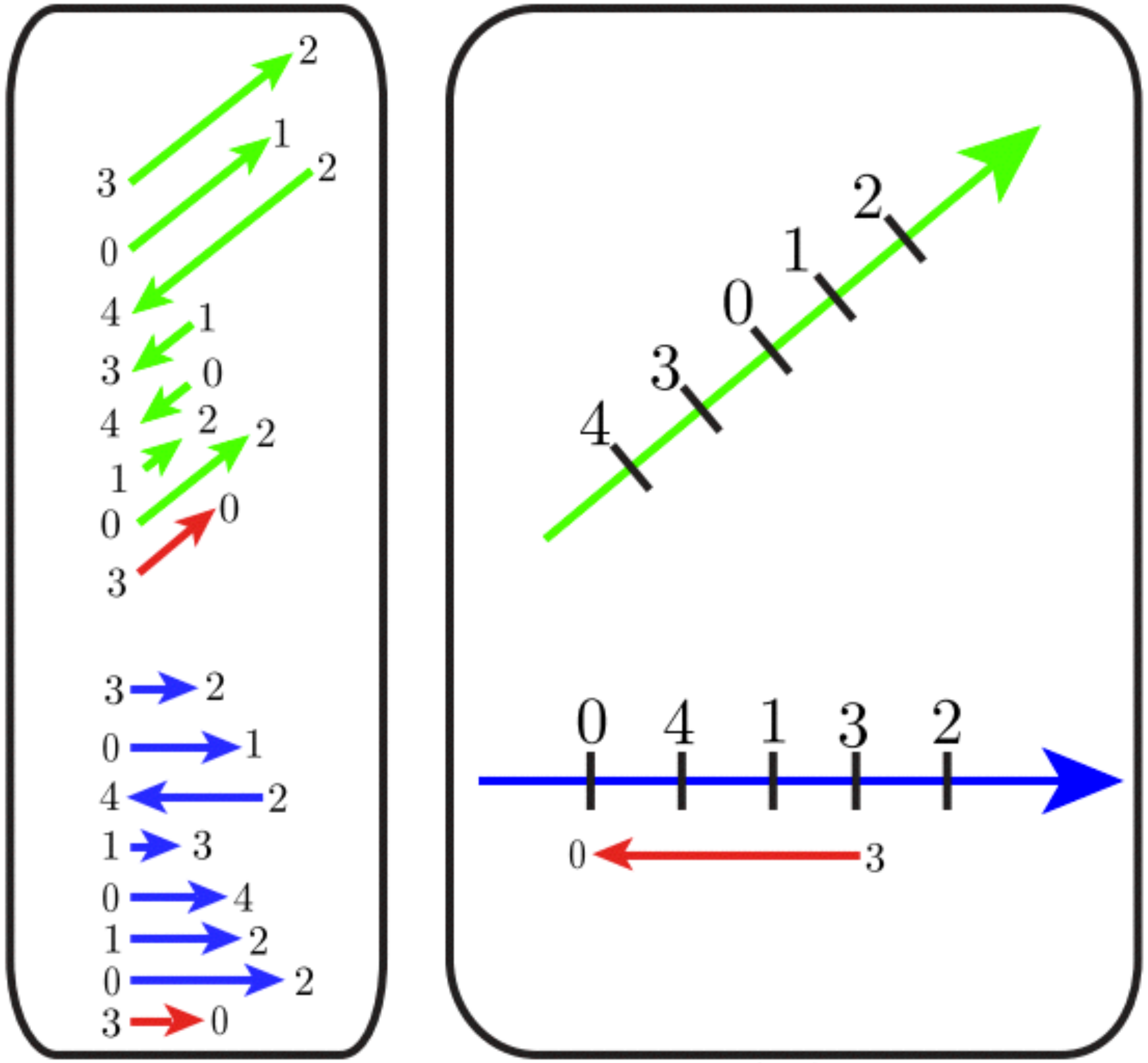
- **Input:** unit relative translations $\mathbf{t}^{I}_{ij}$ between cameras in the global coordinate system
- **Output:** absolute camera positions



- Given the output (right), it is easy to tell which input translation is an outlier, as is done in iterative SfM.
- The goal here is to do the same before constructing the output embedding.



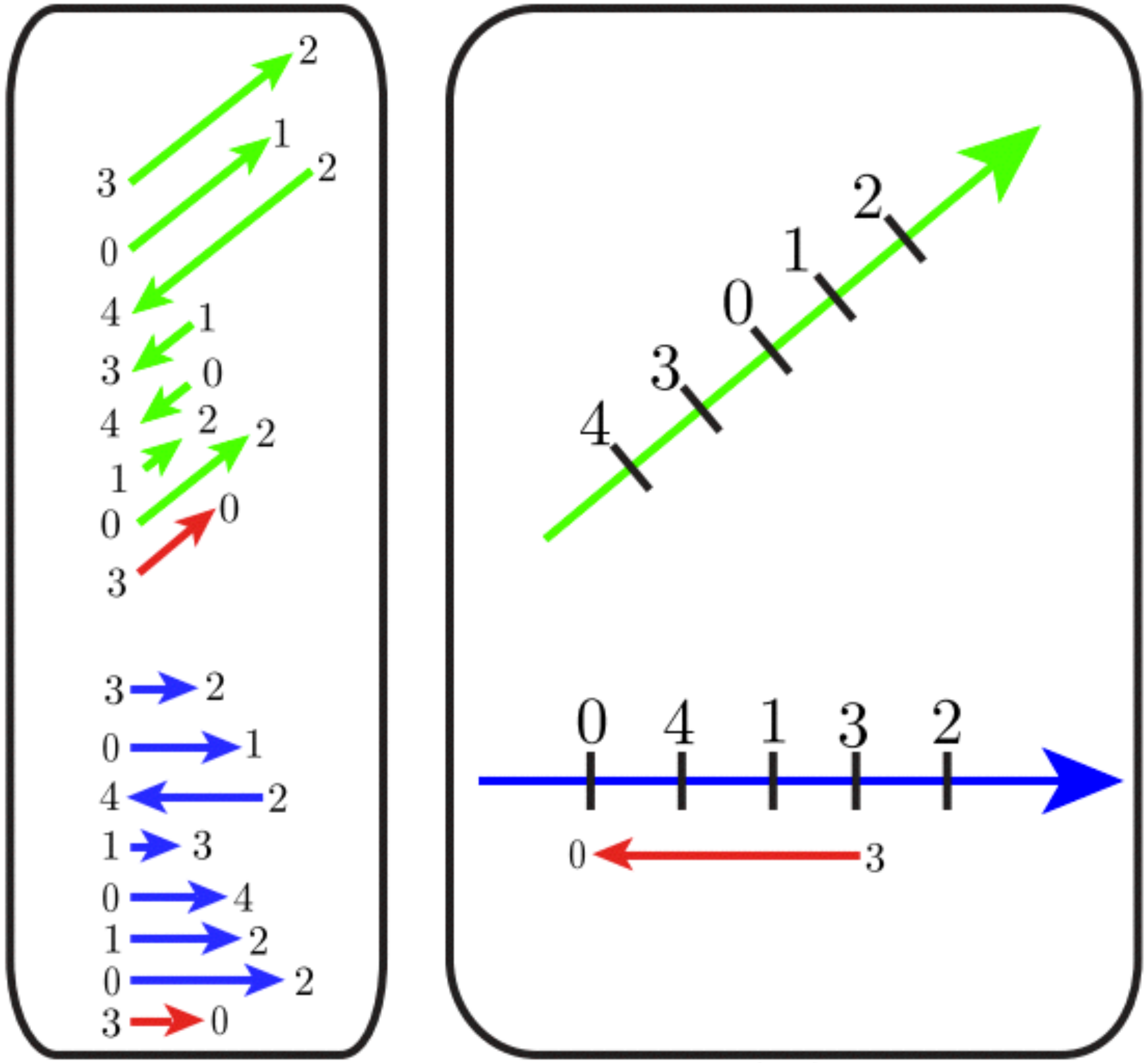
- Input vectors are projected into a number of directions.



- We want an ordering like the right image, where the direction of each vector on the left is consistent with the final ordering.

- This is equivalent to a DAG.

- It is not always possible.



- So, we want to construct a DAG from a probably non-acyclic (cyclic?) graph.
- This is an instance of the Minimum Feedback Arc Set problem, which is NP-Complete “but there is a rich literature of approximation algorithms”
- Not every direction is guaranteed to correctly produce outliers, so many directions have to be used.

Translation Solver

- Input: the output of 1DSfM, a graph where cameras are vertices and translations between them are edges.
- Goal: Find an embedding of this graph, such that the translation directions are close to $\mathbf{t}$

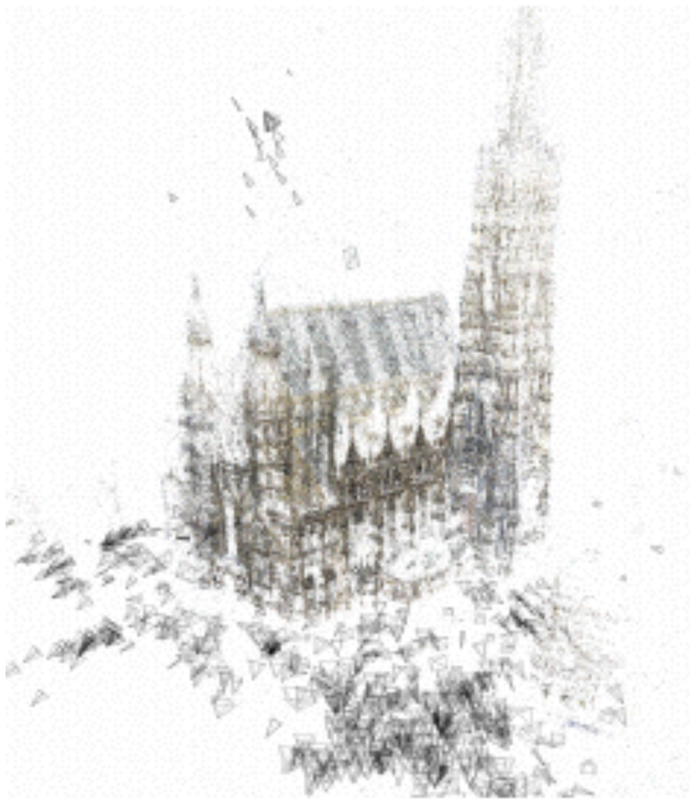
Loss function

Name	Formula	Equivalent
Geodesic	$\angle(\mathbf{u}, \mathbf{v})$	θ
Cross Product	$\mathbf{u} \times \mathbf{v}$	$\sin\theta$
Inner Product	$1 - \mathbf{u} \cdot \mathbf{v}$	$2\sin^2\frac{\theta}{2}$
Squared Chordal Distance	$\|\mathbf{u} - \mathbf{v}\|^2$	$4\sin^2\frac{\theta}{2}$

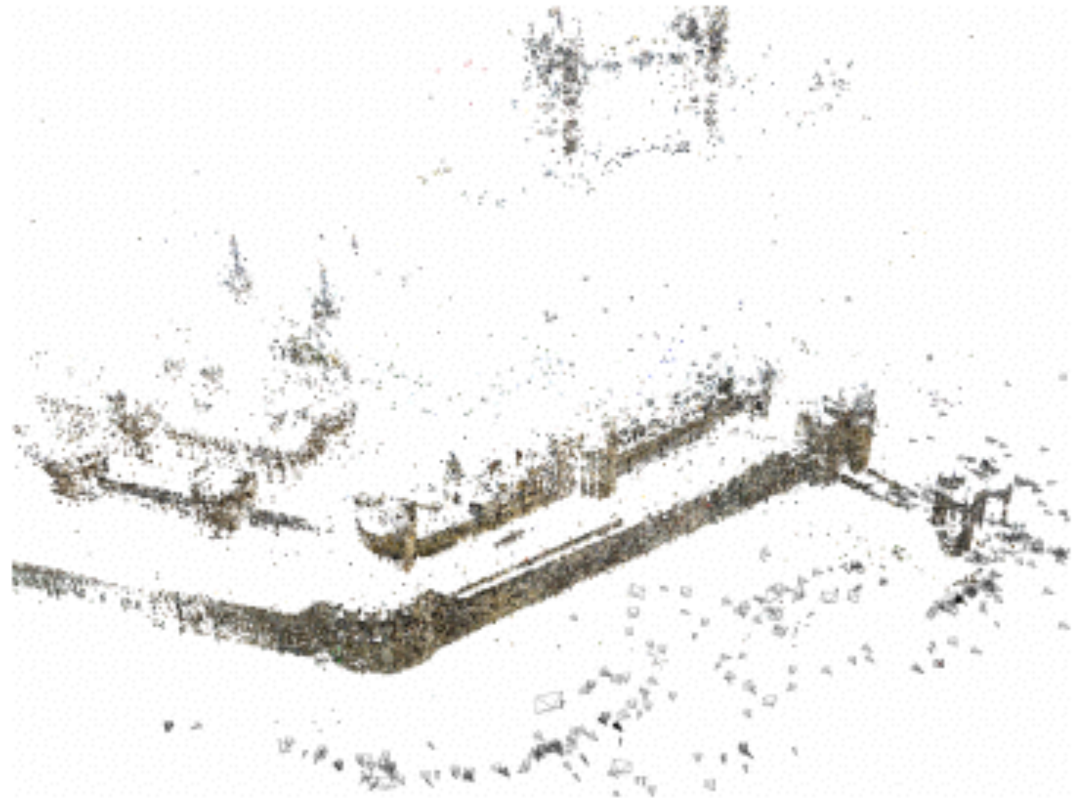
$$err_{ch}(\mathcal{T}) = \sum_{(i,j)\in E} d_{ch}\left(\hat{\mathbf{t}}_{ij}, \frac{\mathbf{t}_j - \mathbf{t}_i}{\|\mathbf{t}_j - \mathbf{t}_i\|}\right)^2$$

$$d_{ch}(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_2$$

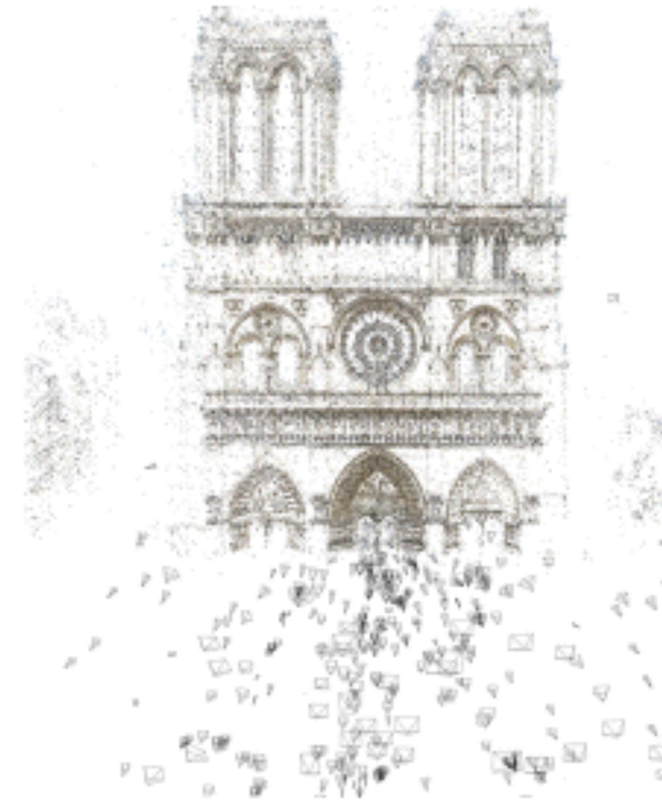
- The final metric used in the loss function is the squared chordal distance.
- This is a nonlinear least squares problem, so it is not guaranteed to converge at the global minimum, and good initialization is important.
- $\mathbf{t}_i$ are absolute camera locations



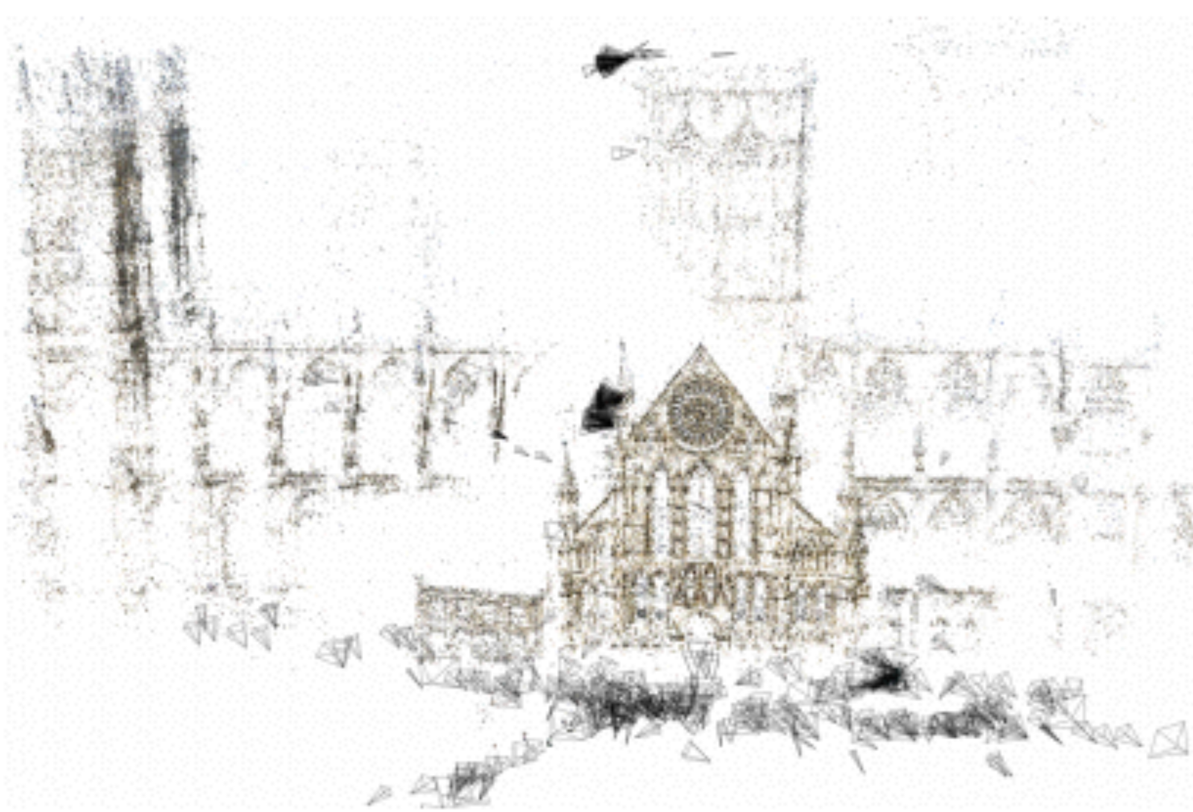
Vienna Cathedral



Tower of London



Notre Dame



Yorkminster

Alamo

			without 1DSfM				with 1DSfM				Robust Loss						[11]	
			no BA	BA			no BA	BA			BA			1DSfM+BA			BA	
Name	Size	N_c	$\tilde{x}$	N_c	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	N_c	$\tilde{x}$	$\bar{x}$	N_c	$\tilde{x}$	$\bar{x}$	N_c	$\tilde{x}$	$\bar{x}$	N_c	$\tilde{x}$
Piccadilly	80	2152	3.2	1905	1.0	9e3	4.1	1932	0.6	**5e1**	1965	**0.3**	9e3	1956	0.7	7e2	1638	10
Union Square	300	789	9.9	700	3.3	3e3	5.6	702	3.5	5e2	699	3.2	2e2	710	3.4	**9e1**	521	10
Roman Forum	200	1084	6.9	973	1.5	3e4	6.1	981	0.3	4e1	1000	2.7	9e5	989	**0.2**	**3e0**	840	37
Vienna Cathedral	120	836	5.5	758	0.9	9e3	6.6	757	0.5	**8e3**	770	0.7	7e4	770	**0.4**	2e4	652	12
Piazza del Popolo	60	328	1.8	311	**1.2**	2e1	3.1	303	2.6	**4e0**	317	1.6	9e1	308	2.2	2e2	93	16
NYC Library	130	332	1.7	297	1.5	7e2	2.5	292	0.9	2e1	307	**0.2**	8e1	295	0.4	**1e0**	271	1.4
Alamo	70	577	1.0	528	0.2	7e3	1.1	521	0.3	**7e0**	541	0.2	7e5	529	0.3	2e7	422	2.4
Metropolis	200	341	6.0	282	0.5	1e3	9.9	288	1.2	**9e0**	292	0.6	3e1	291	0.5	7e1	240	18
Yorkminster	150	437	7.0	405	0.2	3e3	3.4	395	0.2	1e4	416	0.4	9e3	401	0.1	**5e2**	345	6.7
Montreal N.D.	30	450	0.9	431	0.2	4e3	2.5	425	0.9	1e0	431	**0.1**	**4e-1**	427	0.4	1e0	357	9.8
Tower of London	300	572	9.4	417	1.1	2e3	11	414	0.4	3e3	427	**0.2**	3e4	414	1.0	**4e1**	306	44
Ellis Island	180	227	4.1	211	0.4	4e0	3.7	213	0.4	4e1	213	0.3	3e0	214	0.3	3e0	203	8.0
Notre Dame	300	553	19	524	**0.7**	2e4	10	500	2.1	**7e0**	530	0.8	7e4	507	1.9	**7e0**	473	2.1

- BA refers to performing a final bundle adjustment
- The translations solver is used for all four methods
- $\tilde{\mathbf{x}}$ is the median error, $\bar{\mathbf{x}}$ is the mean error
- Robust loss refers to the use of a robust cost function (Huber loss) for the removal of outliers. This method often increases the average error while decreasing the median.
- The last column is the baseline method, which solves the translations problem by minimizing the cross product of solution translations with input pairwise translations
- Error is with respect to models produced by Bundler (which is a sequential method)

Name	T_R	without 1DSfM T_S	T_{BA}	Σ	with 1DSfM T_O	T_S	T_{BA}	Σ	using [11] T_S	T_{BA}	Σ	using [20] T
Piccadily	570	177	3252	3999	122	366	2425	3483	9497	1046	11113	44369
Union Square	17	71	401	489	20	75	340	452	277	150	444	1244
Roman Forum	37	104	1733	1874	40	135	1245	1457	290	694	1021	4533
Vienna Cathedral	98	225	3611	3934	60	144	2837	3139	1282	893	2273	10276
Piazza del Popolo	14	28	213	255	9	35	191	249	98	26	138	1287
NYC Library	9	38	382	429	13	54	392	468	21	190	220	3807
Alamo	56	96	646	798	29	73	752	910	1039	308	1403	1654
Madrid Metropolis	15	32	224	271	8	20	201	244	57	67	139	1315
Yorkminster	11	60	955	1026	18	93	777	899	81	302	394	3225
Montreal Notre Dame	17	76	1043	1136	22	75	1135	1249	25	382	424	2710
Tower of London	9	52	750	811	14	55	606	648	17	238	264	1900
Ellis Island	12	17	276	305	7	13	139	171	7	108	127	1191
Notre Dame	53	152	2139	2344	42	59	1445	1599	299	841	1193	6154

Table 3. Timing information, in seconds for the results in Table 2. Times are listed for solving for rotations with [5] (T_R), removing outliers with 1DSfM (T_O), running a translations problem solver (T_S), and for bundle adjustment (T_{BA}).

[5] is the rotation solver used for this paper

[20] is a sequential SfM method (Bundler)

Limitations

- Cases where epipolar geometries are sparse.
- Cases where outliers are self-consistent, due to strange or ambiguous structure in the scene.

Discussion

- Should nonlinear optimization being reconsidered as a tool for global structure-from-motion?
- Are there better methods of extracting outliers?