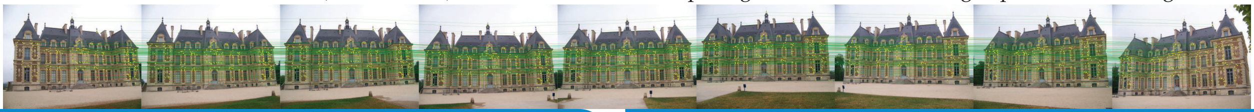
Unordered feature tracking made fast and easy

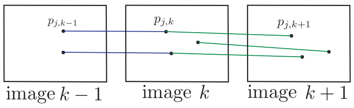
Pierre Moulon[1,2], Pascal Monasse[1]
[1]Université Paris-Est, LIGM (UMR CNRS), Center for Visual Computing, ENPC, [2]Mikros Image. pmo@mikrosimage.eu

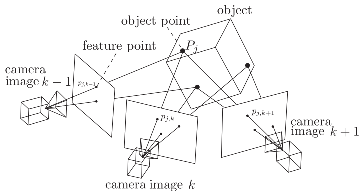
CVMP 2012



Motivation & Contribution

Considering n pairwise feature correspondences we want sets of corresponding matching features across multiple images, i.e., *tracks*.





Track identification in a set of images is an important task in computer vision. It allows solving geometry-related problems like:

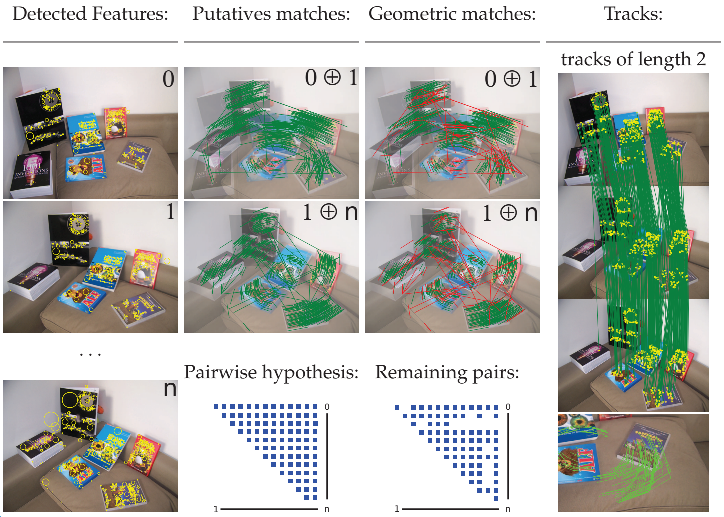
- region tracking, match-moving, video stabilization,
- image-stitching,
- structure from motion [4] and odometry.

We present an efficient algorithm to fuse two-view correspondences into multi-view consistent tracks that is:

- simple,
- efficient: lower computational complexity than [2, 3],
- reliable: able to identify all tracks, contrary to [2, 3].

The Track Computation Problem

The problem of feature point tracking consists in linking multi-view correspondences from pairwise matches that share a common point. It consists of a correspondence fusion problem.



Available Solutions

Available algorithms do not solve the problem in an optimal way:

Bundler [2]	ETH-V3D [3]
• requires a start image index • depends on image pair order	• requires many sorting operations • heavy memory consumption

Neither approach is able to identify all valid tracks.

Conclusion

Our approach is quasi-linear.
It does not depend on the image pair order.
Open source implementation is available in the openMVG library [4].

The Union-Find Solution

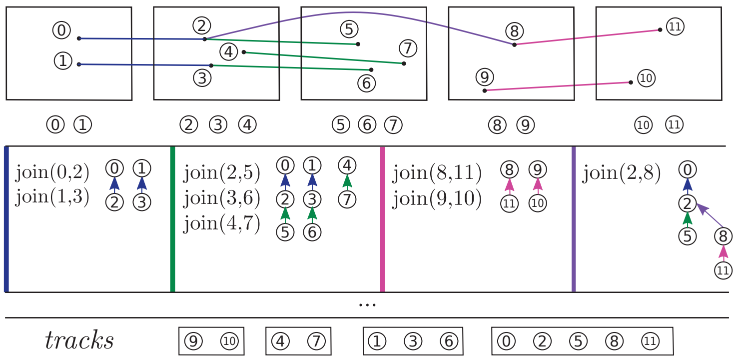
- We see the correspondence fusion problem as set unions,
- It can be solve efficiently using the Union-Find [1] algorithm.

Consider a graph $\mathcal{G}$:

- vertices: features $\{ImageId, FeatureId\}$ = **sets**,
- edges: correspondences $\{LeftFeature, RigthFeature\}$ = **sets union**,
- $\Rightarrow$ *Tracks* are connected components of $\mathcal{G}$.

Our algorithm enumerates edges (pairwise correspondences) and fuses the sets containing the edges' endpoints (features).

Algorithm 1 Unordered feature tracking
Input: list of pairwise correspondences
Output: tracks
  create a singleton for each feature
  for each pairwise match **do**
    join($leftMatch$, $rightMatch$)
  end for
  return each connected set as a track



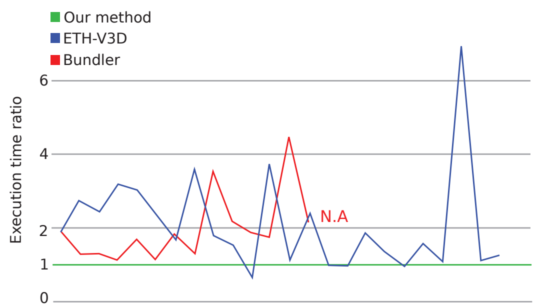
- **Union-Find algorithm allows efficient:**
  - fusion: $\Rightarrow$ join(a,b) = union(find(a), find(b)),
  - connected set search and traversal.

Complexity & Performance

Experimental results on small to large datasets (10 to 2,600 images), largest dataset with more than 41 million pairwise matches, yielding a million tracks.

	Bundler [2]	ETH-V3D [3]	Our approach
Complexity	$O(n \log n)$	$O(n \log n)$	$\mathbf{O(n\alpha(n))}$
Execution time ratio	1.57	2.09	**1.00**
Track count	78%	90%	**100%**
Lines of code (C++)	≈ 200	≈ 150	**≈ 100**

$\alpha(.)$ is the inverse Ackermann function, quasi-constant in practice.



References

[1] B. A. Galler and M. J. Fischer. An improved equivalence algorithm. In ACM, V7, I5, May 1964.
[2] N. Snavely, S. M. Seitz and R. Szeliski. Photo tourism: exploring photo collections in 3D. In SIGGRAPH 2006.
[3] C. Zach. ETH-V3D Structure-and-Motion software. © 2010-2011. ETH Zurich.
[4] P. Moulon, P. Monasse and R. Marlet. Adaptive Structure from Motion with *a contrario* model estimation. In *ACCV 2012*.