UNORDERED FEATURE TRACKING MADE FAST AND EASY

P. Moulon*[†], P. Monasse[†]

[*]Mikros Image, France, pmo@mikrosimage.eu
[†]Université Paris-Est, LIGM (UMR CNRS), Center for Visual Computing, ENPC, France, monasse@imagine.enpc.fr

Abstract

We present an efficient algorithm to fuse two-view correspondences into multi-view consistent tracks. The proposed method relies on the Union-Find [1] algorithm to solve the fusion problem. It is very simple and has a lower computational complexity than other available methods. Our experiments show that it is faster and computes more tracks.

Keywords: Tracking, Struture from Motion, computer vision.

Introduction. The problem of feature points tracking is to follow the position of a characteristic point in a set of images. These multi-view correspondences are called **tracks**. Track identification in a set of images (ordered, or not) is an important task in computer vision. It allows solving geometry-related problems like video-stabilization, tracking, match-moving, image-stitching, structure from motion and odometry.

The track problem. Considering n pairwise feature correspondences as input we want sets of corresponding matching features across multiple images, as illustrated in fig.1 with video frames.

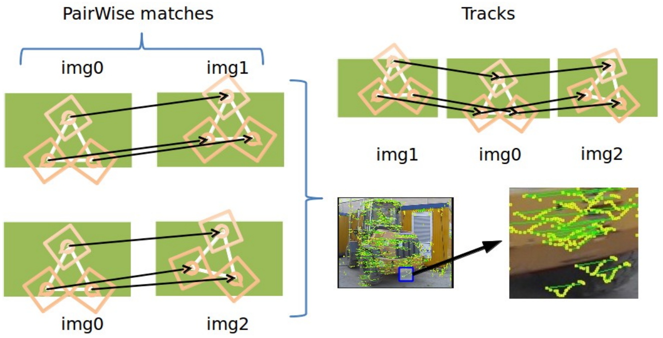

Figure 1: From pairwise frame matches to video tracks. Green lines link features positions in first and last frame.

Existing solutions. In the case of unordered feature tracking, the standard available algorithms, Bundler [2] and ETH-V3D [3] do not solve the problem in an optimal way, producing outputs that depend on the pairing order. Besides, they are complex and require at least *O(n log n)* operations (correspondence sorting).

The Union-Find approach. Our solution is simple and does not depend on the image pairing order (algo.1). A singleton is initially created for each matched feature. Then each pairwise match entails the union of the two sets containing them. Each resulting set corresponds to a track. The key ingredient is the join function built on the Union-Find algorithm to fuse correspondence subsets, defined as join(a,b) = union(find(a), find(b)). The complexity is $O(n\ \alpha(n))$ where α grows like the inverse of the Ackermann function — i.e., quasi-linear in practice.

Algorithm 1 Unordered feature tracking

```
Require: list of pairwise correspondences
Ensure: tracks
  create a singleton for each feature
  for each pairwise match do
    join({leftMatch}, {rightMatch})
  end for
  return each connected set as a track
```

Experimental results. We have performed experiments on large datasets: up to 2,600 images, with more than 41 million pairwise matches, yielding a million tracks. We compared our method with Bundler and ETH-V3D. It is up to 4 times faster, with a median speedup of 1.8. Moreover it is able to find more tracks than Bundler and ETH-V3D on all tested data, on average a 10% increase.

Conclusion. The simplicity of our method guarantees the reliability of the track computation; it actually produces more tracks than existing methods. Moreover, it has arguably the lowest possible complexity and is indeed faster than other methods.

Acknowledgements. We would like to thank Guillaume Chatelet and Renaud Marlet for their feedback.

References

[1] *B. A. Galler and M. J. Fischer. An improved equivalence algorithm. ACM, V7, I5, May 1964.*

[2] N. Snavely, S. M. Seitz and R. Szeliski. Photo tourism: exploring photo collections in 3D. SIGGRAPH 2006.

[3] C. Zach. ETH-V3D Structure-and-Motion software. © 2010-2011. ETH Zurich.