

experimental comparison of some of these algorithms on image retrieval datasets. You can also find more details on related techniques and systems in Section 6.2.3 on visual similarity search, which discusses global descriptors that represent an image with a single vector (Andjelovic, Gronat et al. 2016; Radenović, Tolias, and Chum 2019; Yang, Kien Nguyen et al. 2019; Cao, Araújo, and Sim 2020; Ng, Balntas et al. 2020; Tolias, Jenicock, and Chum 2020) as alternatives to bags of local features, Section 11.2.3 on location recognition, and Section 11.4.6 on large-scale 3D reconstruction from community (*internet*) photos.

7.1.5 Feature tracking

An alternative to independently finding features in all candidate images and then matching them is to find a set of likely feature locations in a first image and to then search for their corresponding locations in subsequent images. This *detect then track* approach is more widely used for video tracking applications, where the expected amount of motion and appearance deformation between adjacent frames is expected to be small.

The process of selecting good features to track is closely related to selecting good features for more general recognition applications. In practice, regions containing high gradients in both directions, i.e., which have high eigenvalues in the auto-correlation matrix (7.8), provide stable locations at which to find correspondences (Shi and Tomasi 1994).

In subsequent frames, searching for locations where the corresponding patch has low squared difference (7.1) often works well enough. However, if the images are undergoing brightness change, explicitly compensating for such variations (9.9) or using normalized cross-correlation (9.11) may be preferable. If the search range is large, it is also often more efficient to use a *hierarchical search strategy*, which uses matches in lower-resolution images to provide better initial guesses and hence speed up the search (Section 9.1.1). Alternatives to this strategy involve learning what the appearance of the patch being tracked should be and then searching for it in the vicinity of its predicted position (Avidan 2001; Jurie and Dhume 2002; Williams, Blake, and Cipolla 2003). These topics are all covered in more detail in Section 9.1.3.

If features are being tracked over longer image sequences, their appearance can undergo larger changes. You then have to decide whether to continue matching against the originally detected patch (feature) or to re-sample each subsequent frame at the matching location. The former strategy is prone to failure, as the original patch can undergo appearance changes such as foreshortening. The latter runs the risk of the feature drifting from its original location to some other location in the image (Shi and Tomasi 1994). (Mathematically, small misregistration errors compound to create a *Markov random walk*, which leads to larger drift over time.)



experimental comparison of some of these algorithms on image retrieval datasets. You can also find more details on related techniques and systems in Section 6.2.3 on visual similarity search, which discusses global descriptors that represent an image with a single vector (Arundjelovic, Gronat *et al.* 2016; Radenović, Tolias, and Chum 2019; Yang, Kien Nguyen *et al.* 2019; Cao, Arujo, and Sim 2020; Ng, Balntas *et al.* 2020; Tolias, Jenícek, and Chum 2020) as alternatives to bags of local features, Section 11.2.3 on location recognition, and Section 11.4.6 on large-scale 3D reconstruction from community (internet) photos.

7.1.5 Feature tracking

An alternative to independently finding features in all candidate images and then matching them is to find a set of likely feature locations in a first image and to then search for their corresponding locations in subsequent images. This kind of *detect then track* approach is more widely used for video tracking applications, where the expected amount of motion and appearance deformation between adjacent frames is expected to be small.

The process of selecting good features to track is closely related to selecting good features for more general recognition applications. In practice, regions containing high gradients in both directions, i.e., which have high eigenvalues in the auto-correlation matrix (7.8), provide stable locations at which to find correspondences (Shi and Tomasi 1994).

In subsequent frames, searching for locations where the corresponding patch has low squared difference (7.1) often works well enough. However, if the images are undergoing brightness change, explicitly compensating for such variations (9.9) or using normalized cross-correlation (9.11) may be preferable. If the search step is large, it is also often more efficient to use a *hierarchical search strategy*, which uses matches in lower-resolution images to provide better initial guesses and hence speed up the search (Section 9.1.1). Alternatives to this strategy involve learning what the appearance of the patch being tracked should be and then searching for it in the vicinity of its predicted position (Artiouk 2001; Jude and Lindeberg 2002; Williams, Blake, and Cipolla 2003). These topics are all covered in more detail in Section 9.1.3.

If features are being tracked over longer image sequences, their appearance can undergo larger changes. You then have to decide whether to continue matching against the originally detected patch (feature) or to re-sample each subsequent frame at the matching location. The former strategy is prone to failure, as the original patch can undergo appearance changes such as foreshortening. The latter runs the risk of the feature drifting from its original location to some other location in the frame (Shi and Tomasi 1994). (Mathematically, small mislocalization errors compound to create a *Markov random walk*, which leads to larger drift over time.)

recognition
photos
measure tracking
use to independent
of live

for more than one direction, i.e., which to search for more locations at which to search well even for both directions at which to search. It often works well for the search range to be explicitly compensating for such matches in low stable locations in subsequent frames, and hence speed up the search (Section 9.11) may be preferable. If the search range is predicted position (Avidan et al., 2000), and the appearance of the patch being searched is hierarchical, then topics are all considered to be in the same location.

cross-correlation search window. These strategies involve learning window shapes that are efficient to use as initial guess for the search window. One way to provide better initial guess is to use a multi-scale search window (Kroemer et al. 2008). Another way to provide better initial guess is to use a learned initial guess (Hartley and Stenger 1994; Williams, Blake, and Cipolla 2003). These strategies involve learning sequences, which are being tracked over longer image sequences, and then searching for it in the vicinity of its previous position. You then have to decide whether to continue tracking the feature as the original patch can move or to re-sample each subsequent frame as the original patch can change the location of the feature (Section 9.1.3).

preferable solution is to compare the original translational model and the registration model (Section 9.2). Shi and Tomasi's registration model is based on the assumption that the registration error is the sum of two components: a translational component and a rotational component. The translational component is modeled as a Gaussian distribution, while the rotational component is modeled as a uniform distribution. The registration error is then calculated as the sum of the squared differences between the observed and predicted feature locations. The registration model is then used to estimate the registration parameters, which are then used to register the two images.

A previous slide showed how to estimate affine motion in overlapping frames using a two-frame search. In this step we will extend this to initialize an affine frame.

Their appearance can undergo matching against the originally undergone appearance changes such as drifting from its original location to a walk, which leads to larger drift over a patch to later image locations using an algorithm (Tomasi, 1994) first compare patches in neighborhoods and then use the location estimates produced by between the patch in the current frame and the