# Approximate Nearest Neighbor Fields in Video

Nir Ben-Zrihem, Lihi Zelnik-Manor
Department of Electrical Engineering, Technion, Israel.

The Approximate Nearest Neighbor (ANN) problem could be defined as follows: given a set of reference points and incoming query points, quickly report the reference point closest to each query. Approximate solutions perform the task fast, but do not guarantee the exact nearest neighbor will be found. When the set of queries consists of all patches of an image the result is an ANN-Field (ANNF). Several excellent solutions have been developed for computing ANNF between a pair of images [1, 2, 4, 5, 6, 7]. It is thus not surprising that ANNF methods have become very popular and now lie in the heart of many computer vision applications, e.g., texture synthesis, image denoising, super-resolution, and image editing to name a few.

In this paper we present an efficient ANNF algorithm for video. The problem setup we address is matching all patches of a video to a set of reference patches, in real-time, as illustrated in Figure 1. In our formulation the reference set is fixed for the entire video. In fact, we could use the same reference set for different videos. Additionally, our reference set of patches is not restricted to originate from a single image or a video frame, as is commonly assumed for image ANNF. Instead, it consists of a non-ordered collection of patches, e.g., a dictionary [3]. This setup enables real-time computation of the ANN Fields for video. We show empirically, that it does not harm accuracy.

Our algorithm is designed to achieve low run-time by relying on two key ideas. First, we leverage the fact that the reference set is fixed, and hence moderate pre-processing is acceptable. At pre-processing we construct a data structure of the reference set, that enables efficient indexing during run-time. Second, we rely on temporal-coherency to adapt our hashing functions to the data. The hashing we use is not fixed a-priori as in previous works [1, 2, 4, 5, 6, 7], but rather it is tuned per query patch during runtime. In regions with high temporal change the hashing is tuned to be coarse, which leads to higher computation times (larger bins translate to checking out more candidate matches). On the contrary, in regions with low change, the hashing is finer and hence the computation time is lower. We refer to this approach as "query-sensitive hashing". Our hashing is generic to any distance metric.

Figure 2 provides a visualization of our hashing approach. We denote by $q_{x,y,t-1}$ and $q_{x,y,t}$ two consecutive query patches at position $x, y$ in frames $t-1$ and $t$, respectively. Let $r_i$ and $r_j$ be their NN matches in the reference set. Once $r_i$ is found, one could search for $r_j$ within a "fat" ring of radius $dist(r_i, q_{x,y,t}) \pm \varepsilon$, around $r_i$, as illustrated in Figure 2.(a). In areas with significant change $q_{x,y,t}$ will be far from $r_i$, the ring radius will be large and will include many reference patches. On the contrary, in areas with little change $q_{x,y,t}$ will be near $r_i$, the ring radius will be small and will include only a few candidate reference patches.

As can be seen in Figure 2.(a), the ring around $r_i$ includes the neighbors of $q_{x,y,t}$, but it also includes reference patches that are very far from $q_{x,y,t}$, e.g., on the other side of the ring. To exclude these patches from the set of candidates we further draw rings of radius $dist(r_k, q_{x,y,t}) \pm \varepsilon$, around points $r_k$ selected at random from the current set of candidates. The final candidate NNs are those that lie in the intersection of all rings, as illustrated in Figure 2.(b). Hence, the name of our approach Ring Intersection Approximate Nearest Neighbors (RIANN).

Our experiments show that RIANN computes ANNF for video in real-time, even for XVGA resolution. To confirm the usefulness of the proposed framework the paper describes how it can be adopted for real-time video processing. We show that a broad range of patch-based image transformations can be approximated using our nearest neighbor matching. Specifically we provide examples of realtime video denoising, colorization and several styling effects.

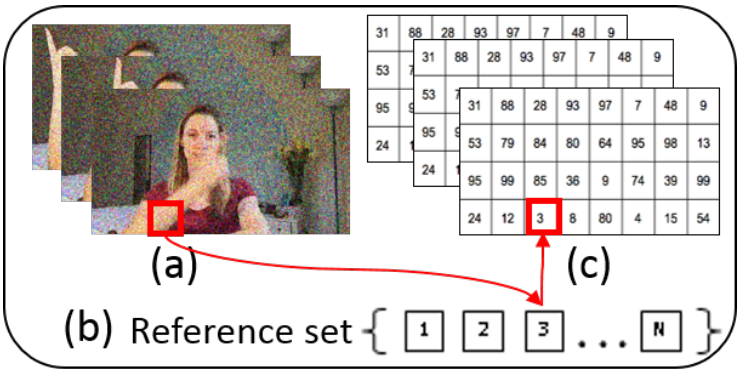


Figure 1: Video ANN Fields: (a) Input: Live stream of video frames. (b) A reference set of image patches. (c) For each frame, a dense ANN Field is produced by matching patches from the reference set. This is done in realtime.

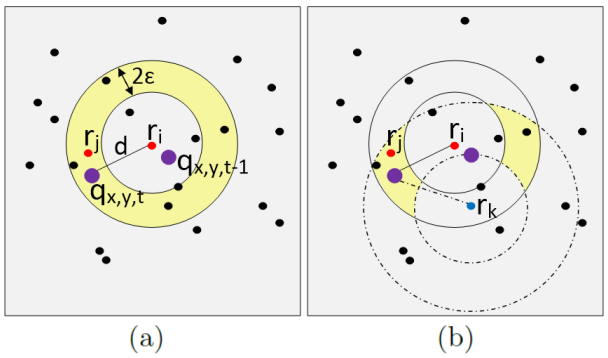


Figure 2: Ring Intersection Hashing: (a) To find candidate neighbors for $q_{x,y,t}$ we draw a ring of radius $d = dist(r_i, q_{x,y,t})$ and width $2\varepsilon$ around $r_i$. Here $r_i$ is the match found for $q_{x,y,t-1}$. (b) To exclude candidates that are far from $q_{x,y,t}$, we draw another ring, this time around $r_k$, one of the current candidates. We continue to add rings, and leave in the candidate set only those in the intersection.

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *Int. Conf. Multimedia*, 28(3), 2009.

[2] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. The generalized PatchMatch correspondence algorithm. In *European Conf. Comput. Vision*, 2010.

[3] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42, 2006.

[4] Kaiming He and Jian Sun. Computing nearest-neighbor fields via propagation-assisted kd-trees. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 111–118, 2012.

[5] Olonetsky Igor and Avidan Shai. Treecann - k-d tree coherence approximate nearest neighbor algorithm. In *European Conf. Comput. Vision*, volume 7575, pages 602–615, 2012.

[6] Simon Korman and Shai Avidan. Coherency sensitive hashing. In *Proc. Int. Conf. Comput. Vision*, pages 1607–1614, 2011.

[7] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340, 2009.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.