On the convergence of *PatchMatch* and its variants

Thibaud Ehret and Pablo Arias

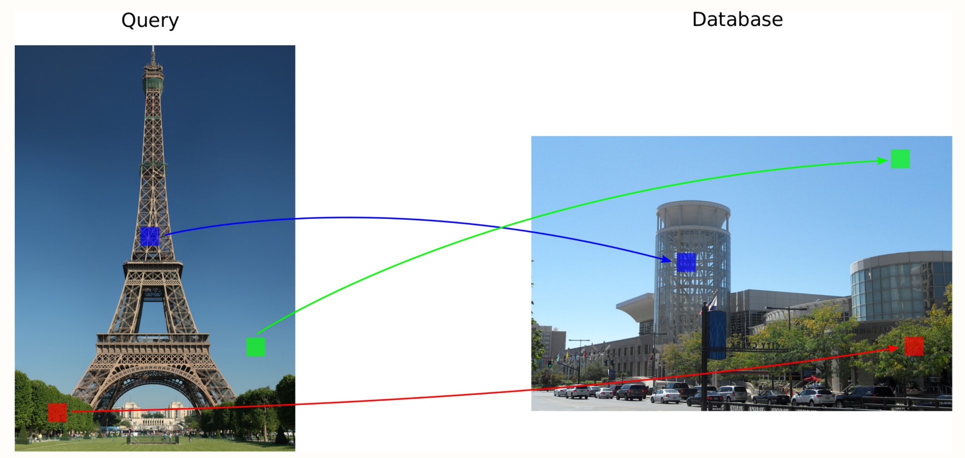
CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France

Contributions

Helps to understand the mechanisms of *PatchMatch* [1] and its variants. To that aim we:

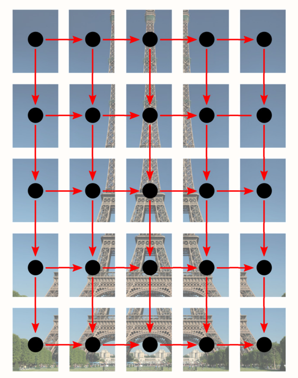
- Formalize a generic *PatchMatch* as a collaborative optimization of a family of energies related by a propagation graph;
- Derive convergence bounds for this generic *PatchMatch*;
- Derive specific convergence bounds for two versions of *PatchMatch*: the original *PatchMatch* [1] and *CSH* [2].

PatchMatch review

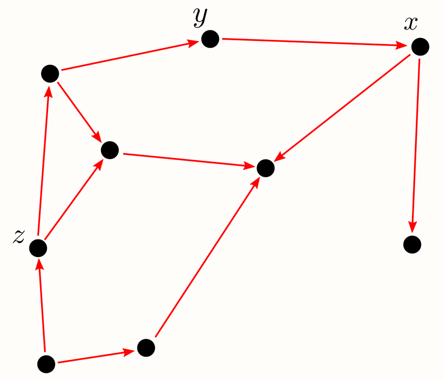


PatchMatch is a fast algorithm to find the k best matching patches in the database image for each patch in a query image. It computes an approximate solution iteratively via a collaborative random search. In each iteration the query patches sample database patches and propagate their best current candidates to their neighbors in the query image.

We provide a generic framework for PatchMatch algorithms and study their convergence rate towards exact matches.



Propagation graph of the original *PatchMatch*



Any DAG can be used as a propagation graph

Generic *PatchMatch* formulation

Problem statement ($k = 1$): We have a family of energies $U_x : \mathbf{B} \to \mathbb{R}$, for $x \in \mathcal{V}$. We want to find for each x, $\varphi_x \in \mathbf{B}$ such that:

$$U_x(\varphi_x) = \min_{\xi \in \mathbf{B}} U(\xi)$$

To define a *PatchMatch* algorithm we need:

A propagation graph: A DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We denote $(y, x) \in \mathcal{E}$ as $y \sim x$.

Propagation actions: To each $y \sim x$ we associate a transformation $A_{y,x} : \mathbf{B} \to \mathbf{B}$.

Transition kernels: Q_i such that for $\xi \in \mathbf{B}$, $Q_i(\xi, \cdot)$ is a probability distribution over $\mathbf{B}$. Random samples are denoted by $S_i\xi \sim Q_i(\xi, \cdot)$.

Best operator: Selects the best matching patch from a set of candidates based on the matching energy U_x.

Algorithm 1: Generic patch matching algorithm

Initialize propagation graph $\mathcal{G}$
Initialize matching φ^0

for $n \in \mathbb{N}$ **do**
`# update candidates`
for $x \in \mathcal{V}$ *following the topological ordering* **do**

$$\varphi_x^{n+1/2} = \text{best}_x\left(\varphi_x^n \cup \bigcup_{y\sim x} A_{y,x}\varphi_y^{n+1} \cup \bigcup_{y\sim x} S_2 A_{y,x}\varphi_y^{n+1}\right)$$

$$\varphi_x^{n+1} = \text{best}_x\left(\varphi_x^{n+1/2} \cup S_1\varphi_x^{n+1/2}\right)$$

end

`# reverse propagation graph`
`# update propagation graph`
end

Intuition of the proof

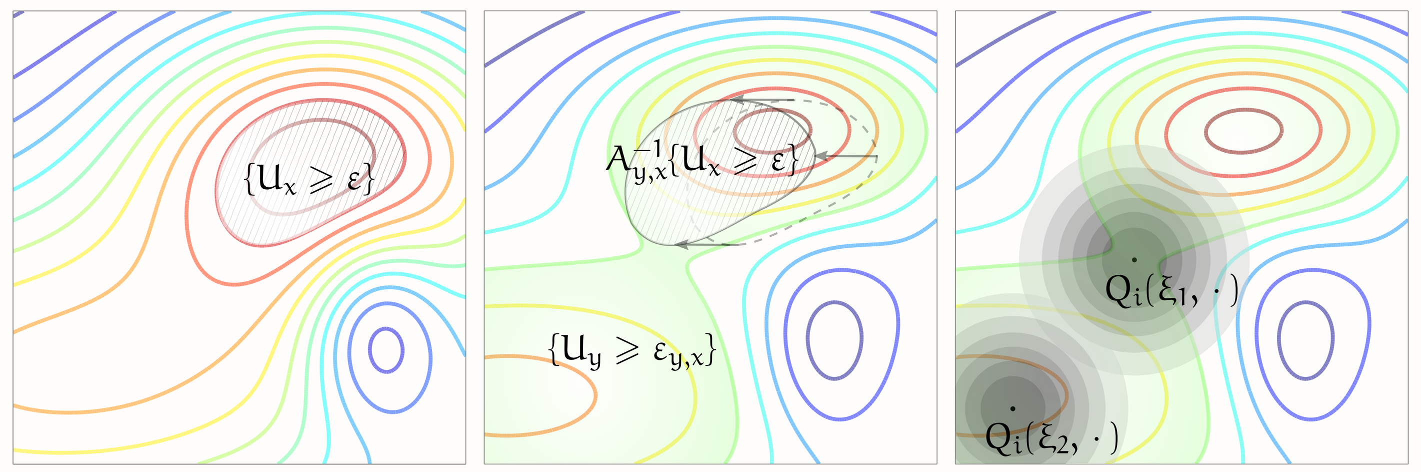
Step 1: Constraints propagation. The assumption that $U_x(\varphi_x^{n+1}) \geqslant \varepsilon$ imposes that $U_z(\varphi_z^{n+1}) \geqslant \varepsilon_{z,x}$ for x's ancestors z. The levels $\varepsilon_{z,x} \geqslant 0$ are defined by a backwards recursion on the propagation graph:

$$\varepsilon_{z,x} = \min\left\{ U_z(\theta) \,\middle|\, \theta \in \bigcap_{y \text{ s.t. } z\sim y} A_{z,y}^{-1}(\{U_y \geqslant \varepsilon_{y,x}\}) \right\}$$

Step 2: Worst-case transition. Highest probability of keeping an energy higher than ε after a random search.

$$C_i(z, \varepsilon) := \sup_{\xi \in \{U_z \geqslant \varepsilon\}} Q_i(\xi, \{U_z \geqslant \varepsilon\}).$$

$C(z, .)$ is a non-increasing function such that for $\varepsilon < 0$, $C(z, \varepsilon) \in [0, 1[$ and $C(z, \varepsilon) = 1$ for $\varepsilon \leqslant 0$.



(a) U_x energy landscape (b) U_y energy landscape (c) transition kernels

Theoretical results

Theorem (point-wise convergence) For $\varepsilon > 0$ and $x \in \mathcal{V}$, we have after an iteration of the generic PatchMatch:

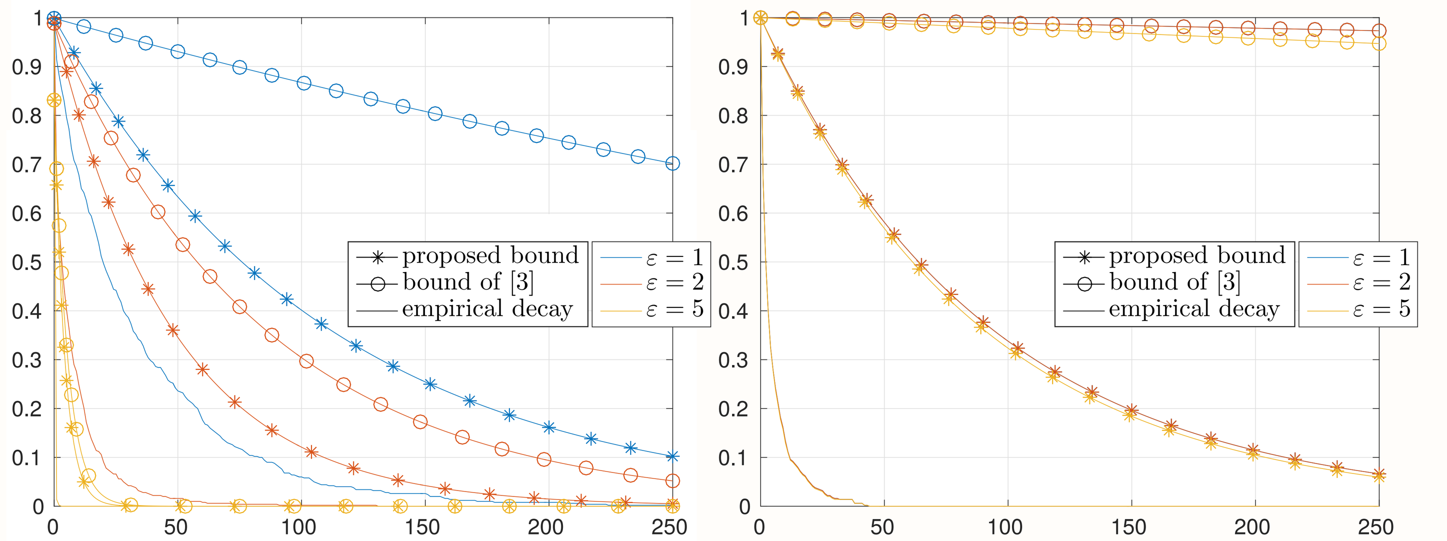
$$\mathbb{P}(U_x(\varphi_x^{n+1}) \geqslant \varepsilon) \leqslant \mathbb{P}(U_x(\varphi_x^n) \geqslant \varepsilon) \prod_{z \in A} C_2(z, \varepsilon_{z,x})^{\mu(z)} C_1(z, \varepsilon_{z,x}),$$

where $\mu(z)$ is the number of parents of z.

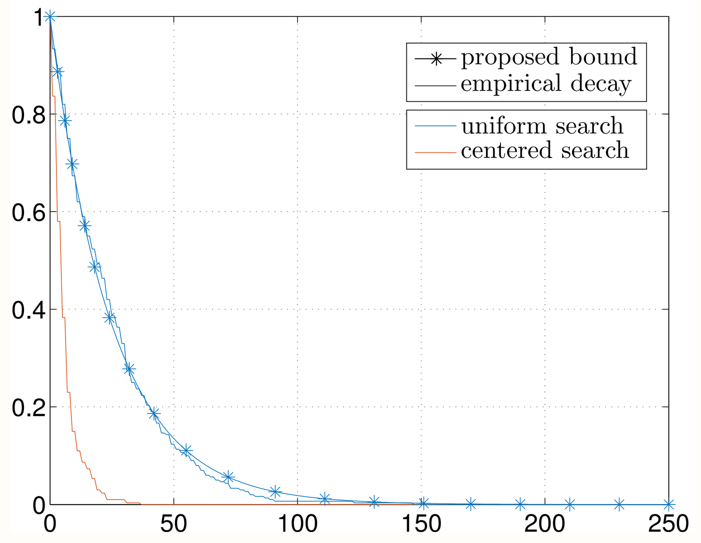
Additional results in our paper:

- The case of k best matches;
- Uniform convergence and convergence in the mean;
- Specific bounds for the original *PatchMatch* [1] (improving over [3]) and for *CSH* [2].

Experimental validation



Comparison with the empirical bound: We estimate $\mathbb{P}(U_x(\varphi_x^n) \geqslant \epsilon)$ at each iteration n for two query patches x and compare it with our bound and the one of [3].



Uniform random search: For $k = 1$, the gap between the bound and the empirical decay is mainly due to the worst-case transition C. To verify this we use the **uniform** random search. We match two images of random noise with the query image copied inside the database image. The plots are for $\varepsilon = 0.5$, and only the unique match is below ε. With the uniform search, the empirical decay matches the bound, which coincides with $(1 - 1/p)^{qn}$, where p and q are the numbers of database and query patches.

References

[1] Barnes et al. "PatchMatch: a randomized correspondence algorithm for structural image editing". *ACM Transactions on Graphics-TOG*, 2009.

[2] Korman and Avidan "Coherency sensitive hashing". *IEEE ICCV*, 2011.

[3] Arias et al. "Analysis of a variational framework for exemplar-based image inpainting". *SIAM*, 2012.

[4] Kaiming and Sun "Computing Nearest-Neighbor Fields via Propagation-Assisted KD-trees". *IEEE CVPR*, 2012.