An implementation of Maximally Stable Extremal Regions

Andrea Vedaldi

January 18, 2007

These notes describe the implementation[1] of Maximally Stable Extremal Region of [1]. Sect. 1 formalizes the concept of "maximally stable extremal regions" and Sect. 2 describes the algorithm that we use to compute them.

1 Maximally stable extremal regions

Here an image $I(x), x \in \Lambda$ is a real function of a discrete domain $\Lambda \subset \mathbb{R}^2$ with a neighborhood topology τ. Elements of Λ are called *pixels*. Here we consider regular domains (i.e. finite subsets of $\mathbb{Z}^2$) and the elementary 4-way or 8-way neighborhood topology. A *level set* $S(x)$, $x \in \Lambda$ is the set of pixels that have intensity not greater than $I(x)$, i.e.

$$S(x) = \{y \in \Lambda : I(y) \leq I(x)\}.$$

A *path* $(x_1, \ldots, x_n)$ is a sequence of pixels such that x_i and x_{i+1} are neighbors for all $i = 1, \ldots, n-1$. A *connected component* C of the domain Λ is a subset $C \subset \Lambda$ for which each pair $(x_1, x_2) \in C^2$ of pixels is connected by a path fully contained in C. The connected component is *maximal* if any other connected component C' containing C is equal to C. An *extremal region* R is a maximal connected component of a level set $S(x)$.

Among all extremal regions $\mathcal{R}(I)$ of image I, we are interested to the onesi that satisfy certain stability criterion which we introduce next. Let the *level* $I(R)$ of the extremal region R be the maximum image intensity $\max_{x \in R} I(x)$ attained in the region R. Consider the biggest extremal region $R_{+\Delta}$ contained in R that has intensity greater at least $\Delta \geq 0$ than R, i.e.

$$R_{+\Delta} = \operatorname{argmax}\{|Q| : Q \in \mathcal{R}(I), \ Q \subset R, \ I(Q) \geq I(R) + \Delta\}. \qquad (1)$$

Similarly, consider the smallest extremal region containing R that has intensity at least Δ smaller than R, i.e.

$$R_{-\Delta} = \operatorname{argmin}\{|Q| : Q \in \mathcal{R}(I), \ Q \supset R, \ I(Q) \leq I(R) - \Delta\}. \qquad (2)$$

---

[1]The implementation can be downloaded at `http://vision.ucla.edu/~vedaldi/code/mser/`

Consider the area variation

$$\rho(R; \Delta) = \frac{|R_{+\Delta}| - |R_{-\Delta}|}{|R|}.$$

The region R is *maximally stable* if it is a minimum for the area variation, i.e. if $\rho(R; \Delta)$ is smaller than $\rho(Q; \Delta)$ for any region Q "immediately contained" or "immediately containing" R. We say that an extremal region R *immediately contains* another extremal region Q if $R \supset Q$ and if R' is another extremal region with $R \supset R' \supset Q$, then $R' = R$.

2 Regions computation

We describe an efficient algorithm for the computation of the maximally stable extremal regions of an image $I(x)$ defined on a discrete domain Λ.

2.1 Enumerating extremal regions

We describe first a method to enumerate all extremal regions of a given image I. Let $x_1, x_2, \ldots, x_N \in \Lambda$ be a sorting of the image pixels by increasing inteisty value, i.e.

$$I(x_1) \le I(x_2) \le \ldots I(x_N).$$

We compute extremal regions incrementally, by considering larger and larger image subdomains $\Lambda_t = \{x_1, x_2, \ldots, x_t\} \subset \Lambda$ for $t = 1, \ldots, N$. Denote by $I_t = I|_{\Lambda_t}$ the restriction of the image I to the subset Λ_t.

For $t = 1$, $\Lambda_1 = \{x_1\}$ is trivially an extremal region of the image $I_!$ and level $I(x_1)$. For $t = 2$, either x_1 and x_2 are connected and Λ_2 is an extremal region of I_2, or they are not and $\{x_2\}$ is an extremal region of Λ_2. Moreover Λ_1 is an extremal region of I_2 if, and only if, $I(x_2) \ne I(x_1)$. This is captured in general by:

Lemma 1. *Let t be one of $1, 2, \ldots, N - 1$. Let $R_1, \ldots, R_K$ be all the extremal regions of I_t. Let*

- $\mathcal{K}_1$ *the subset of indices k for which $I(R_k) \ne I(x_{t+1})$ and*

- $\mathcal{K}_2$ *the subset of indices k for which $I(R_k) = I(x_{t+1})$ but x_{t+1} is not connected to R_k and*

- *let $\mathcal{K}_3$ be the subset of indices k for which x_{t+1} is connected to R_k.*

Then

1. *for all $k \in \mathcal{K}_1 \cup \mathcal{K}_2$ the set R_k is an extremal region of I_{t+1};*

2. *the set $R = \{x_{t+1}\} \cup_{k \in \mathcal{K}_3}$ is an extremal region of I_{t+1};*

3. *all extremal regions of I_{t+1} are obtained either as (1) or (2).*

Proof. By definition each R_k is a maximal connected component of the set $S_t(R_k) = \{x \in \Lambda_t : I(x) \leq I(R_k)\}$. If $k \in \mathcal{K}_1$, then $I(R_k) \neq I(x_{t+1})$, $S_t(R_k) = S_{t+1}(R_k)$ and R_k is a maximal connected component of $S_{t+1}(R_k)$ as well. If $k \notin \mathcal{K}_1$, then $S_{t+1}(R_k) = S(R_k) \cup \{x_{t+1}\}$. However if $k \in \mathcal{K}_2$, then R_k and x_{t+1} are not neighbors and R_k is still maximal in $S_{t+1}(R_k)$. Finally, $\{x_{t+1}\}$ together with all the regions R_k of level $I(R_k) \leq I(x_{t+1})$ which are neighbors of x_{t+1}, i.e. $k \in \mathcal{K}_3$, constitute a new extremal region. To see this, note that (i) $R \subset S(x_{t+1})$, (ii) R is connected because the subregions R_k are connected and any two points in two different subregions are connected through x_{t+1} by construction and (iii) R is maximal as if not, one could add a pixel $y \in \Lambda_t = \Lambda_{t+1} - \{x_{t+1}\}$ to R that would be either an extension of one of the extremal regions R_k of image I_t or $\{y\}$ would be a new extremal region of image I_t by itself.

Finally, we need to show that the listing is exhaustive. So let R be an extremal region of image I_{t+1}. If $R \subset S_t(R)$, then $x_{t+1} \notin R$ and R is equal to some R_k for $k \in \mathcal{K}_1 \cup \mathcal{K}_2$ by the inductive hypotesis. If, on the other hand, $x_{t+1} \in R$, then R is obtained as (2). □

Lemma 1 suggests a simple algorithm to enumerate extremal regions. The idea is to consider one pixel at time in the order $x_1, x_2, \dots$ growing extremal regions for the intermediate images I_t until $I_N = I$ is reached.

Formally, this process can be implemented by means of a forest of pixels. At time t the forest represents all the union operations that have been performed so far according to point (2) of Lemma 1. Since extremal regions are only generated by such union operations, the tree stores all the extremal regions of all intermediate images $I_1, \dots, I_t$.

Let us consider the addition of pixel x_{t+1} to the forest. Following Lemma 1, we must search for all extremal regions $R_1, \dots, R_k$ of image I_t which are neighbors of x_{t+1} and join them to x_{t+1} to obtain the new region R. This is done by scanning the neighbors $y \in \Lambda_t$ of x_{t+1} and, for each of them, climbing the tree in search for the appropriate extremal regions R_k. In practice, we simply take the union of all sets $S(y) \cup S(\pi(y)) \cup S(\pi^2(y)) \cup \cdots \cup S(\text{root}(y)) = S(\text{root}(y))$, where $S(y)$ is the subtree rooted at y, $\pi(y)$ is the parent of y and $\text{root}(y)$ is the root of the tree that contains y. While only some of $S(\pi^n(y))$ are indeed extremal regions of image I_t, $S(\text{root}(y))$ always is and, since it covers all other subsets anyway, it is sufficient to join that. The join operation is then encoded in the forset by making x_{t+1} parent of $\text{root}(y)$, i.e. $\pi(\text{root}(y)) \leftarrow x_{t+1}$.

This basic algorithm can be improved significantly by keeping the tree balanced. This is an optimization of the join operation, for which x_{t+1} is not necessarily added to the forest as root; instead one uses as root one of the nodes $\text{root}(y)$ with the goal keeping the tree height short. Although this disrupts partially the property of the forest (some of the extremal regions of the intermediate images $I_1, I_2, \dots$ are lost), the relevant information (i.e. the regions that are extremal regions of I) is preserved, as it can be verified. In particular, regions can be emitted as soon as condition (1) of the Lemma is encountered, which correspond to the case $I(y) \neq I(\pi(y))$.

2.2 Computing the stability score

Once the extremal region tree is computed, we need to calculate the area variation for each region and then selecting the maximally stable one. The area $|R|$ of each region is computed efficiently as explained in Sect. 2.4. In order to compute the area variation of a region R, we need to figure out the regions $R_{-\Delta}$ and $R_{+\Delta}$. To do this we begin by arranging the extremal regions of image I into a tree where R is parent of R' if R immediately contains R'. Then each region R is considered and the tree is explored to find a region Q for which $R = Q_{-\Delta}$ and the region $R_{+\Delta}$. This is done by scanning the regions $R_0 = R$, $R_1 = \pi(R_0)$, $R_2 = \pi(R_1)$ and so on. If a region $Q = R_i$ satisfies $Q_{-\Delta} = R_0$, then

$$I(R_0) \leq I(R_i) - \Delta < I(R_1).$$

The condition is not necessary though; according to (2) we need to keep the region of maximum area among all the candidate ones. Similarly, if $R_i = R_{+\Delta}$, then

$$I(R_i) \leq I(R_0) + \Delta < I(R_{i+1}).$$

In this case the condition is also sufficient as at most one of such regions exist.

2.3 Cleaning up

The stability score alone may not be sufficient to select only useful regions. In the cleanup phase we

- remove very small and very big regions;

- remove regions which have too high area variation (even if they are indeed minima of the variation score);

- remove duplicated regions.

Duplicated regions arise because, due to noise, the same mode of the local minima score may correspond to more than one local minimum. Duplicated regions are easily found by comparing each MSER R with the MSER R' immediately containing R and removing R if they are too similar.

2.4 Fitting elliptical regions

Fitting elliptical regions amount to computing for each maximally stable extremal region R the first and second order moments, i.e.

$$\mu(R) = \frac{1}{|R|} \sum_{x \in R} x, \qquad \Sigma(R) = \frac{1}{|R|} \sum_{x \in R} (x - \mu)(x - \mu)^\top.$$

Rather than considering directly the centered moment $\Sigma(R)$, it is computationally more convenient to compute

$$M(R) = \frac{1}{R} \sum_{x \in R} x x^\top$$

4

and use the fact that $\Sigma(R) = M(R) - \mu(R)\mu(R)^\top$. The advantage is that any quantity which is obtained by integrating a function $f(x)$, $x \in \Lambda$ of the image domain (in particular $f(x) = x$ and $f(x) = xx^\top$) can be computed for all regions at once by visiting (in breath first order and from the leaves) each pixel of the forest and summing its value to the parent.

References

[1] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002.