Spline-Based Dense Medial Descriptors for Image Simplification Using Saliency Maps

Jieying Wang[4(✉)], Leonardo de Melo[2], Alexandre X. Falcão[2], Jiří Kosinka[1], and Alexandru Telea[3]

[1] Bernoulli Institute, University of Groningen, 9747 AG Groningen, The Netherlands
j.kosinka@rug.nl
[2] Department of Information Systems, Institute of Computing, University of Campinas, São Paulo 13083-852, Brazil
afalcao@ic.unicamp.br
[3] Department of Information and Computing Sciences, Utrecht University, 3584 CC Utrecht, The Netherlands
a.c.telea@uu.nl
[4] Bernoulli Institute, Shandong University of Science and Technology, 579 Qianwangang Road, Huangdao District, Qingdao 266590, Shandong Province, P.R. China
jieying.wang@rug.nl

Abstract. Medial descriptors have attracted increasing interest in image representation, simplification, and compression. Recently, such descriptors have been separately used to (a) increase the local quality of representing salient features in an image and (b) globally compress an entire image via a B-spline encoding. To date, the two desiderates, (a) high local quality and (b) high overall compression of images, have not been addressed by a single medial method. We achieve this integration by presenting Spatial Saliency Spline Dense Medial Descriptors (3S-DMD) for saliency-aware image simplification-and-compression. Our method significantly improves the trade-off between compression and image quality of earlier medial-based methods while keeping perceptually salient features. We also demonstrate the added-value of user-designed, as compared to automatically-computed, saliency maps. We show that our method achieves both higher compression and better quality than JPEG for a broad range of images and, for specific image types, yields higher compression and similar quality than JPEG 2000.

Keywords: Medial descriptors · Saliency maps · B-splines · Image simplification · Image compression

1 Introduction

Image simplification and compression are essential in many applications in science, engineering, and consumer contexts. *Compression* methods, such as the well-known JPEG [44] and the newer JPEG 2000 [39] and BPG [5] efficiently reduce the cost of storing and/or transmitting an image, typically in a lossy manner, by discarding certain image features or details. *Simplification* keeps image structures deemed important while eliminating less-important ones, to ease the analysis and processing of the former structures.

A. A. de Sousa et al. (Eds.): VISIGRAPP 2021, CCIS 1691, pp. 279–302, 2023.
https://doi.org/10.1007/978-3-031-25477-2_13

Fig. 1. Examples of the DSR saliency estimator failing to detect salient objects: (a3–d3) show the SSDMD results using the saliency maps (a2–d2) for images (a1–d1).

A particular class of simplification-and-compression methods models images as a set of luminance threshold-sets [58] and encodes these by their Medial Axis Transforms (MATs). Wang *et al.* [50] followed this approach to propose Dense Medial Descriptors (DMD), a lossy image compression method. While DMD showed promising quantitative and qualitative results, it cannot yet compare in *both* visual quality and compression ratio (CR) with state-of-the-art compression methods like JPEG or similar. Two lines of research tried to address this issue.

Improving Quality: DMD simplifies an image *globally*, making it hard to preserve fine details in some areas while strongly simplifying the image in other areas. The SSDMD method [49] addressed this by adding a saliency map to DMD, allowing users to specify different spatial simplification levels over an image. SSDMD delivers higher local quality than DMD (as specified by the saliency map) but has two key limitations. First, it only marginally improves CR when compared to DMD, since highly-salient image areas actually increase the MAT information needed to be stored. Secondly, SSDMD uses *automatically* computed saliency maps to control simplification. Such maps can significantly fail to capture what users perceive as salient (thus, to be preserved) *vs* non-salient (thus, to be simplified). Figure 1 outlines this problem for four images (a1–d1) with saliency maps (a2–d2; bright=salient; dark=non-salient) automatically computed by the DSR method [25]. SSDMD compression results (a3–d3) arguably lose details that humans would find salient, such as blurred faces (a3, b3, c3) and nearly complete loss of the leopard skin texture (d3).

Improving Compression: DMD stores the MATs of an image's threshold sets as pixel chains, which is exact, but inefficient storage-wise. The Spline Dense Medial Descriptors (SDMD) method [47] improved CR by representing MATs with accurate and compact-storage B-spline descriptors for each threshold set [48]. Yet, just as DMD,

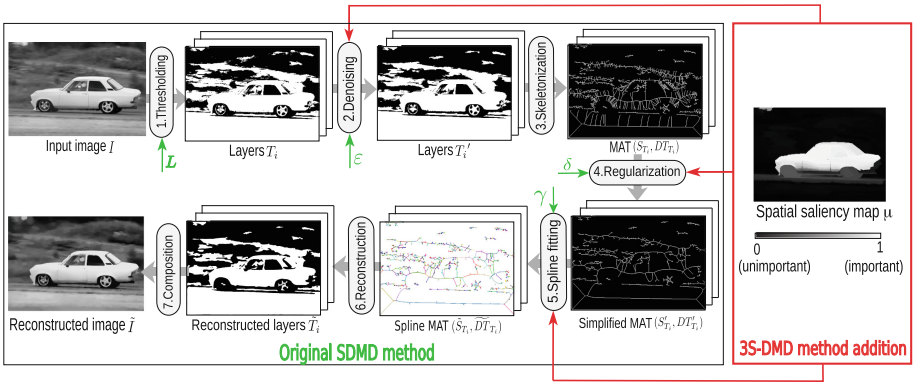

Fig. 2. Spline-based dense medial descriptors pipeline with free parameters in green. Elements added by 3S-DMD method proposed in this paper are marked in red. (Color figure online)

SDMD simplifies images only *globally*, thus increasing CR but achieving limited visual quality.

Our Contributions: We jointly address the visual quality *and* CR goals of all above earlier MAT-based image compression methods by a single method:

- We combine the strengths of SSDMD [49] (*spatial* control of image simplification) with SDMD [47] (*compact* encoding of MATs with B-splines);
- We allow users to interactively tune parameters of the joint method, including full control over the saliency map design;
- We evaluate our proposal on additional image types and compare it, with favorable results, with state-of-the-art methods (JPEG and JPEG 2000).

We organize this paper as follows. Section 2 presents related work on medial descriptors for image compression. Section 3 details our Spatial Saliency Spline-based Dense Medial Descriptor (3S-DMD) method. Section 4 evaluates our results. Section 5 discusses 3S-DMD. Section 6 concludes the paper.

2 Related Work

Our 3S-DMD method (Fig. 2) adapts SDMD to use saliency maps to further simplify less important regions while preserving salient ones (new contributions marked red in the figure). We next discuss related work: dense medial descriptors (Sect. 2.1), spline-based DMD (Sect. 2.2), and saliency maps (Sect. 2.3).

2.1 Dense Medial Descriptors (DMD)

The key idea of DMD [50] is to use medial axes to efficiently encode luminance threshold-sets of an image. Let $I : \mathbb{R}^2 \rightarrow [0, 255]$ be the 8-bit Y channel in the YUV space of a color image. DMD splits I in $n = 256$ threshold sets or *layers*

$$T_i = \left\{ \mathbf{x} \in \mathbb{R}^2 \,|\, I(\mathbf{x}) \geq i \right\}, 0 \leq i < n. \tag{1}$$

Since adjacent layers often contain highly similar information, DMD uses only a subset $D \subset \{T_i\}$ of $L = |D| < 256$ layers to encode I (Fig. 2, step 1). Here and next, $|\cdot|$ denotes set size. In some layers, small-size islands (connected components in the foreground T_i or background $\overline{T}_i$) can appear, due to small local intensity variations. DMD fills in, respectively removes, islands smaller than a fraction ε of $|T_i|$, respectively $|\overline{T}_i|$, which contribute little to the image I (Fig. 2, step 2). Next, DMD extracts medial axis transforms (S_{T_i}, DT_{T_i}) from these L layers (Fig. 2, step 3), where

$$DT_{T_i}(\mathbf{x}) = \min_{\mathbf{y} \in \partial T_i} \|\mathbf{x} - \mathbf{y}\| \tag{2}$$

is the distance transform [15] of the boundary ∂T_i of layer T_i, and

$$S_{T_i} = \{\mathbf{x} \in T_i | \exists \mathbf{f}_1 \in \partial T_i, \mathbf{f}_2 \in \partial T_i, \mathbf{f}_1 \neq \mathbf{f}_2 : \|\mathbf{x} - \mathbf{f}_1\| = \|\mathbf{x} - \mathbf{f}_2\| = DT_{T_i}(\mathbf{x})\} \tag{3}$$

is the medial axis, or skeleton, of T_i. In Eq. 3, $\mathbf{f}_1$ and $\mathbf{f}_2$ are called feature points [19] of skeletal point $\mathbf{x}$. Computing MATs of binary images is a well-studied topic, described in detail in classical work [21,23,31,35,36,38].

The medial axes S_{T_i} contain many so-called *spurious branches* caused by small perturbations along ∂T_i. Storing such branches takes significant space but contributes little to the reconstruction quality. To address this, DMD uses the salient-skeleton metric [40] defined as

$$\sigma(\mathbf{x}) = \frac{\rho(\mathbf{x})}{DT_{T_i}(\mathbf{x})}, \tag{4}$$

where $\rho(\mathbf{x})$ is the fraction of the boundary ∂T_i that the skeletal point $\mathbf{x}$ encodes [42]. Saliency-based regularization removes all pixels $\mathbf{x} \in S_{T_i}$ where $\sigma(\mathbf{x})$ is below a user-specified threshold $\delta > 0$, yielding a simplified skeleton S'_{T_i} and corresponding distance transform DT'_{T_i} (Fig. 2, step 4). From the regularized MAT (S'_{T_i}, DT'_{T_i}), one can reconstruct a simplified version $\widetilde{T}_i$ of each layer T_i as the union $\cup_{\mathbf{x} \in S'_{T_i}} B(\mathbf{x}, DT'_{T_i}(\mathbf{x}))$ of discs B centered at pixels $\mathbf{x} \in S'_{T_i}$ and with radii given by $DT'_{T_i}(\mathbf{x})$. An approximation $\widetilde{I}$ of the input image I is finally obtained by drawing all reconstructed layers $\widetilde{T}_i$ atop each other in increasing order of luminance i (Fig. 2, step 7).

DMD uses the fast GPU implementation of [7], which is pixel-exact and linear in the number of pixels in T_i [19,28]. Full implementation details are available at [50]. DMD provides an *accurate* encoding of grayscale and color images. However, DMD stores the MATs (S'_{T_i}, DT'_{T_i}) using pixel chain delta encoding, which is inefficient and leads to a *poor compression ratio*.

2.2 Spline-Based Medial Descriptors (SDMD)

Compactly encoding MATs has attracted interest in binary shape representation [22, 45]. B-splines were found effective for this as they store fewer (control) points than all pixels in an MAT. Zhu et al. [57] and Yushkevich *et al.* [55] accurately modeled MATs of 2D binary shapes with multiple cubic B-splines. Yet, they require vector representations of the input shape and its MAT and also require a Voronoi-based MAT method [4] which is slow for complex shapes.

Spline-based Medial Axis Transform (SMAT) [48] extended the above B-spline idea to use *raster* representations for T_i, S_{T_i}, and DT_{T_i}, to directly handle any binary raster image. In detail, MAT branches (S'_{T_i}, DT'_{T_i}), seen as 3D pixel curves, and fitted with 3D B-splines. Each control point $\mathbf{c}_j = (\mathbf{p}_j, DT_{T_i}(\mathbf{p}_j)) \in \mathbb{R}^3$ consists of a 2D position $\mathbf{p}_j$ and its corresponding DT value. Fitting uses the least-squares method [13] aiming to get (1) a minimal number of control points and (2) an approximation error γ between the MATs and B-splines below a user-given value γ (Fig. 2, step 5). Reconstruction first rasterizes the B-splines using de Casteljau's algorithm [34] and next creates the layers $\widetilde{T}_i$ by the disc-union method outlined in Sect. 2.1 (Fig. 2, step 6).

Wang *et al.* [47] proposed Spline Dense Medial Descriptors (SDMD) that uses the SMAT method to encode color images. SDMD applies SMAT to all luminance threshold sets of an image but also proposes three improvements to increase CR and quality: adaptively encoding upper or lower threshold-sets, treating chrominance and luminance separately, and removing Y-structures from the skeletons. For details, we refer to [47]. SDMD achieves much higher compression ratios at similar or even better quality to JPEG.

Compared with DMD, which proved to faithfully represent an image, SDMD encodes images both faithfully and *compactly*. Yet, both DMD and SDMD work *globally*: High simplification easily removes small, but visually important, details, leading to poor quality. Conversely, low simplification allocates storage to unimportant image areas, leading to poor compression.

2.3 Saliency Maps

Saliency maps encode the relative importance of various parts of an image for a given task or perceptual standpoint. A saliency map $\mu : \mathbb{R}^2 \rightarrow [0, 1]$ gives, for each image pixel $\mathbf{x}$, its importance or saliency, between totally irrelevant ($\mu = 0$) and maximal importance ($\mu = 1$). Such maps have been used for image quality assessment [27], content-based image retrieval [8], context-aware image resizing [18], and saliency-based image compression [3,59]. Saliency maps can be created either in supervised mode—by users via manual annotation (see next Sect. 3.1)—or in unsupervised mode, automatically computed from images.

Supervised methods use ground-truth images to learn discriminant features of salient objects [29]. The most accurate supervised methods use deep-learning [6,51] and typically outperform unsupervised methods. Ywt, they need large amounts of human-annotated training data, and the generalization of training models across image domains usually requires adaptation and retraining [29].

Unsupervised methods use prior knowledge about salient objects and local image characteristics. Most methods start by finding image regions (*e.g.* superpixels) with high color contrast relative to neighbors [20,25,56]. Besides contrast, objects in focus [20], near the image center [10], or having red and yellow tones, important for human vision [33], are all considered as salient factors. Conversely, regions similar to the boundary will have low saliency as most image boundaries are background in natural images [10,20,25,56]. In our work, we use the DSR [25] unsupervised bottom-up saliency estimation method which provides reliable saliency maps without requiring parameter tuning and is fast. Any other saliency estimators can be directly used instead as long as users find the produced maps suitable for their tasks at hand.

3 Proposed 3S-DMD Method

As stated above, an important limitation of SDMD is that it simplifies an image *globally*. Our earlier work, Spatial Saliency DMD (SSDMD) [49], addressed this by simplifying the DMD MAT's using a spatial saliency map. We next present both SSDMD and our new method, 3S-DMD, which improves SSDMD in several respects. Section 3.1 shows how 3S-DMD benefits from manually-designed saliency maps via an interactive application. Section 3.2 presents SSDMD's saliency-map-based simplification of the MAT and how we improved this by saliency-based spline fitting. Finally, Sect. 3.3 shows how we measure the quality of the results of our new 3S-DMD method.

3.1 User-Driven Saliency Map Generation

Section 2.3 reviewed a variety of techniques to automatically compute saliency maps from an image. As mentioned in Sect. 1, such automatically-computed maps may not fully meet user needs (*cf.* Fig. 1 (a2, b2)) or even fail to detect salient objects (*cf.* Fig. 1 (c2, d2)). Even when such maps fit with what users expect, the simplification they induce can lead to unwanted results due to the hard-to-predict shapes that skeletons have. To handle all such issues, we developed an interactive application that allows users to create their *custom* saliency maps or *adjust* maps created by automatic methods. Figure 3a shows the user interface, in which one can draw the saliency map using tools listed in the toolbar, tune all the method's parameters, run the end-to-end pipeline, and check the obtained results. A video of our tool is provided in the supplementary material [46].

We provide three ways for users to manually design saliency maps, as follows.

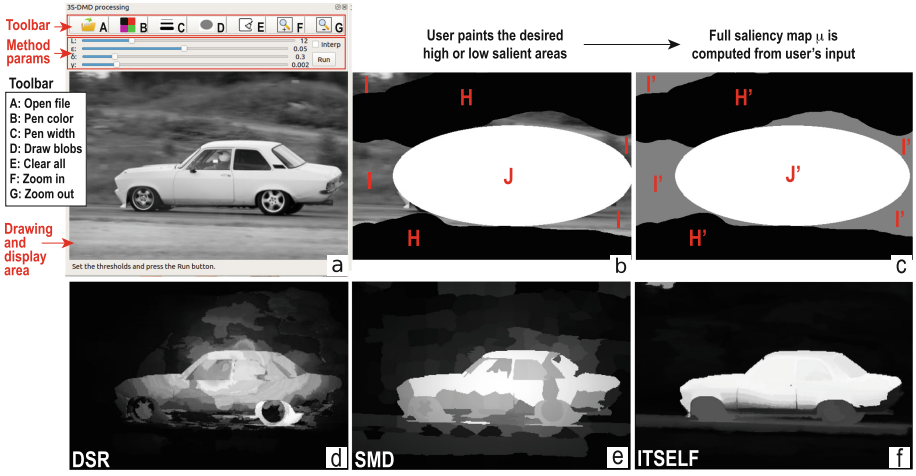

Fig. 3. Directly saliency drawing. (a) Interface with a loaded image. (b) User drawing to specify the saliency. (c) Computed saliency map. (d–f) Generated saliency maps.

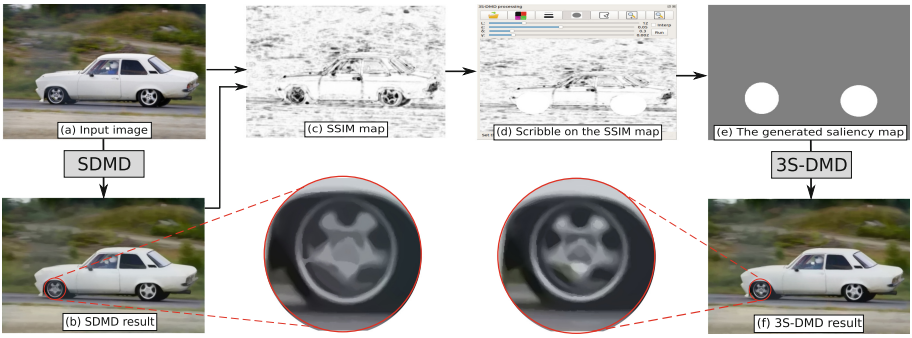

Fig. 4. SSIM-guided user-specified saliency map generation.

Direct Drawing: Users can directly paint a saliency map with various brushes, whose brightness gives the desired saliency (black=0, white=1). Figure 3b shows the drawing of a map for the car image in Fig. 3a. The user marked the car area as highly salient (white ellipse, region J) and background areas farther from the car as zero salient (black scribbled bands, region H). Figure 3c shows the computed saliency map μ. Regions where the user painted saliency are taken over from the drawing (H' and J' are copies of H and J, respectively). Unpainted areas carry no hints that the user found them important or not (Fig. 3b, region I). We set here the saliency to the average value $\mu = 0.5$ (Fig. 3b, region I').

Adjust Precomputed Saliency Map: Fully painting a custom saliency map can be cumbersome, especially when one wants to use multiple saliency values. We support this use-case by allowing users to draw to modify a precomputed saliency map. Figures 3d–f show three such precomputed maps obtained with the DSR method [25], structured matrix decomposition (SMD) method [33], and the recent ITerative Saliency Estimator fLexible Framework (ITSELF) [29] method.

SSIM-guided User-specified Saliency: Users may be unfamiliar with, or unable to run, existing saliency estimation methods. Also, they may not know how to tweak saliency to get the best quality-compression balance. We address these issues by computing the saliency map in a *corrective* way, *i.e.*, by comparing the compression method's output with its input. Figure 4 shows how this works. Given an input image (a), we first run SDMD without a saliency map. Next, we evaluate the quality of the output (b) by the Structural SIMilarity (SSIM) metric [52]. The generated SSIM map (c) shows the per-pixel structural similarity between the original (a) and the output (b), with darker pixels indicating less similar regions. Figure 4 (c) shows that SDMD yields poorer quality over several car details, especially its two wheels. Having this insight, we scribble bright colors on the two wheels to tell their importance (Fig. 4 (d)). We now use this quite simple saliency map (e) to run 3S-DMD to generate a new result (f). As visible in the last image, the quality of the left front wheel has improved.

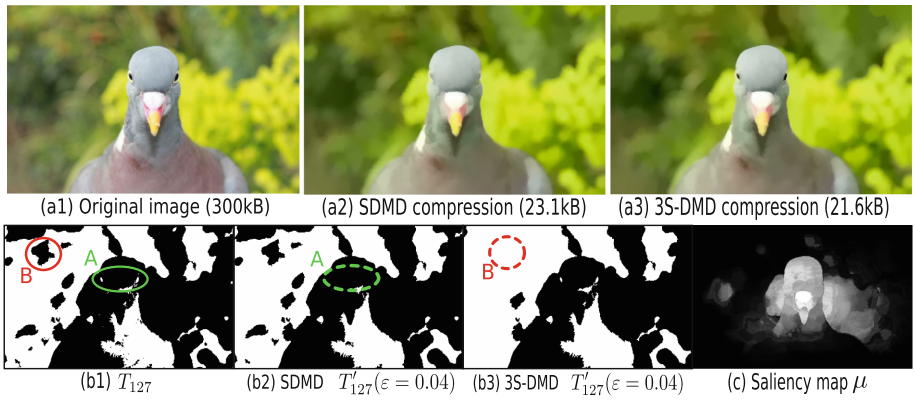

Fig. 5. Pigeon image (a1) encoded with SDMD (a2) and 3S-DMD (a3) with saliency map (c) computed by the DSR [25] method. Images (b1–b3) show details in layer 127.

3.2 Saliency-Based Parameter Control

We next show how to use the saliency maps created by the various methods in the denoising (step 2), regularization (step 4), and spline fitting (step 5) of our end-to-end pipeline (Fig. 2).

Salient Islands Detection: As explained in Sect. 2.1, (S)DMD only keeps islands, or connected components C_i, which meet the condition $|C_i| \geq \varepsilon |T_i|$. This can remove small but salient features (see Fig. 5): For $\varepsilon = 0.04$, the pigeon's eyes, visible in the original image (a1), are removed (a2). We confirm this by verifying that the small islands in region A in the threshold-set T_{127} (b1) get lost in T'_{127} (b2). Lowering ε can alleviate this, but this allocates more information to encode the less important background, thereby increasing image size. To address this, we use the saliency map μ to compute a saliency-aware metric

$$C_i^{\mu} = \sum_{\mathbf{x} \in C_i} k_1^{2\mu(\mathbf{x})-1}, \tag{5}$$

and next remove only islands for which $C_i^{\mu} < \varepsilon$. The factor k_1 in Eq. 5 controls how much μ affects island removal. For $k_1 = 1$, $C_i^{\mu} = |C_i|$, so our method behaves like the original (S)DMD. In practice, we set $k_1 = 5$, which means that the most salient pixels ($\mu(\mathbf{x}) = 1$) are given five times their original unit weight; the least important pixels ($\mu(\mathbf{x}) = 0$), in contrast, get one-fifth of their original unit weight. This keeps small-size, but salient, details in the compressed image. Figure 5 (b3) shows this for a saliency map computed with the DSR method [25]. Islands in region A, while small, have a high μ, so they are retained. In contrast, although large, the island in region B has a low saliency, so it is removed. This ends up with a smaller size, but perceptually better, result (a3).

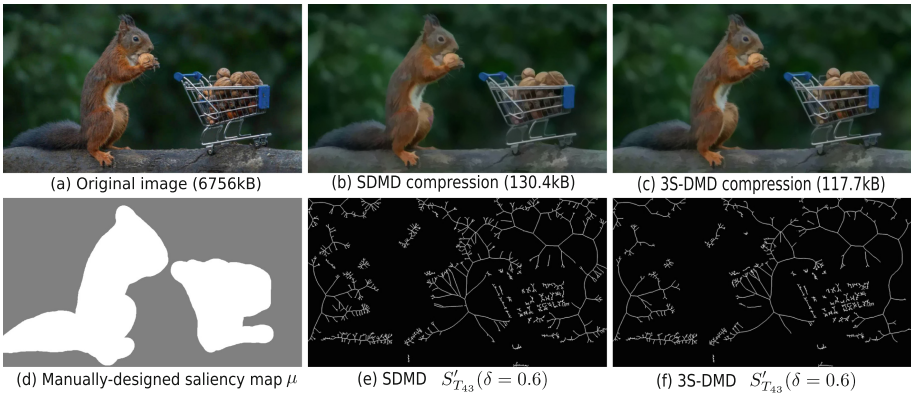

Fig. 6. Salient skeleton detection on a squirrel image (a) with SDMD (b, e) and 3S-DMD (c, f) using a manually-designed saliency map (d).

Saliency-aware Skeleton Simplification: As Sect. 2.1 outlined, (S)DMD regularizes skeletons S'_{T_i} by keeping only pixels $\mathbf{x} \in S_{T_i}$ where $\sigma(\mathbf{x})$ exceeds a user-set threshold δ (Fig. 2, step 4). The SSDMD method [49] further simplifies S'_{T_i} by removing points $\mathbf{x} \in S'_{T_i}$ whose saliency $\mu(\mathbf{x})$ is smaller than a *new* threshold. This not only increases the number of thresholds users have to deal with but also yields poor quality as low-saliency areas get completely removed. SSDMD alleviates this by using various heuristics such as selective layer keeping and interpolation tricks. However, this makes the end-to-end method quite complex.

In contrast to SSDMD, we blend σ with the saliency map μ by computing

$$\sigma'(\mathbf{x}) = \sigma(\mathbf{x}) \cdot k_2^{(\mu(\mathbf{x})-1)} \tag{6}$$

and then obtain S'_{T_i} by upper-thresholding $\sigma'(\mathbf{x})$ with the user-set value δ, *i.e.*,

$$S'_{T_i} = \{\mathbf{x} \in S_{T_i} | \sigma'(\mathbf{x}) > \delta\}. \tag{7}$$

The value k_2 in Eq. 6 controls how much μ affects the skeleton simplification. For $k_2 = 1$, our new metric σ' equals the original σ from (S)DMD. In practice, we set $k_2 = 2$. Hence, the salient-skeleton values $\sigma'(\mathbf{x})$ of the least important pixels ($\mu(\mathbf{x}) = 0$) become half of their original $\sigma(\mathbf{x})$ values; in contrast, the σ values of the most important pixels ($\mu(\mathbf{x}) = 1$) stay unchanged. Figure 6 shows the improvement given by our new metric σ'. Images (e, f) show the regularized skeletons $S'_{T_{43}}$ of one layer, T_{43}, computed with SDMD's σ metric and 3S-DMD's σ' metric, for the same user-set $\delta = 0.6$, and a simple manually-designed saliency map, for illustration purposes (image d). We see how 3S-DMD (image f) simplifies skeletons in the image background more, since the saliency is low there, than SDMD (image e), which has no notion of a low-importance background. In contrast, in the foreground image areas (white areas in the saliency map μ), the 3S-DMD and SDMD skeletons are identical. As a result, 3S-DMD yields the same image quality as SDMD, but with about 10% extra compression.

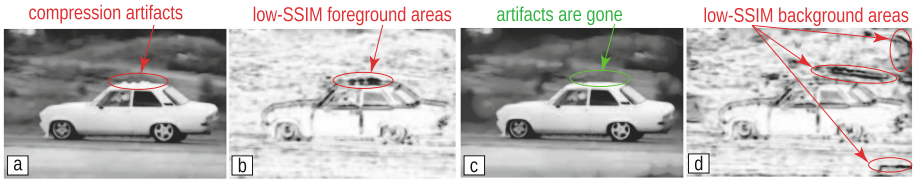

Fig. 7. DMD compression has artifacts (a) found as low-SSIM regions (b). SSDMD (c) removes these but marks subtle background differences important for quality (d). Image taken from [49].

Saliency-Based Spline Fitting: Section 2.2 stated that SDMD finds the minimal number of B-spline control points needed to reach a user-given fitting error γ between a skeleton branch B_i and the B-spline C_i. This error is given by the Hausdorff distance $H(B_i, C_i)$ computed over all pixels $\mathbf{x} \in B_i$. We modify the fixed user-set threshold γ to involve the saliency map μ by

$$\gamma' = \gamma \frac{\sum_{\mathbf{x} \in B_i} k_3^{(1-\mu(\mathbf{x}))}}{|B_i|}, \tag{8}$$

where k_3 controls how much μ influences the spline fitting. For $k_3 = 1$, γ' equals the original γ. We set $k_3 = 2$ in practice. Hence, when a branch is fully within a zero-saliency region ($\mu(\mathbf{x}) = 0$), $\gamma' = 2\gamma$, *i.e.*, we allow a double fitting error as compared to the original SDMD. For branches located in a maximum saliency regions ($\mu(\mathbf{x}) = 1$), the fitting error stays the same, *i.e.*, $\gamma' = \gamma$.

3.3 Saliency-Aware Quality Metric

Quality metrics $Q(I, \tilde{I}) \in \mathbb{R}^+$ measure how close a compressed image $\tilde{I}$ is to the original image I. Such metrics include the mean squared error (MSE) and peak signal-to-noise ratio (PSNR). While simple to compute and with clear physical meanings, these metrics do not match well *perceived* visual quality [53]. The SSIM index [52] alleviates this by measuring, pixel-wise, how perceptually similar two images are. Wang et al. [54] proposed Multiscale SSIM (MS-SSIM), which is an advanced top-down interpretation of how the human visual system comprehends images considering variations of image resolution and viewing conditions.

While MS-SSIM models human perception well, it handles focus (high $\mu(\mathbf{x})$) and context (low $\mu(\mathbf{x})$) areas identically. Figure 7 illustrates this: Image (a) shows the DMD result of a car image and (b) shows the SSIM map. Image (a) shows some artifacts on the car roof, also visible as dark areas in the SSIM map (b). Image (c) shows the SSDMD result [49] of the car image, with strong background simplification and detail retention in the focus (car) area. The car-roof artifacts are removed, so (c) matches better the original image than (a). Yet, the MS-SSIM score of (c) is much *lower* than for DMD (0.9088 *vs* 0.9527). The large dark regions in the SSIM map background (d) explain this: While the saliency map μ clearly says that background is unimportant, MS-SSIM considers it *equally* important to foreground.

Given the above, saliency data should be considered by a perception-aware quality metric. This is also reflected by saliency-based objective metrics [2,14,24,26,27]

Fig. 8. Comparison of DMD (a1–c1) with SSDMD (a2–c2) for three focus-and-context images. For each image, we show the standard MS-SSIM quality Q, and spatial-saliency-aware MS-SSIM Q^{μ}. Image taken from [49].

which integrate a visual saliency map into the quality metric as a weighting map, thereby improving image-quality prediction performance. We follow the same idea by integrating the saliency map μ into the MS-SSIM [54] pooling function, as follows. Take the MS-SSIM metric for a reference image I and a distorted image $\tilde{I}$

$$Q(I,\tilde{I}) = \Big(\mathrm{SSIM}(I,\tilde{I})\Big)^{\beta_M} \prod_{j=1}^{M-1} \Big(c_j(I,\tilde{I})\Big)^{\beta_j}, \tag{9}$$

where c_j is the contrast map $c(I,\tilde{I})$ iteratively downsampled by a factor of 2 on scale $1 \leq j \leq M$; $\mathrm{SSIM}(I,\tilde{I})$ is the structural similarity of I and $\tilde{I}$ on scale M [52]; and the factor β_j models the relative importance of different scales. We weigh Q by the saliency map μ yielding the saliency-aware quality metric

$$Q^{\mu} = \left(\frac{\sum_{\mathbf{x}\in I} \mu(\mathbf{x})\mathrm{SSIM}(\mathbf{x})}{\sum_{\mathbf{x}\in I} \mu(\mathbf{x})}\right)^{\beta_M} \prod_{j=1}^{M-1} \left(\frac{\sum_{\mathbf{x}\in I} \mu_j(\mathbf{x}) c_j(\mathbf{x})}{\sum_{\mathbf{x}\in I} \mu_j(\mathbf{x})}\right)^{\beta_j}, \tag{10}$$

where μ_j is the saliency map at scale j. For notation brevity, the arguments I and $\tilde{I}$ are omitted in Eq. 10. Using Q^{μ} instead of Q allows in-focus values (high $\mu(\mathbf{x})$) to contribute more to similarity than context values (low $\mu(\mathbf{x})$).

Figure 8 compares the results of DMD (a1–c1) and SSDMD (a2–c2) for three focus-and-context images, using the standard MS-SSIM quality Q and our spatial-saliency-aware quality Q^{μ}. The Q values for SSDMD are lower than those for DMD, which suggests that SSDMD has a poorer quality than DMD. Yet, we see that SSDMD creates images that are visually almost the same as DMD, in line with the almost identical Q^{μ} values for SSDMD and DMD. Thus, we argue that Q^{μ} is a better quality measure for focus-and-context simplification than Q. We consider Q^{μ} next for evaluating the image quality.

4 Results

Section 3 proposed 3S-DMD, a method that incorporates three schemes for users to create a spatial saliency map, three ways for adjusting the original SDMD with these maps, and a saliency-aware quality metric Q^{μ} to measure how well the reconstructed image $\tilde{I}$ captures the input image I. We next evaluate 3S-DMD's results in detail, as follows.

- First, we describe our evaluation methodology (Sect. 4.1).
- We show how 3S-DMD depends on its free parameters (Sect. 4.2).
- We compare 3S-DMD with DMD [50] and SDMD [47] (Sect. 4.3).
- Atop [49], we also compare with the JPEG and JPEG 2000 methods (Sect. 4.4).

4.1 Evaluation Methodology

The 3S-DMD encoding consists of a tuple $(w, h, \{\mathbf{l}_i\})$, *i.e.*, the pixel width w and height h of the input image I, and the L selected layers $\mathbf{l}_i$. A layer $\mathbf{l}_i = (i, f, \{\mathbf{b}_i^k\})$ has an intensity value i, a flag f that tells if it uses upper- or lower-thresholding (for details, see [47]), and a B-spline set $\{\mathbf{b}_i^k\}$ encoding its MAT. Each B-spline $\mathbf{b}_i^k = (d_i^k, \{\mathbf{c}_j\})$ has a degree $d_i^k \in \mathbb{N}$ and control points $\mathbf{c}_j \in \mathbb{R}^3$ (see Sect. 2.2).

Sizes of the images $\tilde{I}$ and I are typically measured by bits per pixel (bpp), *i.e.*, the number of bits used to encode a pixel's grayscale or color value [12]. Yet, in an encoding context, we want to *compare* the sizes of $\tilde{I}$ and I, rather than measure their absolute sizes. For this, we define $CR = |I|/|3SDMD(\tilde{I})|$. Here, $|3SDMD(\tilde{I})|$ is the byte-size of the 3S-DMD storage scheme outlined above, while $|I|$ is the size (in bytes) of the original image I.

The quality Q^{μ} (Sect. 3.3) and compression ratio CR of 3S-DMD depend on four parameters (Fig. 2): the number of selected threshold-sets L, the size of removed islands ε, the skeletal saliency threshold δ, and the spline fitting tolerance γ. We establish *ranges* for these parameters based on results of previous work [47,49,50], as follows: $L \in [1, 60]$, $\varepsilon \in [0.001, 0.1]$, $\delta \in [0.01, 3]$, and $\gamma \in [0.001, 0.005]$. We further sample these ranges by the following representative values: $L \in \{15, 25\}$, $\varepsilon = 0.02$, $\delta \in \{0.3, 0.8\}$, and $\gamma = 0.0015$. We use these values to compare DMD, SDMD, and 3S-DMD (Sect. 4.3) and 3S-DMD with JPEG and JPEG 2000 (Sect. 4.4). We test all these methods on a 50-image database, which is selected randomly from the MSRA10K [9], SOD [30], and ECSSD [37] benchmarks. In addition to these real-world pictures, we also tested 3S-DMD on several artificially-designed images (Sect. 4.4). All test images have a resolution between 1000^2 to 2000^2 pixels.

4.2 Effect of Parameters

To intuitively illustrate how 3S-DMD performs for different parameter values, we first group these into *weighting factors* and *user thresholds*, and show the effect of these for a specific image.

Fig. 9. Progressive simplification of a flower image (a) using a saliency map (b) for different weight values k_1, k_2, and k_3 (c1–c4).

Weighting Factors Effect: As explained in Sect. 3.2, the k_1, k_2, and k_3 factors control how much the saliency map μ affects the island detection, skeleton simplification, and spline fitting, respectively. We call these weighting factors—in contrast to the user parameters discussed next—since they are more technical parameters, which do not arguably need to be exposed to end users. Secondly, their effect is strongly related to the way 3S-DMD treats image areas of different saliency. Let 'foreground' and 'background' describe areas of high, respectively, low saliency map μ values. Simply put, increasing all (or any) of these three weighting factors progressively simplifies the image background, similarly to a (soft) blurring effect, but keeps the image foreground relatively untouched. Figure 9 shows this for a flower image under different values for k_1, k_2, and k_3. The user parameters are all fixed to the default values $L = 25, \varepsilon = 0.02, \delta = 0.3, \gamma = 0.0015$.

The setting $k_1 = k_2 = k_3 = 1$, shown in Fig 9 (c1), corresponds to the original SDMD method since, for this setting, μ has no effect on island detection, skeleton regularization, and spline fitting (see Eqs. 5, 6, and 8). As we increase k_1, k_2, and k_3, the image background gets progressively more simplified; see Figs. 9 (c2–c4). However, the flower in the foreground stays roughly the same in all images. The CR and Q^μ values shown below the images match the above observations: as the weights increase, Q^μ drops only slightly decreases, but CR increases strongly. In practice, as stated in Sect. 3.2, we found $k_1 = 5$, $k_2 = 2$, and $k_3 = 2$ to be a good default for balancing CR and Q^μ.

User Thresholds Effect: 3S-DMD depends on four thresholds, as follows:

– L controls how smoothly the simplified image captures *color gradients*; larger values yield smoother gradients;

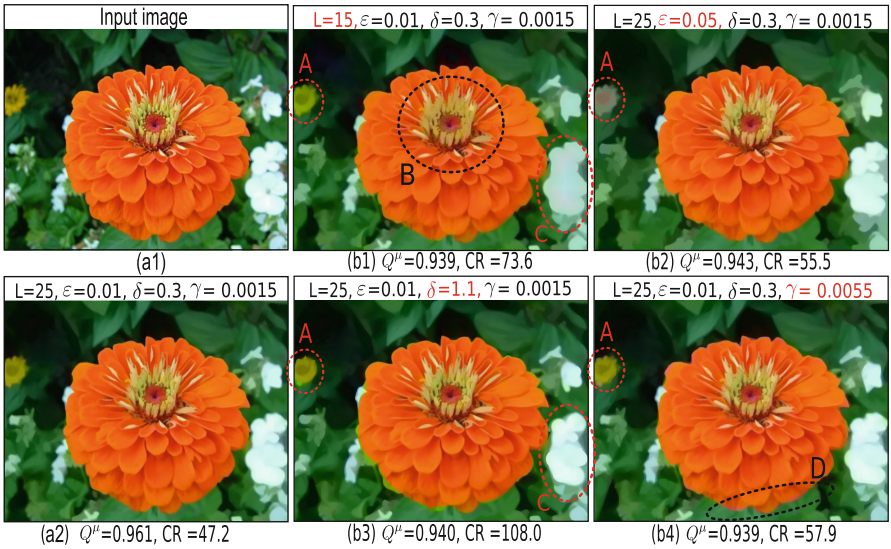

Fig. 10. 3S-DMD results for a flower image (a) using the saliency map of Fig. 9(b) for different combinations of parameters L, ε, δ, and γ.

- ε gives the scale of *details* that are kept in the image; larger values remove larger details;
- δ controls the scale of *corners* that are kept in the image; larger values round off larger corners;
- γ tells how accurately B-splines fit skeleton branches; larger values yield more distorted results.

In contrast to the weighting factors discussed earlier, these four thresholds significantly influence the 'style' of the simplified image. Hence, we believe they are best left under the direct control of the end users.

Figure 10 shows the effect of the thresholds L, ε, δ, and γ by showing the 3S-DMD results on the same flower image, using the same saliency map, as in Fig. 9. Image (a2) shows the results of 3S-DMD when setting user thresholds. The remaining images (b1–b4) are each the effect of a single user threshold change (red in the legend). If we decrease L(image (b1)), even if we select only$L = 15$ layers, we still get a visually convincing result. Yet, the stamens in region B and the flowers in regions A and C look duller than in image (a). Image (b2) uses a higher ε value, which removes many large islands in the image background, *e.g.*, the one corresponding to the yellow flower in region A. Image (b3) uses a higher δ, which rounds off corners of background shapes, *e.g.* the flowers in regions A and C. Finally, image (b4) uses a higher γ, which distorts the boundaries of the flower in region A and creates subtly false colors in region D.

4.3 Comparison with DMD and SDMD

Figure 11 compares the Q^{μ} and CR values of DMD (blue markers), SDMD (red markers), and our proposed 3S-DMD (green markers), for the four user-parameter settings

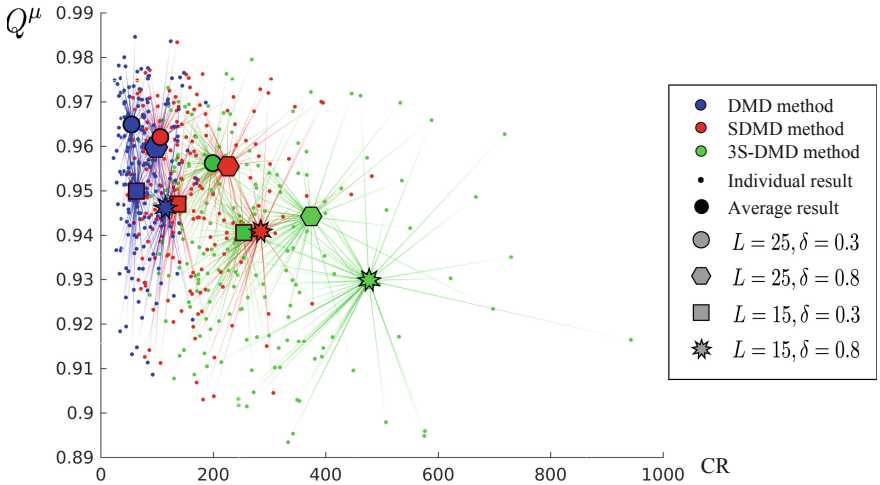

Fig. 11. Star plots of DMD (blue markers), SDMD (red markers), and 3S-DMD (green markers) for 50 real-world images. The actual image data (smaller dots) are connected to the corresponding average value (star center markers) for each method. Marker shapes indicate the four parameter settings being used. (Color figure online)

listed in Sect. 4.1, using a star plot. Small dots indicate metric values for a run involving a method-and-parameter-setting on a single image. Markers at the 'star centers' show average values for all runs over the 50 images in the benchmark for one parameter setting and one method. For each method (color), there are four stars, one for each of the four parameter-settings used, as indicated by the four glyph types in the figure's legend. The star center triples depicted using the same glyph show runs that use the same parameter settings. We fixed $\varepsilon = 0.02$ and $\gamma = 0.0015$ so these user parameters are not listed in the figure's legend.

Figure 11 offers several insights. Small stars show little variance in CR and Q^μ from the average for a given method-and-parameter-set. Large stars indicate more variance as a function of the actual images. The sizes and shapes of the stars in the figure are quite similar. Hence, DMD, SDMD, and 3S-DMD show a similar dependency of CR and Q^μ on the real-world image type. This is due to the fact that SDMD and 3S-DMD inherit the thresholding and skeletonization used by DMD. Yet, the green stars are slightly larger and more spread horizontally, indicating that 3S-DMD can produce greater changes in CR for similar Q^μ.

For each color (method), its four stars show an inverse correlation of CR with Q^μ. Indeed, more layers and smaller δ yield higher quality but less compression; conversely, fewer layers and larger δ slightly reduce quality, but strongly increase compression. The axes ranges show this too: CR varies roughly from 50 to 700, while quality varies between 0.91 and 0.98. The three large dots of the same glyph types let us compare the DMD, SDMD, and 3S-DMD methods under the same parameter setting. We see a clear inverse correlation pattern going from high Q^μ and low CR (DMD, blue dots) to average Q^μ and CR (SDMD, red dots) and then to lower Q^μ and highest CR (3S-DMD, green dots). Hence, 3S-DMD always gets higher CR than DMD and SDMD

for only a small quality loss. On average, 3S-DMD increases CR by 234.2% relative to DMD, while Q^{μ} drops by only 0.014. Compared with SDMD, 3S-DMD increases CR on average by 53.8%, while Q^{μ} drops by a tiny 0.009. More importantly, when we compare CR and Q^{μ} for different parameter settings, *e.g.*, comparing the large round green marker with the star-shaped blue marker and square red marker, 3S-DMD not only yields a higher CR but also better quality.

Fig. 12. Comparison of 3S-DMD (a3–f3) with DMD (a1–f1) and SDMD (a2–f2) for six focus-and-context images. For each result, we show the saliency-aware MS-SSIM Q^{μ} and CR. The rightmost column shows the manually-modified DSR saliency map.

Figure 12 further compares the three methods for six focus-and-context images of insects, birds, animals, and plants from the MSRA10K benchmarks [9]. More results

are given in the supplementary material [46]. The zoomed-in areas show that, compared with DMD (a1–f1) and SDMD (a2–f2), 3S-DMD (a3–f3) preserves well important features marked as such by the saliency maps, like highlights (a, d), animals' eyes (b, c, f), and the flower stamen (e). For background areas, all three methods perform visually roughly the same. The quality values Q^{μ} are also similar for the three methods, with 3S-DMD scoring twice as best, three times as second-best, and once in the third place. CR values show that 3S-DMD achieves (significantly) higher compression than DMD and SDMD, except for image (b), where it scores slightly below SDMD. On the other hand, 3S-DMD retains for this example more details than SDMD for the foreground area, such as the bird's eye, as also reflected by its higher quality score.

4.4 Comparison with JPEG and JPEG 2000

Tens of image compression exist, see *e.g.* [1,11,43] and methods cited therein. Comparing 3S-DMD with all of them is not feasible in the scope of this work. However, we provide a comparison with JPEG [44] and JPEG 2000 (J2K) [39] which are arguably among the most well-known, frequently-used, and generic, image compressors.

Comparison with JPEG: Figure 13 compares 3S-DMD with JPEG on our image benchmark. The parameter setting of 3S-DMD (green dots) follows Sect. 4.3. JPEG (blue dots) is run under five quality settings: 10%, 30%, 50%, 70%, and 90%. As in Fig. 11, we use star plots for both 3S-DMD and JPEG: small dots are individual runs and large dots are averages. We see that 3S-DMD cannot reach the same Q^{μ} values as when JPEG uses its 90% quality setting: the topmost blue dot is above the topmost green dot. However, the vertical spread of the blue *vs* green dots shows that the difference in quality (Q^{μ}) is small, about 4% on average. If we accept this small quality loss, 3S-DMD always gets higher compression rates than JPEG. In the limit, compared to JPEG with a quality of 10% (point A), 3S-DMD (point B) gets both higher CR and better quality.

Figure 14 refines the above insights by showing six real-world images (building, plant, animal, natural scene, man-made structure, and people), compressed by 3S-DMD (a1–f1), JPEG (a2–f2), and J2K (a3–f3). We see that JPEG with a 10% quality creates obvious artifacts: checkerboarding (b2, c2, e2, f2), banding (a2, c2), and color faking (d2). 3S-DMD yields better quality (Q^{μ}) and does not exhibit such artifacts. Yet, 3S-DMD loses small-scale, faint, details in the background, like the gravel in the sea (c1) and the red color of the traffic sign (f1). We argue that these are acceptable losses since these details are located in low-saliency areas. Separately, 3S-DMD always achieves higher CR than JPEG.

Comparison with J2K: Figure 13 shows J2K (red dots) run under five fixed compression ratios: 100, 200, 300, 400, and 500. As CR increases, J2K has only a slightly quality loss and performs practically always better than JPEG. Figure 14 also verifies this: J2K's quality Q^{μ} is always higher than 0.99 and the compressed results are indistinguishable from the originals. 3S-DMD cannot (yet) achieve such quality. However, 3S-DMD can obtain comparable, and sometimes higher, CR values. We further refine the comparison with J2K by considering a narrower class of artificially made images,

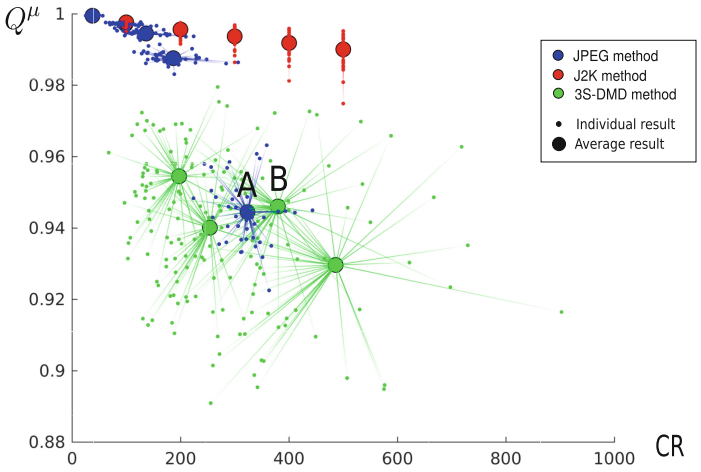

Fig. 13. Comparison of JPEG (blue dots), J2K (red dots), and 3S-DMD (green dots) for 50 images. The actual image data (smaller dots) are connected to the corresponding average value (larger dots) for each parameter setting of the three methods. (Color figure online)

such as graphics art (logos, graphics design), scientific visualization images, synthesized images using graphics rendering and vectorization methods [32], and cartoon images. For such images, 3S-DMD produces both higher CR *and* quality than J2K and JPEG. Figure 15 shows four representative images, one from each of the above four categories, compressed with 3S-DMD (a1–d1), JPEG (a2–d2), and J2K (a3–d3). As in earlier cases, JPEG with a quality of 10% generates obvious artifacts such as blocking (a2, b2, c2, d2), banding (c2), and color faking (c2), and has a CR well below the other two methods. When compared with J2K, our method yields similar Q^μ values. We show some zoomed-in areas to expose a few subtle differences: For the graphics design example (a), 3S-DMD achieves visually much better results, without the checkerboarding and blur artifacts of J2K. This is also seen in the first image in Fig. 15 where 3S-DMD got a higher Q^μ than J2K. For the second image (b) in Fig. 15, 3S-DMD captures the smooth luminance gradient in the shadow area quite well. In contrast, J2K causes a slight amount of false color artifacts. For the strong-contrast images (c) and (d), J2K creates some small-scale blur artifacts. 3S-SDMD does not have such problems but suffers from a slight color change issue due to its selection of threshold-sets to be encoded. Most importantly, with a similar or better quality, 3S-DMD always yields higher compression than J2K for such synthetic images.

We conclude that, for real-world images, 3S-DMD gets both higher CR and quality than JPEG but cannot match J2K's quality at the same CR. Yet, for synthetic images, 3S-DMD gets both much higher CR and quality than JPEG, and also achieves higher CR at similar quality but with fewer artifacts than J2K.

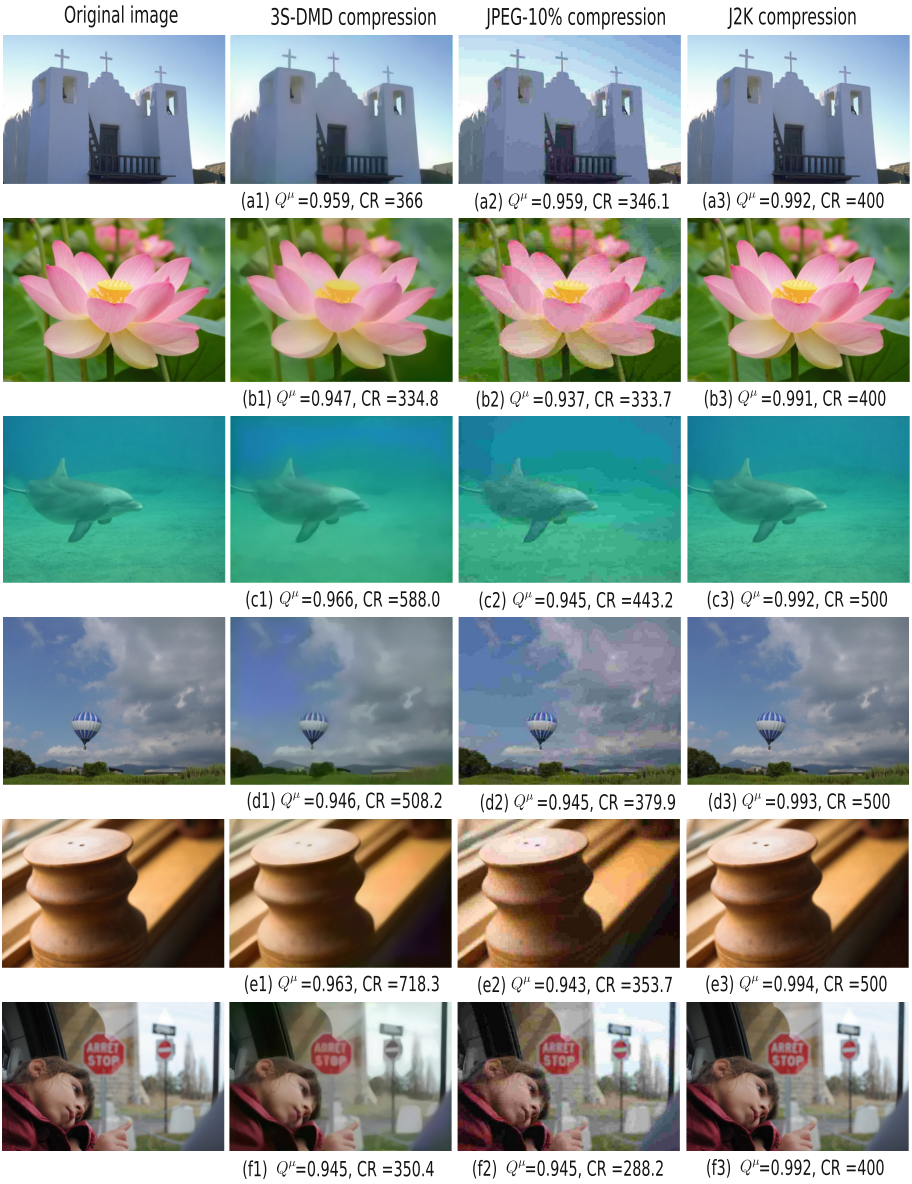

Fig. 14. Comparison of 3S-DMD (a1–f1) with JPEG -10% (a2–f2) and J2K (a3–f3) for six real-world images. For each image, we show the saliency-aware metric Q^{μ} and CR.

5 Discussion

We now discuss several aspects of our 3S-DMD image compression method.

Genericity and Ease of Use: 3S-DMD is a general-purpose compression method for generic grayscale and color images. It relies on well-tested and robust algorithms such as the skeletonization method in [17,42] and the least-squares B-spline fitting algorithm [13]. In contrast to segmentation tasks [16], 3S-DMD does not require precise saliency maps. Any saliency map that encodes which image areas are more important and which less for an application at hand can be used. 3S-DMD has four user parameters: the number of selected layers L, island size ε, skeleton saliency threshold δ, and spline fitting error γ. These parameters have intuitive effects and default values, as detailed in Sect. 4.2.

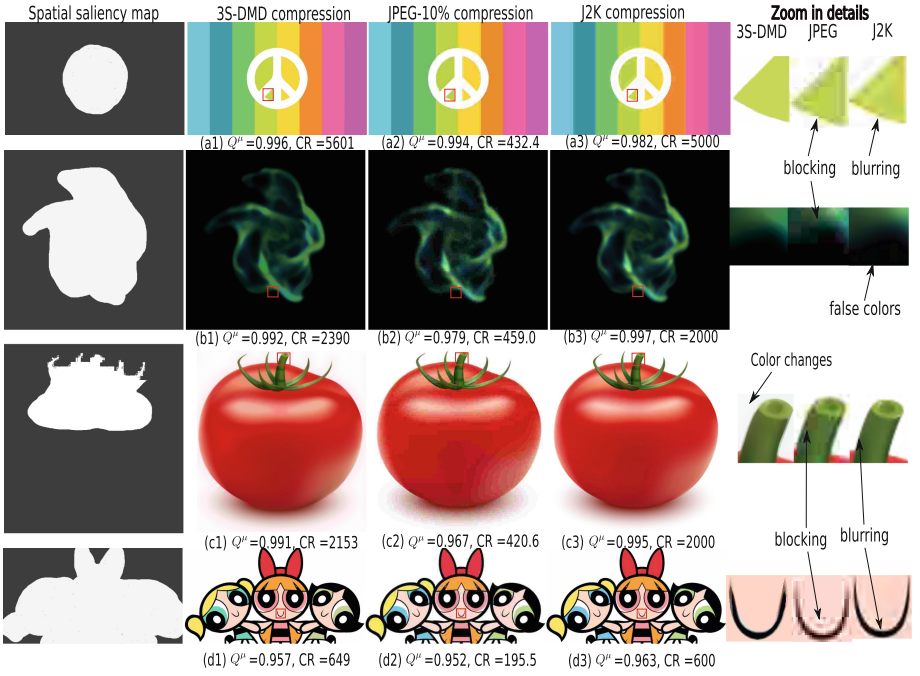

Fig. 15. Comparison of 3S-DMD (a1–d1) with JPEG-10% (a2–d2) and J2K (a3–d3) for four synthetic images. For each image, we show the saliency-aware Q^μ and CR. The leftmost column shows the saliency map obtained by directly scribbling on the input image. The rightmost three columns show zoomed-in areas for 3S-DMD, JPEG, and J2K for detailed comparison.

Speed: We compute the most complex step in 3S-DMD, skeletonization, on the GPU [7,41]. On a Linux PC with an Nvidia RTX 2060, this takes a few hundred milliseconds for images up to 1024^2 pixels. Spline fitting uses about 1 s per color channel, yielding a total of about 3 to 4 s for the compression.

Replicability: We provide our full C++ source code and data for replication purposes, as well as a demo video and additional comparisons with DMD and SDMD, in the supplementary material [46].

Limitations: Image layer components that are one or two pixels thin cannot be encoded by MATs, so 3S-DMD cannot deal optimally with images with many thin-and-long details, such as animal fur, fine textures, and greenery. Figure 16 shows this for two such images. For smooth regions in the background (red boxes), 3S-DMD yields results that are indistinguishable from the originals. However, 3S-DMD cannot capture all the fine-grained details present in the foreground (green boxes). One way to handle such cases is to artificially upscale the images, leading to fine details thicker than a few pixels, which next can be skeletonized with no problems. Studying how to perform this efficiently and with good CR values is an interesting topic for future work.

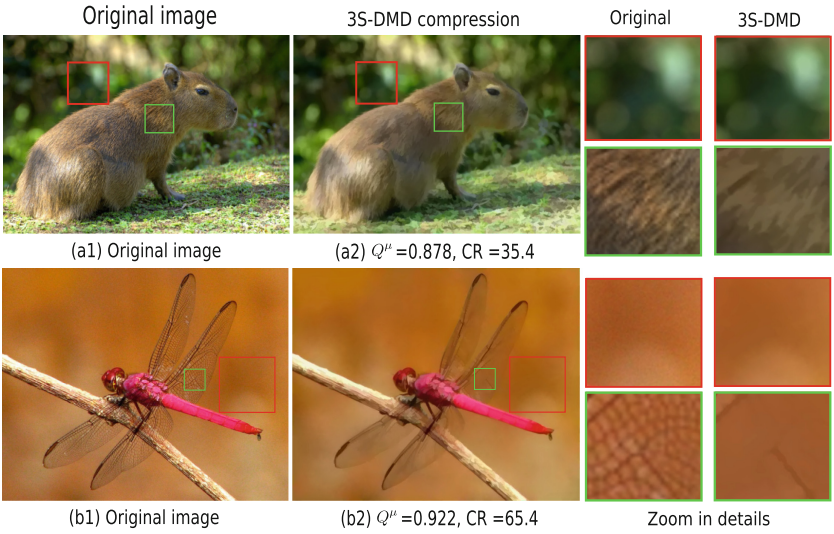

Fig. 16. Poor performance for 3S-DMD when handling images with many small-scale details, such as animal furs (a) and fine textures (b).

6 Conclusion

We have presented 3S-DMD, a method for saliency-aware image simplification and compression. 3S-DMD combines the strengths of two of its precursors: SSDMD [49]

that allows spatial control of image simplification, and SDMD [47] that compactly encodes MATs with B-splines. We have developed an interactive application for users to set parameters and customize saliency maps in three ways. We have illustrated how saliency maps involved in the SDMD pipeline offer spatially-dependent simplification. We have shown graphically and intuitively how 3S-DMD performs under different parameter combinations. To study the effectiveness of 3S-DMD, we have considered a database of 50 real-world images. Quantitative evaluation showed that 3S-DMD greatly improves the compression of SSDMD and SDMD at only a small quality loss. Our method delivers both higher CR and quality than JPEG. While we cannot reach the same high quality at the same CR values as J2K, our method yields similar quality, higher CR, and fewer artifacts for a wide class of synthetic images.

We next aim to consider more extensive comparisons with additional compression techniques, *e.g.*, deep neural network methods. Separately, we aim to extend 3S-DMD beyond grayscale or color image simplifications to simplify 3D scalar fields in scientific visualization, weighted with uncertainty-encoding maps.

Acknowledgments. The first author acknowledges the China Scholarship Council (Grant number: 201806320354) for financial support.

References

1. Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., Van Gool, L.: Generative adversarial networks for extreme learned image compression. In: ICCV, pp. 221–231 (2019)
2. Alaei, A., Raveaux, R., Conte, D.: Image quality assessment based on regions of interest. Signal Image Video Process. **11**, 673–680 (2017)
3. Andrushia, A., Thangarjan, R.: Saliency-based image compression using Walsh and Hadamard transform. In: Lect Notes Comp Vision Biomech, pp. 21–42 (2018)
4. Attali, D., Montanvert, A.: Computing and simplifying 2D and 3D continuous skeletons. Comput. Vision Image Understand. **67**(3), 261–273 (1997)
5. Ballard, F.: Better portable graphics (2018). https://bellard.org/bpg
6. Borji, A., Cheng, M., Jiang, H., Li, J.: Salient object detection: a benchmark. IEEE TIP **24**(12), 5706–22 (2015)
7. Cao, T.T., Tang, K., Mohamed, A., Tan, T.S.: Parallel banding algorithm to compute exact distance transform with the GPU. In: Proceedings ACM I3D, pp. 83–90 (2010)
8. Chen, T., Cheng, M., Tan, P., Shamir, A., Hu, S.: Sketch2photo: Internet image montage. ACM TOG **28**(5) (2009)
9. Cheng, M.: MSRA10K salient object database (2014). mmcheng.net/msra10k
10. Cheng, M., Mitra, N.J., Huang, X., Torr, P.H., Hu, S.: Global contrast based salient region detection. IEEE TPAMI **37**(3), 569–582 (2014)
11. Choi, Y., El-Khamy, M., Lee, J.: Variable rate deep image compression with a conditional autoencoder. ICCV pp. 3146–3154 (2019)
12. Daintith, J., Wright, E.: A Dictionary of Computing. Oxford Univ, Press (2008)
13. Eberly, D.: Least-squares fitting of data with B-spline curves (2014). geometric Tools. www.geometrictools.com/Documentation/BSplineCurveLeastSquaresFit.pdf
14. Engelke, U., Le Callet, P.: Perceived interest and overt visual attention in natural images. Image Commun. **39**(PB), 386–404 (2015)
15. Fabbri, R., Costa, L.D.F., Torelli, J.C., Bruno, O.M.: 2D Euclidean distance transform algorithms: a comparative survey. ACM Comput Surv **40**(1), 1–44 (2008)

16. Falcão, A., Bragantini, J.: The role of optimum connectivity in image segmentation: can the algorithm learn object information during the process? In: Couprie, M., Cousty, J., Kenmochi, Y., Mustafa, N. (eds.) DGCI 2019. LNCS, vol. 11414, pp. 180–194. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14085-4_15
17. Falcão, A., Stolfi, J., Lotufo, R.: The image foresting transform: theory, algorithms, and applications. IEEE TPAMI **26**, 19–29 (2004)
18. Goferman, S., Zelnik, L., Tal, A.: Context-aware saliency detection. IEEE TPAMI **34**(10), 1915–1926 (2011)
19. Hesselink, W.H., Roerdink, J.B.T.M.: Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. IEEE TPAMI **30**(12), 2204–2217 (2008)
20. Jiang, P., Ling, H., Yu, J., Peng, J.: Salient region detection by UFO: Uniqueness, focusness and objectness. In: Proceedings of the ICCV, pp. 1976–1983 (2013)
21. Kimmel, R., Shaked, D., Kiryati, N., Bruckstein, A.M.: Skeletonization via distance maps and level sets. CVIU **62**(3), 382–391 (1995)
22. Kresch, R., Malah, D.: Skeleton-Based morphological coding of binary images. IEEE TIP **7**(10), 1387–1399 (1998)
23. Lam, L., Lee, S., Suen, C.Y.: Thinning methodologies - a comprehensive survey. IEEE TPAMI **14**(9), 869–885 (1992)
24. Le Callet, P., Niebur, E.: Visual attention and applications in multimedia technologies. Proc. IEEE **101**(9), 2058–2067 (2013)
25. Li, X., Lu, H., Zhang, L., Ruan, X., Yang, M.: Saliency detection via dense and sparse reconstruction. In: Proceedings of the ICCV, pp. 2976–2983 (2013)
26. Liu, H., Engelke, U., Wang, J., Callet, Le, P., Heynderickx, I.: How does image content affect the added value of visual attention in objective image quality assessment? IEEE Signal Proc. Let. **20**, 355–358 (2013)
27. Liu, H., Heynderickx, I.: Visual attention in objective image quality assessment: Based on eye-tracking data. IEEE TCSVT **21**(7), 971–982 (2011)
28. Meijster, A., Roerdink, J., Hesselink, W.: A general algorithm for computing distance transforms in linear time. In: Proceedings ISMM, pp. 331–340 (2002)
29. de Melo Joao, L., de Castro Belem, F., Falcao, A.X.: Itself: Iterative saliency estimation flexible framework. Available at https://arxiv.org/abs/2006.16956 (2020)
30. Movahedi, V., Elder, J.: Design and perceptual validation of performance measures for salient object segmentation. In: IEEE Computer Society Conference (2010)
31. Ogniewicz, R., Kübler, O.: Hierarchical voronoi skeletons. Patt. Recogn. **28**(3), 343–359 (1995)
32. Orzan, A., Bousseau, A., Barla, P., Winnemöller, H., Thollot, J., Salesin, D.: Diffusion curves: a vector representation for smooth-shaded images. Commun. ACM **56**(7), 101–108 (2013)
33. Peng, H., Li, B., Ling, H., Hu, W., Xiong, W., Maybank, S.J.: Salient object detection via structured matrix decomposition. IEEE TPAMI **39**(4), 818–832 (2016)
34. Piegl, L., Tiller, W.: The NURBS Book (2nd Ed.). Springer-Verlag (1997). https://doi.org/10.1007/978-3-642-59223-2
35. Pizer, S., Siddiqi, K., Székely, G., Damon, J., Zucker, S.: Multiscale medial loci and their properties. IJCV **55**, 155–179 (2003)
36. Saha, P.K., Borgefors, G., Sanniti di Baja, G.: A survey on skeletonization algorithms and their applications. Patt. Recogn. Lett. **76**, 3–12 (2016)
37. Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended CSSD. IEEE TPAMI **38**(4D) (2016)
38. Siddiqi, K., Pizer, S.: Medial representations: mathematics, algorithms and applications (1nd Ed.). Springer (2008). https://doi.org/10.1007/978-1-4020-8658-8

39. Taubman, D.S., Marcellin, M.W.: JPEG 2000: Image compression fundamentals, standards and practice. Kluwer Academic Publishers (2001)
40. Telea, A.: Feature preserving smoothing of shapes using saliency skeletons. In: Proc. VMLS, pp. 153–170 (2012)
41. Telea, A.: CUDASkel: real-time computation of exact Euclidean multiscale skeletons on CUDA (2019). webspace.science.uu.nl/∼telea001/Shapes/CUDASkel
42. Telea, A., Wijk, van, J.: An augmented fast marching method for computing skeletons and centerlines. In: Eurographics, pp. 251–259 (2002)
43. Toderici, G., et al.: Variable rate image compression with recurrent neural networks. In: 4th ICLR (2016)
44. Wallace, G.K.: The JPEG still picture compression standard. IEEE TCE **38**(1), xviii-xxxiv (1992)
45. Wang, H., Schuster, G.M., Katsaggelos, A.K., Pappas, T.N.: An efficient rate-distortion optimal shape coding approach utilizing a skeleton-based decomposition. IEEE TIP **12**(10), 1181–1193 (2003)
46. Wang, J.: 3S-DMD supplementary material (2021). https://github.com/WangJieying/3S-DMD-resources
47. Wang, J., Kosinka, J., Telea, A.: Spline-based dense medial descriptors for lossy image compression. J. Imag. **7**(8), 153 (2021)
48. Wang, J., Kosinka, J., Telea, A.: Spline-based medial axis transform representation of binary images. Comput. Graph. **98**, 165–176 (2021)
49. Wang, J., de Melo Joao, L., Falcão, A., Kosinka, J., Telea, A.: Focus-and-context skeleton-based image simplification using saliency maps. In: Proceedings of the VISAPP, pp. 45–55. SciTePress (2021)
50. Wang, J., Terpstra, M., Kosinka, J., Telea, A.: Quantitative evaluation of dense skeletons for image compression. Information **11**(5), 274 (2020)
51. Wang, W., Lai, Q., Fu, H., Shen, J., Ling, H.: Salient object detection in the deep learning era: An in-depth survey. IEEE TPAMI PP (2021)
52. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**, 600–612 (2004)
53. Wang, Z., Bovik, A.: Mean squared error: Love it or leave it? a new look at signal fidelity measures. IEEE Signal Proc. Mag. **26**, 98–117 (2009)
54. Wang, Z., Simoncelli, E., Bovik, A.: Multiscale structural similarity for image quality assessment. In: ACSSC, pp. 1398–1402 (2003)
55. Yushkevich, P., Thomas Fletcher, P., Joshi, S., Thall, A., Pizer, S.M.: Continuous medial representations for geometric object modeling in 2D and 3D. Image Vision Comput. **21**(1), 17–27 (2003)
56. Zhang, J., et al.: Hypergraph optimization for salient region detection based on foreground and background queries. IEEE Access **6**, 26729–267241 (2018)
57. Zhu, Y., Sun, F., Choi, Y.K., Jüttler, B., Wang, W.: Computing a compact spline representation of the medial axis transform of a 2D shape. Graphical Models **76**(5), 252–262 (2014)
58. Zwan, M.V.D., Meiburg, Y., Telea, A.: A dense medial descriptor for image analysis. In: Proceedings of the VISAPP, pp. 285–293 (2013)
59. Zünd, F., Pritch, Y., Sorkine-Hornung, A., Mangold, S., Gross, T.: Content-aware compression using saliency-driven image retargeting. In: IEEE ICIP, pp. 1845–1849 (2013)