Dense Medial Descriptors for Image Compression and Manipulation

Wang, Jieying

2 RELATED WORK

As outlined in Chapter 1, our main research question regards the potential of using medial descriptors to implement image simplification and compression. As such, we review related work on medial descriptors (Sec. 2.1), image compression methods (Sec. 2.2), and metrics to evaluate compression quality (Sec. 2.3) in this chapter.

2.1 MEDIAL DESCRIPTORS

Medial descriptors, also known as medial axes, or more generally the skeletons, are efficient and effective tools for a wide range of computer science applications, including shape recognition (Shen et al., 2016; Ayzenberg, 2019), shape segmentation (Shah, 2005; Reniers and Telea, 2007, 2008), matching and retrieval (Bai and Latecki, 2008; Goh, 2008; Sundar et al., 2003; Xie et al., 2008), and shape simplification (Tam and Heidrich, 2003; Hajdu et al., 2007) and denoising (Telea, 2012; Schubert et al., 2020). The process of computing skeletons from given binary shapes is called *skeletonization*. Conceptually, skeletonization $S^{n,m}$ generates m-dimensional descriptors from an n-dimensional shape $\Omega \subset \mathbb{R}^n$ (Reniers, 2009). In this thesis, we explore the potential of medial descriptors for 2D image compression. Thus, we focus on $S^{2,1}$, or, in other words, skeletons of two-dimensional binary shapes. For readers interested in 3D skeletonization $S^{3,2}$ (surface skeletons) and $S^{3,1}$ (curve skeletons) and their applications, we refer to Tagliasacchi et al. (2016); Cornea et al. (2007); Saha et al. (2016). In the remainder of this section, we overview the definitions of medial descriptors (Sec. 2.1.1), skeletonization methods (Sec. 2.1.2), skeleton regularization methods (Sec. 2.1.3), and methods for shape reconstruction from medial descriptors (Sec. 2.1.4).

2.1.1 *Definitions*

Medial descriptors, first introduced by Blum (Blum, 1967), were defined as the loci of centers of maximal discs contained in a shape $\Omega \subset \mathbb{R}^2$, as shown in Fig. 2.1 (a). A maximally inscribed disc is completely inside the shape Ω and is not fully included in another inscribed disc. While such a definition is simple and intuitive, it is difficult to derive an effective way for computing the skeleton in practice from it. Some methods have been able to directly use this definition to compute skeletons (Jalba et al., 2013). However, this typically entails the use of brute-force computation to search for the centers of maximally inscribed discs.

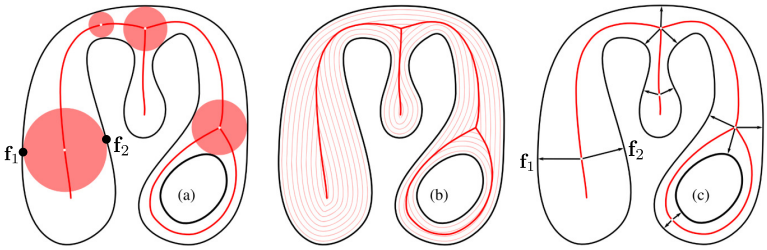


Figure 2.1: Three alternative definitions of medial descriptors: (a) centers of maximally-inscribed discs; (b) shock graph of the grassfire front flow; (c) points with at least two closest points on the boundary. Image taken from Tagliasacchi et al. (2016).

As an alternative, another more commonly used definition, the *grassfire* analogy, was proposed in Leymarie and Levine (1992). Imagine $\Omega \subset \mathbb{R}^2$ as a compact, uniformly growing grass field whose entire boundary $\partial\Omega$ catches fire at the exact same time t_0. The flame propagates from $\partial\Omega$ to the interior of Ω with an isotropic velocity (constant speed) along the internal normal of $\partial\Omega$. The locations where fire fronts coming from different parts of $\partial\Omega$ meet and quench define the skeleton S_Ω of Ω. The arrival time $t > t_0$ of the front is positively related to the inscribed disc radius. The grassfire model definition is depicted in Fig. 2.1 (b). This definition leads to direct techniques, *e.g.*, morphological thinning methods, for computing skeletons, as described next in Sec. 2.1.2.

In the grassfire definition, the medial (skeletal) point, *i.e.*, the quenching point, is always generated by the meeting of at least two fire fronts. Since the grassfire propagates isotropically, a skeleton point always associates with at least two different closest points on the boundary $\partial\Omega$, which are called feature points (Meijster et al., 2002; Hesselink and Roerdink, 2008). This model is shown in Fig. 2.1 (c). These feature points, $\mathbf{f}_1, \mathbf{f}_2$, are exactly the tangent points between the maximally inscribed disc and $\partial\Omega$ in Blum's definition (Fig. 2.1 (a)). Using these feature points, the skeleton S_Ω of Ω can be defined as

$$S_\Omega = \{\mathbf{x} \in \Omega \mid \exists \mathbf{f}_1, \mathbf{f}_2 \in \partial\Omega, \mathbf{f}_1 \neq \mathbf{f}_2 \colon \|\mathbf{f}_1 - \mathbf{x}\| = \|\mathbf{f}_2 - \mathbf{x}\| = DT_\Omega(\mathbf{x})\}. \quad (2.1)$$

Here, DT_Ω is the so-called distance transform of Ω, which is defined as

$$DT_\Omega(\mathbf{x} \in \Omega) = \min_{\mathbf{y} \in \partial\Omega} \|\mathbf{x} - \mathbf{y}\| . \quad (2.2)$$

The notation $\|\cdot\|$ in Eqn. 2.1 and Eqn. 2.2 denotes the Euclidean distance in $\mathbb{R}^2$. Intuitively, $DT_\Omega(\mathbf{x})$ indicates the shortest distance from $\mathbf{x}$ to $\partial\Omega$. Thus, $DT_\Omega(\mathbf{x})$ monotonically increase as $\mathbf{x}$ goes from $\partial\Omega$ towards the interior of Ω; see the example shown in Fig. 2.2 (b).

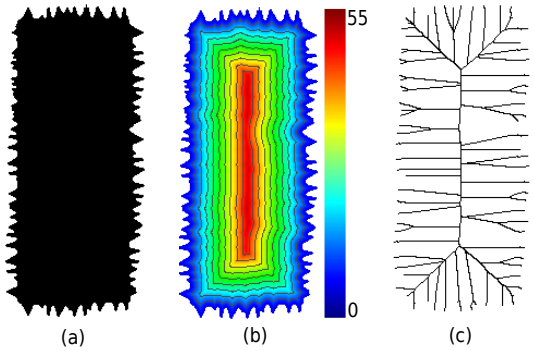


Figure 2.2: A rectangular shape with random noise added to the boundary (a) and its distance transform shown by color-coding and isolines (b), and its corresponding medial axes generated by the Fast Marching Method (FMM) (Sethian, 1996).

For completeness, we note that the concept of feature points defines a so-called feature point transform

$$FT_\Omega(\mathbf{x} \in \Omega) = \arg\min_{\mathbf{y}\in\partial\Omega} \|\mathbf{x} - \mathbf{y}\|\,, \tag{2.3}$$

which associates to each point $\mathbf{x}$ inside Ω the set of its feature points. By definition, a point $\mathbf{x} \in \Omega$ is on S_Ω if $FT_\Omega(\mathbf{x})$ yields at least two points; non-skeletal points $\mathbf{x} \in \Omega$ have a $FT_\Omega(\mathbf{x})$ yielding a single point; and skeleton bifurcations and branch endpoints $\mathbf{x} \in S_\Omega$ have a $FT_\Omega(\mathbf{x})$ that yields more than two points. Analyzing the feature transform has proven useful in many applications of skeletons such as shape segmentation (Kustra et al., 2015; Feng et al., 2016).

The tuple formed by the distance transform and skeleton of a shape, denoted

$$MAT(\Omega) = (S_\Omega, DT_\Omega), \tag{2.4}$$

is called the Medial Axis Transform (MAT) of the shape Ω. More exactly, the MAT records all the points $\mathbf{x} \in S_\Omega$ together with their distance transform values $DT_\Omega(\mathbf{x})$. As we shall see later in Sec. 2.1.4, the shape Ω can be reconstructed from its MAT. Since the MAT can be obviously computed from Ω, one says that the MAT is a *dual* representation of the shape.

2.1.2 *Skeletonization techniques*

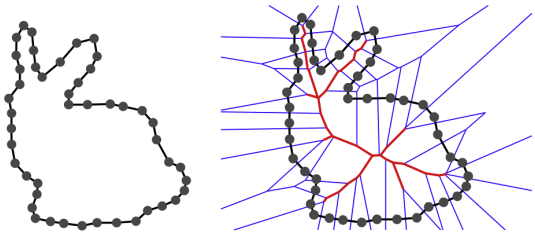
In Sec. 2.1.1, we described three alternative definitions of medial descriptors. More alternative definitions which lead to the same descriptor are presented in Tagliasacchi et al. (2016); Siddiqi and Pizer

(2008). We next overview the most prevalent classes of skeletonization techniques which use various forms of these alternative definitions to compute S_Ω.

Morphological thinning methods simulate the grassfire evolution (Fig. 2.1 (b) and Fig. 2.2 (b)) by iteratively eroding Ω inwards with constant speed until left with a one-pixel-thin connected structure representing S_Ω (Beucher, 1994; Peter and Breuß, 2013). While relatively straightforward to implement and fast, thinning methods generate different results due to different pixel removal orders. Thus, such methods do not in general guarantee that S_Ω is centered within Ω, *i.e.*, DT_Ω can be poorly approximated (Lam et al., 1992; Pudney, 1998). Another deficiency is that it is difficult to prune the skeleton (see next in Sec. 2.1.3) based on the feature points which are typically not stored during the thinning process.

Geometric methods (Ogniewicz and Kübler, 1995; Attali and Montanvert, 1997) find S_Ω as a subset of the edges of the *Voronoi diagram* of a piecewise-linear (polyline) representation of $\partial\Omega$, as illustrated in Fig. 2.3. Voronoi diagrams decompose Ω into cells based on *sites*, where each point in a cell is closer to that cell's site than to any other site. Sites are a dense discrete set of sampling points on $\partial\Omega$ (Fig. 2.3 (a)), *i.e.*, the endpoints of the segments forming the above mentioned polyline representation of the boundary. Sampling criteria play an important role in the quality of the obtained skeletons. The more sampling points on $\partial\Omega$, the more spurious branches the skeleton has, which need to be eliminated next. Conversely, the fewer sampling points are used on $\partial\Omega$, the worse can this sampling capture fine-scale details of the shape boundary. In practice, the sampling density is determined by the local feature size (Amenta and Bern, 1999; Zhu et al., 2014) to capture the boundary topology faithfully. The strongest point of geometric methods is that they fully work with a *vector* representation of both $\partial\Omega$ and S_Ω which, as outlined in Chapter 1, is advantageous for many reasons. Although also very accurate, compact, and connected in representation, geometric approaches are rather complex to implement, require a robust boundary discretization, and are computationally expensive (Telea and van Wijk, 2002).

Distance field methods compute DT_Ω (Eqn. 2.2) from $\partial\Omega$, and next find S_Ω along *singularities* of DT_Ω (Kimmel et al., 1995; Sethian, 1996; Telea and van Wijk, 2002; Falcão et al., 2004; Meijster et al., 2002; Hesselink and Roerdink, 2008). To start with, these methods need to compute DT_Ω. A well known technique for this is the Fast Marching Method (FMM) introduced by Sethian (Sethian, 1996) as an $\mathcal{O}(n\log(n))$ algorithm (for n pixels used to discretize Ω) to solve the Eikonal equation $|\nabla DT_\Omega| = 1$. Other, even faster, techniques compute the exact Euclidean



(a) (b)

Figure 2.3: Voronoi diagram based skeletonization method. (a) Sample the boundary of a 2D shape to create *sites*. (b) The Voronoi diagram of the boundary of sites, subsets of which (in red) can be approximated as skeletons. Image taken from Zhu et al. (2014).

distance transform in time linear in the pixel count n (Meijster et al., 2002).

However, the detection of the singularities of the DT_Ω field is non-trivial. Direct computation of singularities does not guarantee connected, one-pixel-thin skeletons (Niblack et al., 1990; Bouix and Siddiqi, 2000; Reinders et al., 2000), and is numerically unstable (Saha et al., 2016). As shown in Fig. 2.2 (c), perturbations along $\partial\Omega$ introduce many so-called spurious medial branches. To address this, Telea and van Wijk (2002) proposed the Augmented Fast Marching Method (AFMM), which is simple to implement, behaves robustly to boundary noise, and delivers connected skeletons. Key to this technique is tracking, for every point $\mathbf{x} \in \Omega$ visited by the FMM during the computation of DT_Ω, the 'span' of the boundary $\partial\Omega$ that is delimited by the feature points in $FT_\Omega(\mathbf{x})$. For non-skeletal points, this span is exactly one pixel. For skeletal points on branches that correspond to small-scale bumps on $\partial\Omega$, this span is small – equal to the arc-length of the boundary fragments corresponding to those bumps. Points located increasingly deeper along the skeleton have increasingly large spans. As such, the AFMM is a method for both skeleton *detection* and *regularization* – the latter is discussed in more detail in Sec. 2.1.3. The AFMM technique can be further accelerated on the GPU, yielding real-time skeletonization computation (Cao et al., 2010; Telea, 2014). Therefore, we next adopt this GPU-based distance-field-and-skeleton (thus MAT) computation approach for our work in this thesis.

2.1.3 *Regularization methods*

As stated in Sec. 2.1.2, medial axes S_Ω estimated directly from skeletonization approaches are notoriously unstable (Saha et al., 2016): Small perturbations along $\partial\Omega$, created *e.g.* by sampling inherent to both raster and vector representations, introduce spurious medial

branches (Fig. 2.2 (c)), which contribute little (or not at all in practice) to the description of Ω, but considerably complicate S_Ω. Effort has been invested in regularizing or simplifying medial axes, by removing (parts of) the spurious branches, to make them stable. However, a simplified skeleton $\tilde{S}_\Omega$ cannot *exactly* represent, or encode, Ω. Hence, accuracy (of representing a shape) and stability (of MAT computation) are related, but competing goals. We classify attempts to improve stability and accuracy into two groups, as follows.

Reconstruction-based methods approach the joint stability-accuracy problem by maximizing reconstruction accuracy, *i.e.*, the difference between Ω and $\tilde{\Omega}$ (Attali et al., 2009). This can be estimated using the Hausdorff distance (Rote, 1991) between (sampled representations) of $\partial\Omega$ and $\partial\tilde{\Omega}$, defined as

$$H(\Omega, \tilde{\Omega}) = \max\left\{h(\Omega, \tilde{\Omega}), h(\tilde{\Omega}, \Omega)\right\}, \tag{2.5}$$

where $h(A, B)$ is the one-sided Hausdorff distance given by

$$h(A, B) = \max_{\mathbf{a}\in\partial A}\left\{\min_{\mathbf{b}\in\partial B} \|\mathbf{a} - \mathbf{b}\|\right\}. \tag{2.6}$$

These methods (such as that of Zhu et al. (2014)) compute the simplified $\tilde{S}_\Omega$ by iteratively removing endpoints from S_Ω, continuously checking their reconstruction error (Eqn. 2.5) and stopping when this reaches a user-allowed level. While yielding accurate medial axes due to their explicit goal of optimizing for H, reconstruction-based methods are computationally expensive.

Medial-axis-based methods aim mainly to compute a stable, or regularized, $\tilde{S}_\Omega$ by removing spurious branches from S_Ω following criteria that only use the information present in the MAT. Arguably the most successful class of such criteria computes a so-called *importance* $\rho(\mathbf{x})$ of every medial point $\mathbf{x} \in S_\Omega$ as the boundary length between the feature points $\mathbf{f}_1$ and $\mathbf{f}_2$ of $\mathbf{x}$. This is precisely the length of the so-called boundary 'span' mentioned earlier in Sec. 2.1.2. Only medial points with $\rho(\mathbf{x})$ above a user-given threshold are taken over from S_Ω into $\tilde{S}_\Omega$. Importance thresholding is simple to implement for both raster (Falcão et al., 2004; Telea and van Wijk, 2002) and vector (Ogniewicz and Kübler, 1995; Attali and Montanvert, 1997) medial representations, delivers connected skeletons, and has an intuitive interpretation: The reconstructed shape $\tilde{\Omega}$ from $\tilde{S}_\Omega$ replaces all bumps along $\partial\Omega$ shorter than the threshold by circular arcs, effectively acting like a low-pass noise-boundary filter or multiscale representation of the shape and its skeleton.

However, the collapsed boundary length metric ρ has a key limitation: It treats boundary details of a given size (scale) similarly. In practice, this can be undesirable. Consider *e.g.* the rightmost noisy rectangle in Fig. 2.4. Using ρ to simplify this shape (and/or its skeleton, the

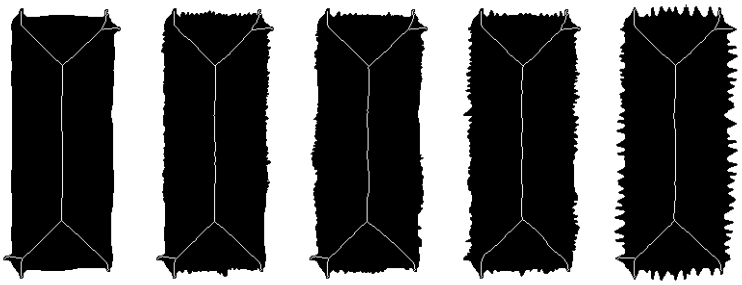


Figure 2.4: Skeleton regularization for five rectangular shapes with different amounts of noise added to the boundary using salience metric σ.

two being duals) will yield a shape where all small scale jaggies along the boundaries are eliminated, *including* rounding off the corners of the shape. Arguably, the latter is not desired. From a perceptual standpoint, the corners of the rectangle are more important, so we would like to have a way to obtain a shape with the jaggies along the edges removed but still having sharp corners, like the leftmost image in Fig. 2.4. For this goal, a modified version of the collapsed boundary length, called the *salience* metric (Telea, 2012) was proposed, as

$$\sigma(\mathbf{x}) = \rho(\mathbf{x})/DT_\Omega(\mathbf{x}). \tag{2.7}$$

Using this metric to regularize the skeleton, branches representing small-scale boundary bumps will be removed while branches that represent important (salient) corners are kept untouched. Figure 2.4 explores this insight by showing five rectangular shapes with randomly added noise of different scales on their boundary, and their simplified medial axes for $\sigma > 1.5$. We see that as the noise increases, the simplified medial axes change little, and are thus quite stable to noise.

Other skeleton-based regularization metrics include the angle between feature vectors (Attali and Montanvert, 1996; Foskey et al., 2003; Dey and Zhao, 2004; Hesselink and Roerdink, 2008), the divergence of the distance transform (Siddiqi et al., 2002), and higher order moments of the distance transform (Rumpf and Telea, 2002). Such metrics have proven very effective in producing simplified stable skeletons for 2D but also for 3D shapes (Tagliasacchi et al., 2016). However, they do not have the joint properties of multiscale representation/simplification, delivering a connected skeleton, and capturing directly an intuitive geometric property of the boundary that the collapsed boundary length ρ and its refinement, the saliency metric σ mentioned above, have. Moreover, the computation of ρ and σ do not add any overheads to their underlying skeletonization processes, as they directly use the distance and feature transform information that these processes need to compute. As such,

we next choose the salience metric σ (Telea, 2012) to simplify skeletons in our work.

2.1.4 *Shape reconstruction*

As stated in Chapter 1 and earlier in Sec. 2.1.1, the pair (S_Ω, DT_Ω), called the Medial Axis Transform (MAT), is a *dual* representation of shape: All information in a shape can be derived from its MAT and vice versa. Section 2.1.2 has illustrated three types of methods on how to extract the MAT from a shape. In turn, in this section, we discuss how to reconstruct a shape from its MAT.

Reverse fast marching method is exactly the reversed execution of the FMM depicted in Sec. 2.1.2. We evolve starting from each skeleton point $\mathbf{x} \in S_\Omega$ *outwards* until reaching its distance transform value $DT_\Omega(\mathbf{x})$, which is equivalent to solving the Eikonal equation $|\nabla (-DT_\Omega)| = 1$, and obtain the reconstructed shape $\tilde{\Omega}$ when $DT_\Omega = 0$. The advantages of this method are its computational efficiency ($\mathcal{O}(n \log n)$ for a shape Ω of n pixels) and the ability to stop the 'inflation' of the skeleton S_Ω towards Ω at any desired moment, based on local criteria. The latter can be important in applications where one wants to construct intermediate shapes between S_Ω and Ω, see *e.g.* Rumpf and Telea (2002).

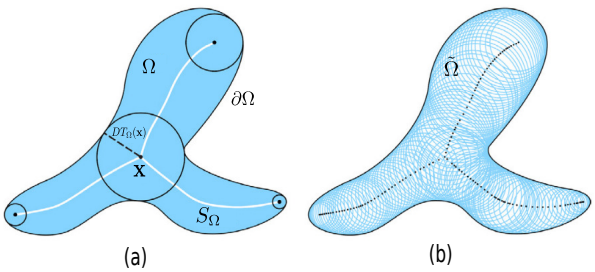


Figure 2.5: Illustration of the medial discs envelope method. (a) Medial axis transform of a 2D shape Ω. (b) The reconstructed shape $\tilde{\Omega}$ is approximated as the envelope of medial discs. Image adapted from Zhu et al. (2014).

Medial discs envelope methods reconstruct $\tilde{\Omega}$ as the union $\cup_{\mathbf{x}\in\tilde{S}_\Omega} B(\mathbf{x}, DT_\Omega(\mathbf{x}))$ of discs B centered at pixels $\mathbf{x}$ of the simplified skeleton $\tilde{S}_\Omega$ with skeletal-point radii given by the distance transform $DT_\Omega(\mathbf{x})$, as illustrated in Fig. 2.5. The disc B, called the *medial disc*, is the maximal inscribed disc in Ω as given by Blum's definition (Fig. 2.1 (a)). The envelope, or boundary, of the union-of-discs yields thus the reconstructed shape boundary $\partial\tilde{\Omega}$. Medial discs envelope methods can be efficiently

implemented on the GPU, see *e.g.* the detailed pseudocode in Meiburg (2011). As such, we next adopt this method to reconstruct a binary shape from its simplified skeleton.

2.2 IMAGE COMPRESSION METHODS

As our goal is to use medial descriptors to efficiently and effectively represent (encode) binary, grayscale, and color images; methods that aim for similar goals are of interest. We now briefly review the main classes of such methods.

Image compression is a well-studied field (Shum et al., 2003; Satone et al., 2017) which can be divided into two main classes: Lossless and lossy methods. Lossy compression has seen great interest due to its particularly high compression ratio (CR) while maintaining visual quality. In the past few decades, countless lossy compression approaches have been proposed. In the early days, transform domain coding dominated, such as the well-known discrete cosine transform (DCT) and related mechanisms used by JPEG (Wallace, 1992). DCT-based method divides the image into non-overlapping blocks for processing. When a high compression rate is desired, the results tend to show specific artifacts such as blocking or banding. To address this, the JPEG committee subsequently replaced the DCT's block-based algorithm with wavelet transform and proposed the efficient JPEG 2000 (Taubman and Marcellin, 2001), which not only yields better compression performance than JPEG, but has significant flexibility in the codestream.

In recent years, Deep Neural Network (DNN) methods have attracted increasing interest due to their high compression rate and good quality. Important methods in this area use Recurrent Neural Networks (RNNs) (Toderici et al., 2016, 2017; Johnston et al., 2018) and autoencoders (Choi et al., 2019; Theis et al., 2017). Generative Adversarial Network (GAN) methods (Agustsson et al., 2019; Mentzer et al., 2020) have also been developed recently. However, all such approaches expose issues with the distortion metric that was used to train the networks (Mentzer et al., 2020). They can react in hard to predict ways to unseen data (images that are far from the types present during training). Besides, DNN methods require significant training data and training computational effort.

Let us position our research with respect to the above existing developments. Our dense skeleton-based methods, described in Chapters 3, 4, 6, and 7, do not aim to compete with the compression rates of the above DNN techniques. However, our explicit 'feature engineering' approaches offer more control over how images are simplified during compression, are fast, and do not require training data. Separately, technique-wise, we show, for the first time, that medial descriptors, represented suitably by raster or vector models, are useful and usable tools for image compression.

2.3 IMAGE COMPRESSION QUALITY METRICS

Section 2.2 introduced a few main classes of methods for image compression. For all such methods, quality metrics are required to measure their performance. As stated earlier, lossy compression – our field of interest – has an inherent trade-off between how much we can compress a given image and how similar the compressed image will look to the raw, uncompressed, one. We thus distinguish two types of metrics to capture these two aspects, as follows.

First, we need to measure the *amount of compression* of an image. Let I be a raw, uncompressed, image and $\tilde{I}$ be its compressed version, by whichever method we want to assess. The amount of compression can be computed by the so-called compression ratio metric $CR(I, \tilde{I}) \in \mathbb{R}^+$ which measures the size of the binary representation of I divided by the size of the binary representation of $\tilde{I}$. In this model, the sizes of the images' binary representations are simply the amount of bytes used to store the respective images, and depend on how the images are represented. For example, for a raw (uncompressed) image I of $n \times m$ pixels, using b bytes-per-pixel, the storage size will be $\mathcal{O}(n \times m \times b)$. Storage sizes for the compressed representation $\tilde{I}$ strongly depend on what this representation actually is. Obviously, we want that $CR \gg 1$ for a good compression method.

Secondly, we need to measure how *close* the compressed image $\tilde{I}$ is to the original I, using so-called quality metrics $Q(I, \tilde{I}) \in \mathbb{R}^+$. 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 do not match well perceived visual quality (Wang and Bovik, 2009; Zhang et al., 2011, 2012). As visible in Fig. 2.6 (b), the JPEG compression of the original image (a), with a JPEG quality setting of 10%, is fuzzy, blocky, and has color quantization effects. Image (c) shows the result of one of our proposed compression methods, called SSDMD, discussed further in this thesis (Chapter 4). Although SSDMD's compression result also looks possibly a bit fuzzy, it is, we argue, perceptually closer to the original image than the JPEG compression result. However, the PSNR metric values, listed in the figure, tells the opposite. Thus, PSNR may not correspond well with perceived quality.

The structural similarity (SSIM) index (Wang et al., 2004) alleviates the above issue by measuring, pixel-wise, how similar two images – an image I and its reconstruction $\tilde{I}$, in our case – are by considering human perception, and is defined as

$$SSIM(I, \tilde{I}) = \frac{(2\mu_I\mu_{\tilde{I}} + C_1)(2\sigma_{I\tilde{I}} + C_2)}{(\mu_I^2 + \mu_{\tilde{I}}^2 + C_1)(\sigma_I^2 + \sigma_{\tilde{I}}^2 + C_2)}, \tag{2.8}$$

where μ_I and σ_I are the mean intensity and the standard deviation of I, respectively, and similarly for $\tilde{I}$. In the above, $\sigma_{I\tilde{I}}$ is the covariance between I and $\tilde{I}$. Also, $C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$, in which L is

(a) Original image (b) JPEG compression (10%) PSNR = 28.1 (c) SSDMD compression PSNR = 22.7

Figure 2.6: Example of PSNR tending not to match perceived visual quality well. (a) The original image. (b) JPEG compression with a quality of 10%. (c) SSDMD compression (introduced in Chapter 4).

the dynamic range of the pixel values (255 for 8-bit grayscale images) and $k_1 = 0.01$ and $k_2 = 0.03$ are typical defaults to evaluate Eqn. 2.8. SSIM was extended to three-component SSIM (3-SSIM) by using non-uniform weights for the SSIM map over three region types: edges, texture, and smooth areas (Li and Bovik, 2010). Further on, multiscale SSIM (MS-SSIM) (Wang et al., 2003) is an advanced top-down interpretation of how the human visual system interprets images that considers variations of image resolution and viewing conditions, and is defined as

$$MS\text{-}SSIM(I,\tilde{I}) = [\mathrm{SSIM}(I,\tilde{I})]^{\beta_M} \prod_{j=1}^{M-1} [c_j(I,\tilde{I})]^{\beta_j}, \tag{2.9}$$

where c_j is the contrast map $c(I,\tilde{I})$ iteratively downsampled by a factor of 2 on scale $1 \leq j \leq M$. $c(I,\tilde{I}) = (2\sigma_I\sigma_{\tilde{I}} + C_2)/(\sigma_I^2 + \sigma_{\tilde{I}}^2 + C_2)$. The exponent β_j models the relative importance of different scales. Comprehensive evaluations (Sheikh et al., 2006; Ponomarenko et al., 2009) have demonstrated that SSIM and MS-SSIM can offer statistically much better performance in assessing image quality than other quality metrics. Therefore, we next consider either SSIM or MS-SSIM in this thesis to compare a raw image I with its simplified, or compressed, representation $\tilde{I}$.

Having introduced related work on medial descriptors, image compression methods, and metrics to evaluate compression quality, we are now ready to present our first contribution in the area of dense medial descriptor compression.