

# 1 Introduction to wavelet density estimation

Wavelet density estimation was a promising technique for shape matching, as explored by Peter and Rangarajan in their 2008 paper [?]. Generally, the goal is to estimate a probability density function over the shape points and use that representation to compare shapes. The density is estimated by expansion into a wavelet basis form, which has a theory with strong mathematical foundations and provides several advantages over competing function bases (e.g. the Fourier basis, or simple histograms).

This wavelet density estimation framework can be applied for any dimensional data (e.g. 1D, 2D, and 3D shapes) by using the tensor product (multidimensional) forms of the 1D wavelet bases. The coefficients of this functional basis (of the probability density function) form our feature representation, and also conveniently map to the unit hypersphere due to constraints we impose during density estimation. This manifold geometry allows us to use a simple, efficient, and well-defined distance metric as our shape dissimilarity measure.

## 2 Wavelet density estimation

Our feature representation is formed by estimation of the probability density function of the shape, which we model as sample points from a distribution.

First, we must explain the use of wavelets as the foundation of our density estimation framework. An  $L^2$  function, also known as a square-integrable function, is a real- or complex-valued function whose square integrates to a finite number. Probability density functions fall under this function space, therefore allowing us to use wavelets for density estimation using a linear combination of these basis functions. In other words, wavelet density estimation as a paradigm is built on strong mathematical foundations.

The equation for 1D wavelet density estimation is

$$p(x) = \sum_{j_o, k} \alpha_{j_o, k} \phi_{j_o, k}(x) + \sum_{j \geq j_o, k} \beta_{j, k} \psi_{j, k}(x) \quad (1)$$

where  $\phi(x)$  and  $\psi(x)$  are the scaling and wavelet basis functions, also known as the “father” and “mother” wavelets, respectively. The functions can also be set to some starting resolution level that determines its dilation and contraction. To extend to multiple resolutions, we incorporate the wavelet basis function with a starting ( $j_o$ ) and stopping ( $j$ ) resolution level. In order to form a basis, we need to make multiple copies of the function over different translates,  $k$ . These functions are scaled by their scaling and wavelet basis function coefficients,  $\alpha_{j_o, k}$  and  $\beta_{j, k}$ . To easily retain properties of true densities, such as non-negativity and integration to one, we compute the square root of the probability density function

$$\sqrt{p(x)} = \sum_{j_o, k} \alpha_{j_o, k} \phi_{j_o, k}(x) + \sum_{j \geq j_o, k} \beta_{j, k} \psi_{j, k}(x) \quad (2)$$

Since we optimized for 2D wavelet density estimation, we consider the second dimension with  $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$  and  $\mathbf{k} = (k_1, k_2) \in \mathbb{Z}^2$ . We also extend the equation to

$$\sqrt{p(\mathbf{x})} = \sum_{j_o, \mathbf{k}} \alpha_{j_o, \mathbf{k}} \phi_{j_o, \mathbf{k}}(\mathbf{x}) + \sum_{j \geq j_o, \mathbf{k}} \sum_{w=1}^3 \beta_{j, \mathbf{k}}^w \psi_{j, \mathbf{k}}^w(\mathbf{x}) \quad (3)$$

where we calculate for the basis functions by using the tensor product method

$$\begin{aligned} \phi_{j_o, \mathbf{k}}(\mathbf{x}) &= 2^{j_o} \phi(2^{j_o} x_1 - k_1) \phi(2^{j_o} x_2 - k_2) \\ \psi_{j, \mathbf{k}}^1(\mathbf{x}) &= 2^j \phi(2^j x_1 - k_1) \psi(2^j x_2 - k_2) \\ \psi_{j, \mathbf{k}}^2(\mathbf{x}) &= 2^j \psi(2^j x_1 - k_1) \phi(2^j x_2 - k_2) \\ \psi_{j, \mathbf{k}}^3(\mathbf{x}) &= 2^j \psi(2^j x_1 - k_1) \psi(2^j x_2 - k_2) \end{aligned} \quad (4)$$

A mathematical advantage in using wavelets is its compact support, meaning they are nonzero on a small domain. Any point falling outside of the basis function’s support does not contribute to the basis function value at that translation.

Intuitively, the scaling and wavelet basis function coefficients determine the height of its basis function. These coefficients thus uniquely define the probability density function of a shape, which means we can use them as our

feature representation in shape retrieval. In order to initialize coefficients to a reasonable first approximation, we use a histogram approach. This is shown in the following equations:

$$\alpha_{j_o, \mathbf{k}} = \left( \frac{1}{N} \right) \sum_{i=1}^N \frac{\phi_{j_o, \mathbf{k}}(\mathbf{x}_i)}{\sqrt{p(\mathbf{x})}} \quad (5)$$

$$\beta_{j_o, k} = \left( \frac{1}{N} \right) \sum_{i=1}^N \frac{\psi_{j_o, k}(\mathbf{x}_i)}{\sqrt{p(\mathbf{x})}} \quad (6)$$

As we estimate the coefficients of our probability density function, we use a loss function to determine whether our coefficients are as close to optimal as possible. To do this we try to minimize a negative loglikelihood objective function with respect to the coefficients

$$-\log p(X; \{\alpha_{j_o, k}, \beta_{j_o, k}\}) = -\frac{1}{N} \sum_{i=1}^N \log \left[ \sum_{j_o, k} \alpha_{j_o, k} \phi_{j_o, k}(x_i) + \sum_{j \geq j_o, k} \beta_{j, k} \psi_{j, k}(x_i) \right]^2 \quad (7)$$

where  $X = \{x_i\}_{i=1}^N$ . To determine the direction of the gradient descent, we compute the following expression:

$$-2 \left( \frac{1}{N} \right) \sum \frac{\phi_{j_o, \mathbf{k}}(\mathbf{x})}{\sum \alpha_{j_o, \mathbf{k}} \phi_{j_o, \mathbf{k}}(\mathbf{x})} \quad (8)$$

In order for maintain essential properties of density estimation, we constrain the coefficients to one

$$\sum_{j_o, k} \alpha_{j_o, k}^2 + \sum_{j \geq j_o, k} \beta_{j, k}^2 = 1 \quad (9)$$

This constraint conveniently maps our feature representation into a unit hypersphere which allows us to take advantage of its geometric properties to signify similarity and dissimilarity between shapes in shape retrieval.