

# How Do Users Map Points Between Dissimilar Shapes?

Michael Hecher, Paul Guerrero, Peter Wonka, Michael Wimmer

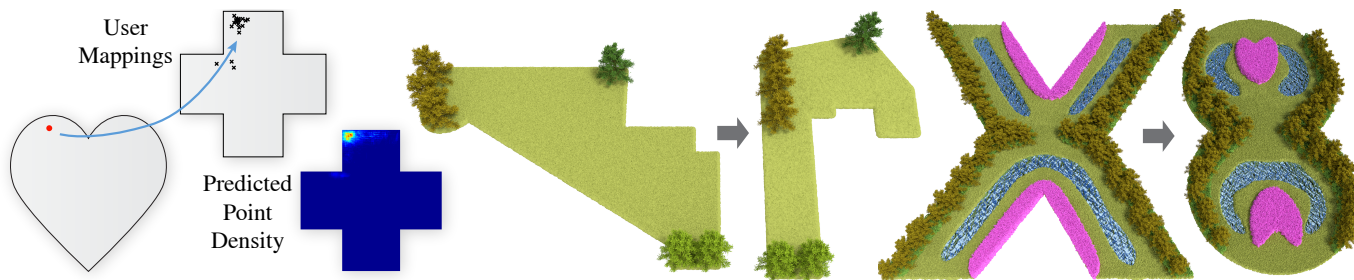


Fig. 1. We investigate how people map points between dissimilar shapes (left) and show that our findings can be used to define statistical models that allow us to transfer individual objects, like trees (middle), and/or procedural content, like ponds, forests and flowers (right), between dissimilar shapes.

**Abstract**— Finding similar points in globally or locally similar shapes has been studied extensively through the use of various point descriptors or shape-matching methods. However, little work exists on finding similar points in dissimilar shapes. In this paper, we present the results of a study where users were given two dissimilar two-dimensional shapes and asked to map a given point in the first shape to the point in the second shape they consider most similar. We find that user mappings in this study correlate strongly with simple geometric relationships between points and shapes. To predict the probability distribution of user mappings between any pair of simple two-dimensional shapes, two distinct statistical models are defined using these relationships. We perform a thorough validation of the accuracy of these predictions and compare our models qualitatively and quantitatively to well-known shape-matching methods. Using our predictive models, we propose an approach to map objects or procedural content between different shapes in different design scenarios.

**Index Terms**—2D mappings, Shape matching, Shape similarity, Transformations, User studies.

## 1 Introduction

In this work, we aim at characterizing the geometrical properties of intuitively similar points in dissimilar two-dimensional shapes. More precisely, our goal is to find a correlation between the position of similar points picked by humans on the one hand, and simple *geometric relationships* between these positions and the containing shapes on the other hand. Such relationships include the distance to the closest shape boundary, or the local coordinates of the shape normalized to the bounding box, for instance.

Finding similar points in globally or locally *similar* shapes has been studied extensively through the use of various point descriptors or shape matching methods. However, little work exists on finding similar points in *dissimilar* shapes. While this problem does not always have a clear solution, as illustrated in Figure 2, most humans have an intuition as to which points are more similar than others. Studying this type of similarity can guide point-matching methods needed for applications like texture transfer, the propagation of edit operations between

dissimilar shapes, or learning scene layouts by example.

In this paper, we describe a study where users were given two dissimilar two-dimensional shapes and asked to map a given point in the first shape to the point in the second shape they consider most similar. Overall, the study consists of 2,000 such triples each made up of a point and a shape pair, with a total of 40,000 individual trials generated by users. The resulting point distributions can be used to study the correlation between user-placed points and their geometric relationships to a shape (Sections 4.1) and to develop new methods to map points between dissimilar shapes (Section 4.2) by training statistical models that perform better than well-known methods (Section 5.2). Furthermore, we develop algorithms that exploit the higher accuracy of our models to map content between different shapes (Section 6). An example is given in Figure 1.

In our analysis of the point distributions, we found that users agree strongly in their point placements, even though the shapes are dissimilar. Our experiments also suggest that user mappings can be predicted by simple statistical models, and that models which handle interactions between relationships (e.g., point is close to a corner and far away from the centroid) perform much better than those that do not.

This work makes the following main contributions:

- A dataset with a total of 2,000 distinct point mappings between all pairs of 28 two-dimensional shapes, performed by 20 different users each.
- Results showing the correlation of similar points picked by users and simple geometric relationships between the points and the containing shape.
- An analysis of the predictive performance of current well-known methods.

- Michael Hecher is with TU Wien. [hecher@cg.tuwien.ac.at](mailto:hecher@cg.tuwien.ac.at).
- Paul Guerrero is with UCL. [paul.guerrero@ucl.ac.uk](mailto:paul.guerrero@ucl.ac.uk).
- Peter Wonka is with KAUST. [peter.wonka@kaust.edu.sa](mailto:peter.wonka@kaust.edu.sa).
- Michael Wimmer is with TU Wien. [wimmer@cg.tuwien.ac.at](mailto:wimmer@cg.tuwien.ac.at).

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x.

For information on obtaining reprints of this article, please send e-mail to: [reprints@ieee.org](mailto:reprints@ieee.org).

Digital Object Identifier: [xx.xxxx/TVCG.201x.xxxxxx/](https://doi.org/10.1109/TVCG.201x.xxxxxx/)

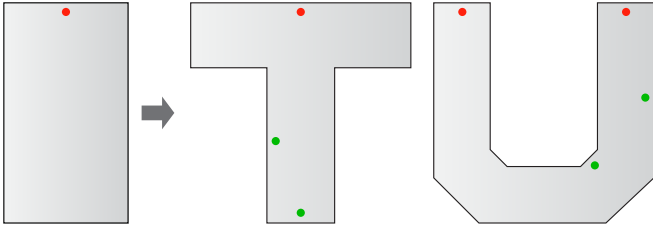


Fig. 2. When mapping the red point on the left to the two dissimilar shapes on the right, there may be no clear best placement, illustrated by multiple red points in the rightmost shape. However, some points (red) are intuitively more similar than others (green).

- Two distinct statistical models to predict the probability of a user picking a particular point in a shape.
- An approach based on these statistical models to map objects and procedural content between shapes.

Overall, our study provides an empirical foundation for using simple geometric relationships as basic building blocks for mapping points between dissimilar shapes.

## 2 Related Work

**Shape matching and mapping** Several types of generalized coordinates for arbitrary planar polygons have been presented that can be used to map a point in one shape to a point with the same coordinates in a second shape. Mean-value coordinates [8, 16] are a generalization of barycentric coordinates that retain a number of desirable properties such as smoothness and invariance to similarity transformations. Harmonic coordinates [17] are always positive, even for concave polygons, but less simple and efficient to compute. Green coordinates [21] relax the constraint that points remain inside the polygon after the mapping in favor of a conformal or quasi-conformal mapping. However, all these coordinates require that the polygons have the same number of vertices, which is not the case in our shapes. Additionally, they map a point in the first shape to a single point in the second shape, which makes them less useful for predicting the response of users, which is often multimodal, as we will see later.

Another approach to map points between shapes is through partial shape matching, where transformations are found that match the local geometry surrounding the given point to some part of the second shape. These transformations can then be applied to the given point to find similar points. Several methods have been presented, ranging from simple point descriptors such as Shape Contexts [1] to more complex approaches based on minimizing some shape deformation induced by the transformation [11, 22, 26]. However, all of these approaches are based on finding local similarities of the shapes, which are not present in our case.

Guerrero et al. [12] presented a method to find similar points in dissimilar shapes based on simple geometric relationships between the points and the shapes. We take the geometric relationships introduced in that work and study their correlation to similar points in dissimilar shapes picked by users.

**Psychology** The perception of similarity has a long history in various fields of psychology, psychophysics and neuroscience, for instance, to investigate the effect of high-level (object parts) and low-level (motion, texture, or luminance contrasts) visual cues, as well as visual cues in the perception and recognition process [10, 19]. Further research has focused on the ability to recognize objects over various viewing conditions [2, 24, 28] or the recognition of specific shapes like human body parts [14, 23].

In this work we focus on the similarity of points in dissimilar simple 2D shapes and examine if geometric relationships between these points and their containing shapes can be used to model this similarity.

**Methodology** Several recent publications investigated similar research questions through user studies [5, 18]. Our methodology and evaluation is mostly inspired by the works of Chen et al. [3], Cole et al. [4] and McCrae et al. [25]. Cole et al. performed a study to investigate where artists draw lines to convey the shape of a 3D object through a 2D line drawing and trained a statistical model to predict these drawings. A user study was also used by McCrae et al. [25] to investigate how 3D shapes can be best represented by planar slices that match user preferences. This is achieved by fitting the parameters of the presented algorithm based on user data. A related problem was addressed by Chen et al. [3]. The aim of their work was to evaluate which points are most salient to users in 3D shapes and how to predict them by an analytic model.

The high-level approach of these works is quite similar in that statistical models are trained based on user data to predict user responses. Our methodology is also based on this principal setup, but we additionally compare two distinct models in their capabilities to predict user mappings in simple 2D shapes.

We use our models in different *example-based* design scenarios like placing a certain amount of objects at the most probable positions or placing objects only if the probability is high enough. The algorithms presented in this paper are therefore closely related to similar works like [6, 20, 27], but our approach is driven by the data of our user study, and opens up new fields of application.

## 3 Study Design

The main goal of this work is to understand how people map points between dissimilar shapes. In particular, we want to investigate which geometric factors influence a user’s decision when mapping a single point (*query point*) from within one shape (*source shape*) to another shape (*target shape*). To this end, design decisions have to be made with respect to the shapes used in the study (Section 3.1), placement of query points within those shapes (Section 3.2), and the way the data is collected (Section 3.3).

### 3.1 Shapes

There are several geometric details of shapes that can potentially guide a user’s mapping of a query point from one shape to another, like corners, edges, or the whole boundary of source and/or target shape. Semantics of a shape, for example when mapping points between two face-like shapes, might also be a critical factor influencing a user’s decision, but in this work we focus solely on the analysis of geometric factors. We also do not make any assumptions about how symmetries may affect the results (e.g., we do not prune points that are at symmetric locations in a shape), since we cannot know which symmetries the users would actually take into account (point symmetries, rotational symmetry, etc.). Therefore, we need to select a reasonable subset of shapes that can cover many possible configurations of geometric details and that also allows the user to focus on clearly recognizable geometric details. We hence make the following design decisions with regard to the shapes used in this study (see also Figure 3):

- **Simplicity:** To ensure that users are not overwhelmed by the amount of geometric detail and are not tempted to abstract fine details, shapes have to be kept relatively simple.
- **Coverage:** All shapes should be easily distinguishable by users and should contain different combinations and numbers of geometric details. This means that no shape can be the product of simple transformations from any other shape.

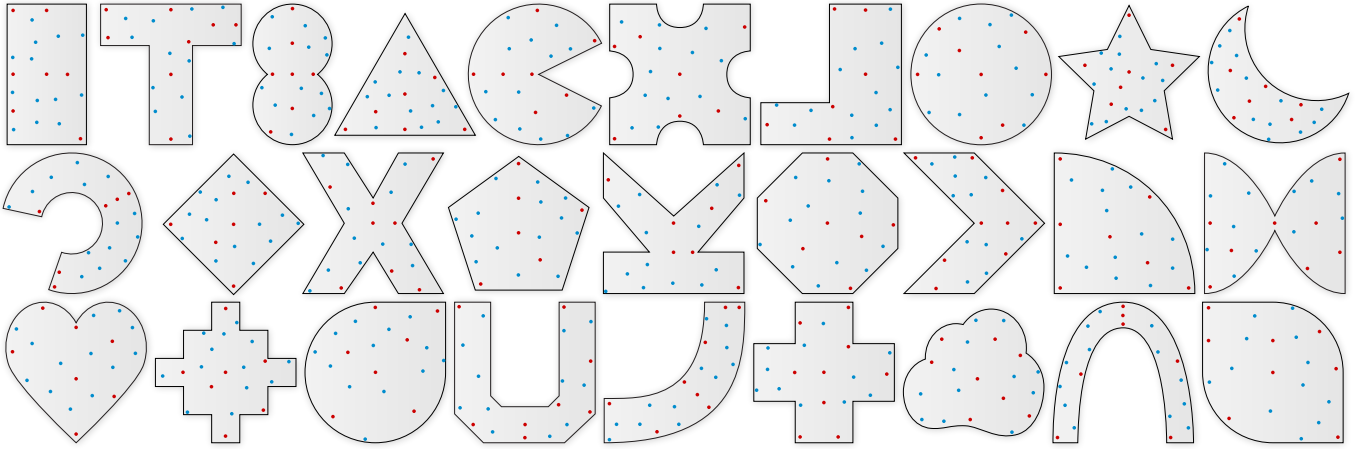


Fig. 3. The 28 shapes and the query points used in the study. Manually placed query points are shown in red, randomly placed points in blue.

- Semantics:** As mentioned before, analyzing the effect of semantics is not the aim of this work. Hence we prefer abstract over semantically meaningful shapes to ensure that users base their decisions primarily on geometric properties. For the same reason, we avoid colors and textures in our shapes. Although we focus on geometry only, applications using our method could add semantics in a separate step.

In total, 28 shapes were used in our study. To bring all shapes into a common size range, we normalize them based on their bounding box diagonal. Although we selected simple shapes, we were able to apply our study results to more complex shapes to facilitate the reuse of assets in content creation. Example applications can be found in Section 6.

### 3.2 Query Points

Another important design decision concerns the placement of query points within shapes. Each shape can have several places of interest (near corners, near the centroid, on symmetry axes, exactly between two opposite boundaries, at a certain distance to a feature) where we would expect query points to result in more consistent user mappings compared to randomly placed query points. In a pure random sampling strategy, we would require too many sample points to ensure that these places are adequately sampled. Therefore, we place query points in two steps. In the first step, we place them manually at the places of interest mentioned above, and in the second step, we place them randomly with a uniform probability inside the shape. Because this random placement can lead to an inefficient sampling of a shape, we use a Poisson-disk like sampling strategy: we introduce a minimal distance condition for a newly placed random point to all previously added points, which is 0.08 times the diagonal of the bounding box of the shape. In case this condition prevents the random placement to converge (the shape cannot contain the predefined number of query points), the minimal distance is reduced by a factor of 0.9 and the random placement process is repeated until a valid result can be found. We choose to place a total of 20 query points per shape (see Figure 3), with 7.32 placed manually and 12.68 randomly on average ( $\sigma = 0.77$ ).

### 3.3 Data Collection

The study was carried out online via Amazon Mechanical Turk. The source and target shape were shown side by side with a random query point shown in the source shape as discussed above. Participants were asked to mark one position inside the target shape they consider the most similar to the query point in the source shape.

Using all  $28 \times 27$  possible shape pairs, each with 20 query points, was not feasible (given a reasonable budget), as we would get too few participant trials per query point and shape pair for meaningful results. Instead, we decided to randomly pick from all possible 15,120 combinations a subset of 2,000 (583 containing a manually placed query point). Reduced variance and a more even sampling of our space of possible combinations was obtained by using a quasi-random Halton sequence [13] when choosing this subset. For each shape-pair combination, an average of 2.55 query points were picked ( $\sigma = 1.04$ ).

In total, 670 anonymous participants completed 40,000 trials with a median number of 20 finished trials per participant. The 20% of the most active participants were accountable for 74% of the gathered data.

### 3.4 How much do participants agree?

To understand whether the study provides a consistent basis for further analysis, we examine how much participants agree in their mappings. As a basis for our measure, we compute the average pair-wise Euclidean distance between the points of a query point mapping and normalize this value by using a generalized version of Fleiss' Kappa [7]:

$$\kappa_c = \frac{a - a_{rand}}{a_{max} - a_{rand}}, \quad (1)$$

where  $a$  is the participant agreement,  $a_{max}$  is the maximum attainable agreement, and  $a_{rand}$  is the agreement of points placed randomly into the target shape. It measures the *agreement above chance* relative to the *maximum attainable agreement above chance*. Thus, with our definition of agreement, we get:

$$\kappa = \frac{\bar{D} - \bar{D}_{rand}}{-\bar{D}_{rand}}, \quad (2)$$

where  $\bar{D}$  is the average pair-wise distance in the set of points placed by the participants inside a target shape, and  $\bar{D}_{rand}$  the average pair-wise distance of the same number of points placed randomly inside the target shape. To reduce variance, we compute  $\bar{D}_{rand}$  for several sets of random points (20 in our evaluations) and take the average. Intuitively, the maximum agreement of  $\kappa = 1$  is reached if all participants placed the point at the exact same position,  $\kappa = 0$  is chance agreement and  $\kappa < 0$  is systematic disagreement (more disagreement than can be expected by chance). Note that multiple modes have a higher  $\kappa$  than a single mode, but still a lower  $\kappa$  than a random placement, since the distances between the points in each mode are low.

The distribution of  $\kappa$  among participants in our dataset is depicted on the top of Figure 4. Agreement is 0.61 on average and well above 0 for most queries. Examples for several values

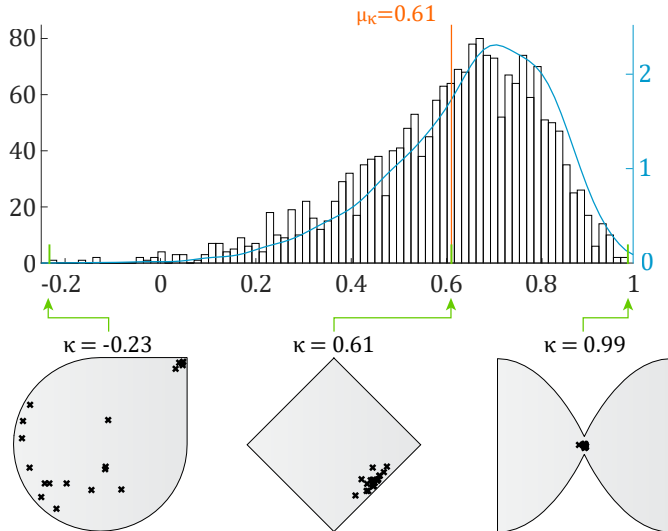


Fig. 4. The distribution of agreements over all queries, where 0 indicates random and 1 perfect agreement. Bars show the histogram and the blue curve the estimated distribution function. Below, three samples for poor ( $\kappa = -0.23$ ), substantial ( $\kappa = 0.61$ ) and almost perfect agreement ( $\kappa = 0.99$ ) are shown. On average, participants have a substantial agreement of  $\mu_{\kappa} = 0.61$ .

of  $\kappa$  are shown at the bottom of Figure 4. This high agreement suggests that our study forms a solid basis for creating predictive models for such point placements.

#### 4 Towards a Statistical Model for Mapping Points between Dissimilar Shapes

We suspect that the high agreement between the point placements of participants (see Section 3.4) is the result of similarities they found in the containing shapes. To quantify these similarities, we use the geometric relationships of a point to different parts of the shape. In this section we will first investigate which of these relationships can predict user mappings, while the second part focuses on designing statistical models that can realize their predictive power.

##### 4.1 Can geometric relationships predict mappings?

We first turn to the question whether there is a connection between the position of points picked by participants and simple geometric relationships between the points and the containing shapes. More specifically, we conjecture that these relationships can be used to predict participant mappings, which is critical if we want to create models based on them. A necessary condition for this hypothesis to be true is that there is some form of correlation between *source relationships*, i.e., the relationships of the query point with the source shape, and *target relationships*, which are the relationships of the mapped point to the target shape. In this section, we perform a correlation analysis on the data set by testing for linear correlation between source and target relationships.

We focus on 17 simple geometric relationships (R1-R17) between a point and its containing shape, inspired by the work of Guerrero et al. [12]. Six relationships are based on a global coordinate frame of the shape:

- **R1-R2** The *Bounding Box Normalized X-* and *Y* coordinates use the lower left corner of the shape’s bounding box as origin, and are normalized to the width and height of the bounding box.
- **R3-R4** The *Bounding Box Center X-* and *Y* coordinates use the center of the shape’s bounding box as origin.

- **R5-R6** The *Centroid X-* and *Y* coordinates use the centroid of the shape as origin.

Two relationships are defined between the point and the centroid of the shape:

- **R7** The *Centroid Distance* measures the Euclidean distance between point and centroid.
- **R8** The *Centroid Direction* measures the direction from point to centroid as the angle in  $[-\pi, \pi]$  comprised with the positive *x*-axis.

Two relationships are based on local properties of the shape boundary:

- **R9** The *Boundary Distance* relationship measures the shortest Euclidean distance between point and boundary of the containing shape.
- **R10** The *Boundary Direction* relationship quantifies the average normal direction of the shape boundary close to the point. The average is weighted by the distance of the boundary to the query point. More specifically, a Gaussian with fixed width centered at the point. The relationship is implemented as a line integral of the weighted normal over the shape boundary, which has a closed-form solution. The average direction is quantified as an angle in  $[-\pi, \pi]$  comprised with the positive *x*-axis.

Edges of the shape are segments of the shape boundary between corners (if no corners are present, neither are edges). Four relationships are defined between a point and an edge of the shape:

- **R11** The *Edge Distance* measures the shortest Euclidean distance between the point and the edge.
- **R12** The *Edge Arc Length* is the arc length from the start of the edge to the position on the edge that is closest to the point inside the shape. Start and end of an edge are defined clockwise on the boundary of the shape.
- **R13** The *Edge Relative Arc Length* is the Edge Arc Length normalized by the total arc length of the edge, i.e., the arc length from start to end.
- **R14** The *Edge Symmetric Arc Length* is defined as the minimum of the Edge Arc Length and the ‘reverse’ arc length, starting at the end of the edge and running towards the start.

Three relationships are defined between a point and a corner of the shape, where corners are defined as  $C_1$  discontinuities of the boundary (i.e., direction discontinuities):

- **R15** The *Corner Distance* is the Euclidean distance between point and corner.
- **R16** The *Corner Direction* measures the direction from point to corner as the angle in  $[-\pi, \pi]$  comprised with the positive *x*-axis.
- **R17** The *Corner Relative Direction* measures the angle comprised by the direction from corner to point and the start of the first boundary segment adjacent to the corner. This angle is normalized with the opening angle of the corner.

This is not an exhaustive list of all geometric relationships that could be found between a point and a shape, but we will show in the following sections that this set is sufficient to successfully characterize participant mappings.



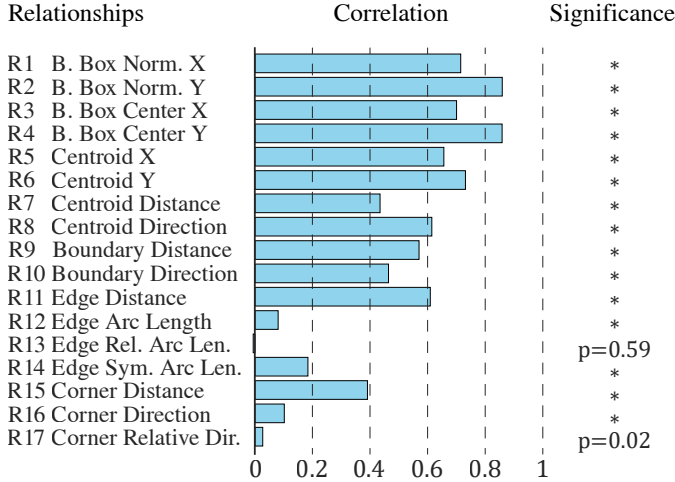


Fig. 5. Correlation between query and participant point relationships. Bars marked with a star are statistically highly significant ( $p < 0.001$ ), otherwise the corresponding  $p$ -values are shown. Of the 17 tested relationships, 15 have a non-negligible correlation.

In order to quantify the linear correlation between source and target relationships, we use Pearson’s linear correlation coefficient [9]. Relationships to corners and edges may result in multiple values for a point in a shape. In this case, we only use the value corresponding to the corner or edge that is closest to the point.

Figure 5 shows the results of the correlation analysis. The six relationships based on the global coordinate frame of the shape exhibit the strongest correlation, followed by the local centroid and boundary relationships. For relationships based on corners and edges, only the distance shows high correlation. All correlations marked with a “\*” have a very high significance level of  $p < 0.001$ . These results suggest that out of the 17 tested relationships, 15 may be useful for predicting user mappings, especially those based on the global coordinate frame, shape boundary and edge and corner distance.

In comparison, edge and corner relations have lower correlation scores. To examine whether these relations are still important in some cases (e.g., close to a corner), we computed a range-restricted Pearson’s correlation coefficient by sliding a window of a fixed size over the value domain for each relationship. As such a window could lead to regions with very few points, we additionally constrain our evaluation to regions where at least 100 query points were mapped by users. For our evaluation we chose a window size of 0.1 times the distance between minimum and maximum relationship value, as this resulted in relatively smooth results. Note that a range-restricted correlation usually results in lower correlation coefficients compared to the global evaluation, because the local inconsistency of user responses has a higher “negative” impact.

For the purpose of illustration, we only show this evaluation for a subset of relationships in Figure 6 (the complete set can be found in the supplementary material). We observe that the *Centroid* R8 and *Boundary Direction* R10 show relatively high correlation coefficients over all ranges. This suggests that they are almost equally important no matter in which direction a query point is placed. In contrast, some relationships, like the *Edge Arc Length* R12 or *Corner Direction* R16, do not correlate strongly in any of the specified ranges, which indicates that they are not as useful to predict user mappings, although they might be helpful overall (significant correlation in Figure 5). We also observe that *Bounding Box Center X/Y* R3-R4, *Centroid X/Y* R5-R6, *Edge Distance* R11 and *Corner Distance* R15 have a noticeably higher correlation closer to the corresponding shape part (i.e., edge, corner, centroid and bounding

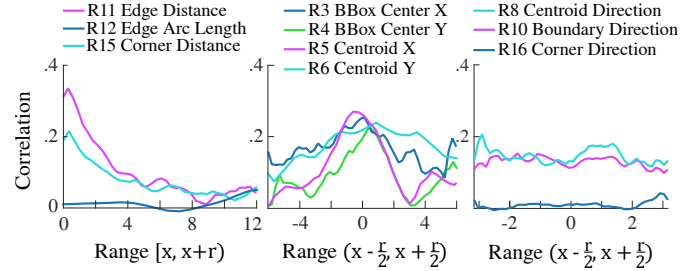


Fig. 6. Range restricted correlation of selected source and target shape relationships with respect to the relationship value of query points. The size of the range  $r$  is a tenth of the difference between the maximum and minimum relationship value.

box center).

These results indicate that even low-correlation relationships could improve the predictive performance of a statistical model in some cases (i.e., when the query point is close to the corresponding shape part). However, this does not hold true for the *Edge Relative Arc Length* R13 and *Corner Relative Direction* R17, which perform poorly overall and in range-restricted scenarios. We hence excluded them from our model design.

## 4.2 Designing Statistical Prediction Models

In the previous section, we have seen that there is a strong correlation between relationship values and user mappings. In this section, we present two separate models to predict the likelihood of point mappings based on these relationships. More specifically, we describe the similarity between the geometric relationships of points in the source (*source points*) and target shape (*target points*) as features. Our models will use this similarity to predict a likelihood for any source-target point mapping to be done by a user (Section 4.2.1). To allow our models to predict such likelihoods, we first need to train them on ground-truth likelihoods, which we approximate by fitting a Gaussian Mixture Model to the user-study data (Section 4.2.2). Based on this, we propose a linear model (Section 4.2.3) and a random-forest model (Section 4.2.4), which allow us to predict the likelihood of point mappings between dissimilar shapes.

### 4.2.1 Relationship Similarity Features

Because the training of our statistical models cannot be done directly on the relationships of query and target points, we define the *similarity* between a query and target point<sup>1</sup> as a Gaussian of the difference between relationship values:

$$s(q, t) = \mathcal{G}(q - t, 0, \sigma^2), \quad (3)$$

where  $q$  is the query relationship value and  $t$  the target relationship value. We choose sigma to reflect a maximum accuracy of point placements made by users. In all our experiments, we assume this maximum accuracy to be 0.016 of the normalized bounding-box diagonal of the target shape. This accuracy in the spatial domain is then transformed to an accuracy of relationship values for each individual relationship. Note that using this formulation of similarity is a design choice, and other formulations may lead to better predictions. However, we show in Section 5.2 that this formulation achieves significantly improved predictions over well-known methods for shape matching.

The shapes in our data set have different numbers of corners and edges. Consequently, for a relationship type such as the corner distance, we do not have a single value  $q$  or  $t$ , but rather a set of values  $Q$  or  $T$ , one for each corner. To employ standard machine-learning techniques, we have to use the same number of relationships to describe each point placed by a user. We

<sup>1</sup>Note that the term *target point* not only refers to points placed by users, but to any point inside the target shape.

pick the value corresponding to the closest feature (e.g., the corner closest to the query point) from each set to get an equal number of features, leaving us with one similarity value for each relationship type, which can be arranged into a 15-element *feature vector*  $\mathbf{f}$  for any query-target point pair:

$$\mathbf{f} = \left( s(q^{\text{bd}}, t^{\text{bd}}), \dots, S(Q^{\text{ed}}, T^{\text{ed}}) \right)^{\text{T}}, \quad (4)$$

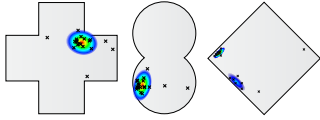
where  $q^{\text{bd}}$  and  $t^{\text{bd}}$  refer to the boundary distance relationship values in the source and target shape respectively, and  $Q^{\text{ed}}$  and  $T^{\text{ed}}$  are the set of query and target edge distance relationship values from which  $S$  picks the closest features of both sets and computes their similarity  $s$ .

#### 4.2.2 Ground-Truth Likelihood of Point Placements

Besides the feature vector  $\mathbf{f}$  for a query-target point pair, we need to define a likelihood  $l$  to describe the mapping from source to target shape statistically and to allow us to train our statistical models. In other words, we need to find a function that returns the likelihoods of user mappings

$$L(\mathbf{f}) = l. \quad (5)$$

Assuming that when users map query points to a target shape the resulting target points will form one or more clusters that follow a normal distribution, the underlying distribution of user points can be represented by a mixture of Gaussians. In order to find the best-fitting mixture, we iteratively increase the number of Gaussians fitted to the data, beginning with one Gaussian. Additionally we require that a single point cannot form a cluster.



We then select the Gaussian mixture distribution (GMD) with the lowest Bayesian information criterion (BIC) and where all Gaussian distributions lie inside the

shape to represent the user point distribution (see examples above). We can then obtain the likelihood  $l$  of a query-target point mapping with a feature vector  $\mathbf{f}$ , by evaluating the GMD at the target point position

$$L(\mathbf{f}) = \text{GMD}(\mathbf{t}^{xy}) = l, \quad (6)$$

where  $\mathbf{t}^{xy}$  can be any of the relationships of  $\mathbf{f}$  based on a Cartesian coordinate frame, as long as the GMD was fitted in the same coordinates (e.g., *Bounding Box Normalized X- and Y*).

#### 4.2.3 Linear Model

The first statistical model we introduce is based on a linear combination of the 15 relationship types. More specifically, the model  $M_{\text{LM}}$  is parametrized by a vector of linear coefficients  $\omega$ :

$$M_{\text{LM}}(\mathbf{f}_i, \omega) = \mathbf{f}_i^{\text{T}} \omega = l_i, \quad (7)$$

where  $\mathbf{f}_i$  is a feature vector and  $l_i$  is the predicted likelihood for a query-target point combination  $i$ , as defined in Section 4.2.2 ( $l_i = L(\mathbf{f}_i)$ ). We fit this linear model to our user-study data by solving the linear least squares problem:

$$\mathbf{F}\omega = \mathbf{l} \quad (\omega_j \geq 0), \quad (8)$$

where  $\mathbf{F}$  is a matrix of size  $N \times 15$  (which we call the *user feature matrix*), with rows composed of  $N$  feature vectors  $\mathbf{f}_i$  as defined in Section 4.2.2. We constrain the coefficients  $\omega$  to be larger than zero, since all non-negligible correlation coefficients found in Section 4.1 are positive. To obtain the feature vectors  $\mathbf{f}_i$  and predicted likelihoods  $l_i$  for the entire target shape for a given query point (and not only in the typically high-likelihood areas covered by the user-placed points), we cover the target shape with a regular grid of points to obtain the feature vectors for

those points that lie inside the target shape. The corresponding placement likelihoods are obtained from the GMD defined in Section 4.2.2. A grid spacing of 0.01 times the maximum of the width and height of the shape is used; tests have shown that using a denser grid has no relevant impact on the results for our set of shapes. The feature vectors and likelihoods of the grids for all query points in the user study are assembled into the  $N$  rows of the user feature matrix  $\mathbf{F}$  and the vector of likelihoods  $\mathbf{l}$ , giving us over  $N \approx 7.4$  million data points.

#### 4.2.4 Random-Forest Model

As a second model we use *random forests* [15], similar to [4], which are generated by training individual regression trees on multiple subsets of the data and merging them into a single tree. Multiple trees are used to reduce overfitting. We use the same user feature matrix  $\mathbf{F}$  and likelihood vector  $\mathbf{l}$  described in the previous section as training data.

Similar to the linear model, the random-forest model  $M_{\text{RF}}$  takes a feature vector as an input and predicts the corresponding likelihood.

$$M_{\text{RF}}(\mathbf{f}_i) = l_i, \quad (9)$$

Unlike the linear model, dependencies between individual features can be modeled through the tree-like structure, as predictions of subtrees of a node can be based on completely different rules. This makes the random-forest model more powerful, but also computationally more expensive.

To optimize the performance, we selected the training parameters in a way that combines a low *out-of-bag error* (predictive performance) with a short training time. For our dataset this resulted in using 15 subsets for training the individual regression trees with a minimal leave size of 100 observations.

## 5 Evaluation

### 5.1 Analysis of Predictive Power

To answer the question whether the linear and random forest models can be used to predict user mappings, we propose two quantitative performance measures that compare the results of query point mappings produced by models to those of users. In the following we will discuss how to define them in case of a single query point mapping:

1. The first measure is intended to evaluate how well a model can estimate the *distribution of user points* inside the target shape for a given query point. Because none of the models included in the evaluation can produce continuous distributions, we densely sample the target shape and compute the mean squared error (MSE) of the predicted likelihoods to the likelihoods and the *reference distribution* (the Gaussian mixture distribution of user responses from Section 4.2.2).
2. With the second measure we want to evaluate how “close” the predicted *most likely user point* is to the actual most likely user point in the reference distribution. Note that measuring the Euclidean distance between these points is not suitable for our purpose, as the quality of the prediction also depends on the density of users placements. The denser the user placements are, the closer we want our predictions to be to the *actual most likely user points*. In other words, we would like to measure a distribution. Fortunately, the likelihoods we obtain from our reference distributions in Section 4.2.2 already fulfill this condition. Hence, we use the ratio between likelihoods of the predicted *most likely user mapping* and the *actual most likely user mapping* as a performance measure and refer to it as the likelihood ratio (LHR). It lies between 0, indicating bad predictions, and 1, perfect prediction of the *most likely user mapping*.

To obtain these measures for a set of query point mappings, we compute the mean of the values.

	Models	a) LHR	b) MSE
baseline	Random Mapping	0.02	–
	Global Mapping	0.20	–
existing	Inner Shape Cont.	0.02	$3.93 \times 10^{-7}$
	B-Spline Shape Cont.	0.17	–
proposed	Linear Model	0.22	$1.29 \times 10^{-7}$
	Random Forest	0.38	$0.41 \times 10^{-7}$
reference	Participants	0.38	0

Table 1. A comparison of a) average likelihood ratios (LHR) and b) average mean squared error of likelihoods (MSE) after cross validation for all approaches discussed in this work (LHR: between 0 and 1, higher is better; MSE: lower is better).

Furthermore, cross validation is used to estimate how well our models generalize. To this end, we divide the data into 30 groups based on shape pairs. For each of these groups, we train the models on the remaining data (the 29 other groups) and perform predictions on the selected group. The final result is the average over all groups (see *proposed* models in Table 1). We compare them to two baseline methods and the reference result based on user responses (see *baseline* and *reference* in Table 1 respectively). The first baseline method, “random mapping”, places points randomly inside the target shape. In “global mapping”, the query point is placed at the same normalized bounding box position in the target shape (as if using only the *Bounding Box Normalized* relationship). Should this point lie outside the target shape, the next closest point inside the shape is used. As reference for the performance of these approaches, we compare to our Gaussian Mixture distribution estimated from the actual user data (Table 1, last row). For the LHR score, we compute the average LHR of all user points. Note that the reference achieves a score below the maximum of 1 because there is disagreement between users – only perfect agreement would give the maximum score.

Compared to the baseline methods, the linear model does not significantly improve the prediction of the most likely user mapping when compared to global mapping (0.22 vs. 0.2) and reaches only about 58% of the average user LHR of 0.38. In comparison, random forests produce better results (LHR of 0.38), which are comparable to an average participant’s performance. Especially the MSE of distributions is significantly smaller than for the linear model (approximately three times). This shows that the random forest model is very well suited to predict user mappings, which matches our expectation that models that can handle interactions between geometric relationships should perform better (Section 4.1).

We conclude that the random forest model significantly improves upon the baseline methods in predicting user mappings and that it is capable of producing results comparable to users.

## 5.2 Comparison to Existing Approaches

In addition to the baseline methods presented in Section 5.1, we compare our results to two further approaches based on shape contexts [1], which is an established shape-matching approach. To our knowledge there is no method in current literature that explicitly attempts to find similar points in dissimilar shapes. Again, cross validation is applied using 30 groups as discussed in Section 5.1 to compute the average likelihood ratios and mean-squared error of likelihoods.

To construct a mapping from the source to the target shape using shape contexts, we employ two strategies. The first strategy results in a likelihood function over the target shape, while the second strategy finds a single most similar point.

**Inner Shape Contexts** Shape contexts describe the geometry in the neighborhood of a point. Two points have similar shape contexts if the geometry in their neighborhood is related by an approximately rigid transformation. Our first ex-

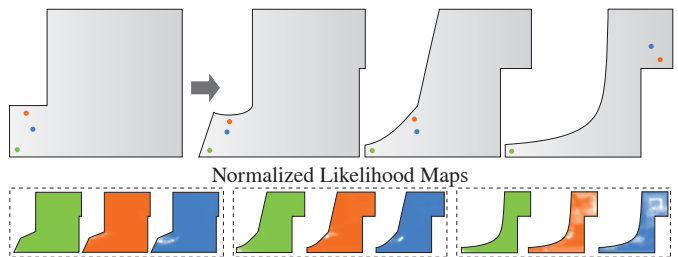


Fig. 9. Three points are mapped to increasingly different shapes using the random-forest model. Local geometric relations in the target shape become more important as locations with similar global relations become unavailable. The likelihood maps for each of the points are shown in the bottom row, higher likelihoods correspond to brighter values.

isting comparison method for mapping points between dissimilar shapes is a straightforward application of shape contexts to a regular grid inside the target shape. For each grid point, we compute a likelihood value as the similarity of the shape context centered at the grid point to a shape context centered at the query point, thus creating a densely sampled likelihood function over the target shape. This likelihood function can be compared to the prediction of our models.

Inner shape contexts have a significantly lower performance than all other non-naive mappings (Table 1). The performance of predicting the most likely user mapping (0.02) is as low as the *Random Mapping* baseline method and performs worse in predicting the likelihood of user mappings ( $3.93 \times 10^{-7}$ ). This shows that the shapes used in our study are indeed dissimilar and that a straightforward shape-matching approach does not provide satisfactory results, as can be seen in Figure 8, were Inner Shape Contexts can lead to unintuitive mappings.

**B-Spline Shape Contexts** The second strategy employs shape contexts only to establish point-wise bijective correspondences between the boundaries of two shapes. Then, a B-spline grid is fitted to the target shape and optimized to minimize distortion while respecting the correspondences. The B-spline grid defines a bijective mapping between the source- and target shape. We use the image of the query point as prediction for the target point.

This method has the great disadvantage that points can be located outside the target shape, since the B-spline grid cannot be constrained to remain inside highly concave shapes without introducing too much distortion, which would lower the quality of the resulting mapping. In such cases we shift the point to the closest location inside the target shape. Additionally, this method cannot be used to predict a likelihood function in the target shape. Therefore, only the LHR performance measure is used for comparisons. Compared to the *Random Mapping* baseline method, it performs noticeably better in predicting the most likely user mapping (0.17), but still relatively bad in comparison to the *Global Mapping* baseline method and the proposed models from Section 4.2. And because points that would be placed outside the target shape have to be post-processed in some way (e.g., in our case shifting them to the closest boundary position), the mapping results can be quite unpredictable.

## 5.3 Qualitative Evaluation

We provide several qualitative results for our random forest model as best-performing predictor in Figure 7, and compare it to the ground-truth solution, linear model and inner shape context model. In each example, points were transferred from a source shape to a target shape by placing them at the maximum of the predicted placement likelihood in each target shape. Figure 8 provides some additional results of query points that were not part of the user study. The source and target shapes

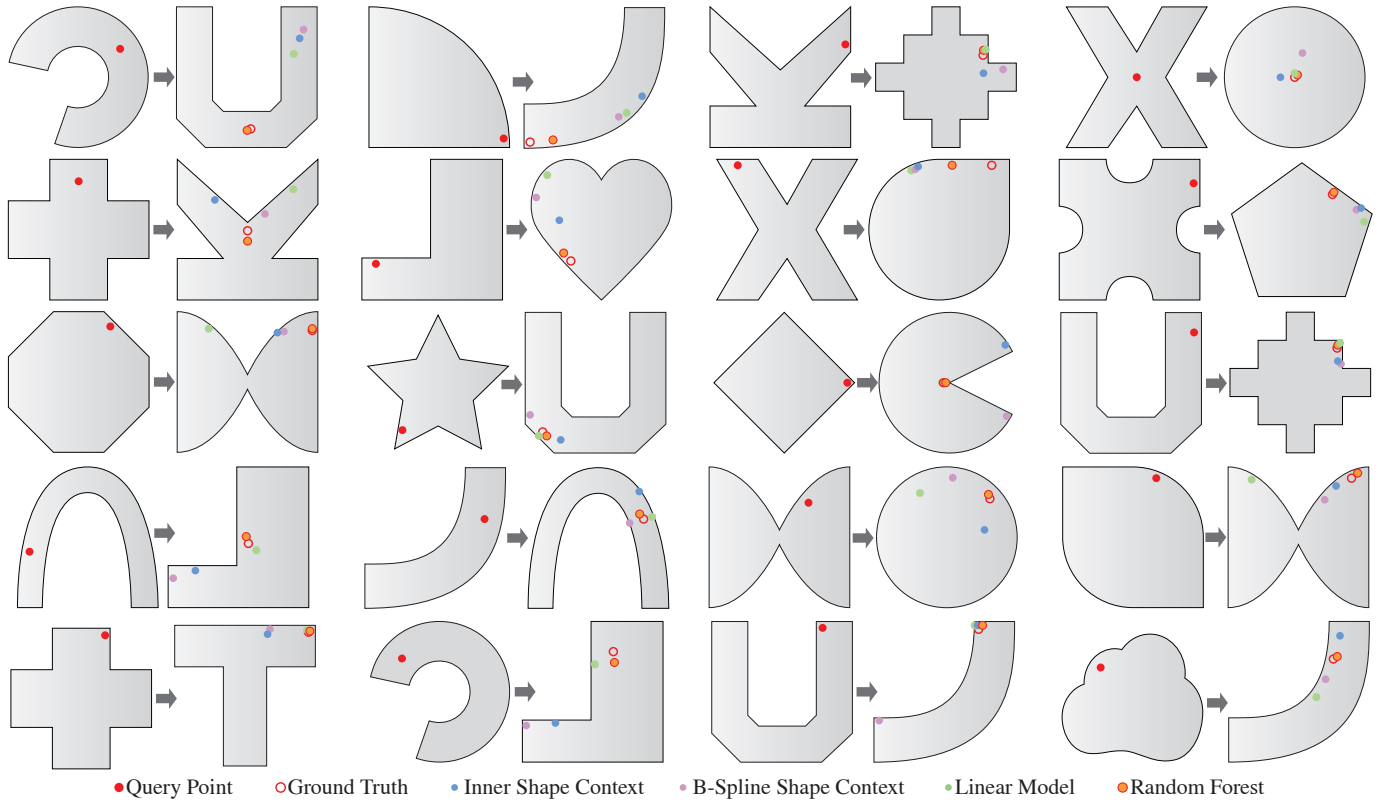


Fig. 7. A comparison of a subset of query point mappings using different approaches and comparing them to the ground-truth mapping obtained from user data. The random forest model consistently outperforms the other methods.

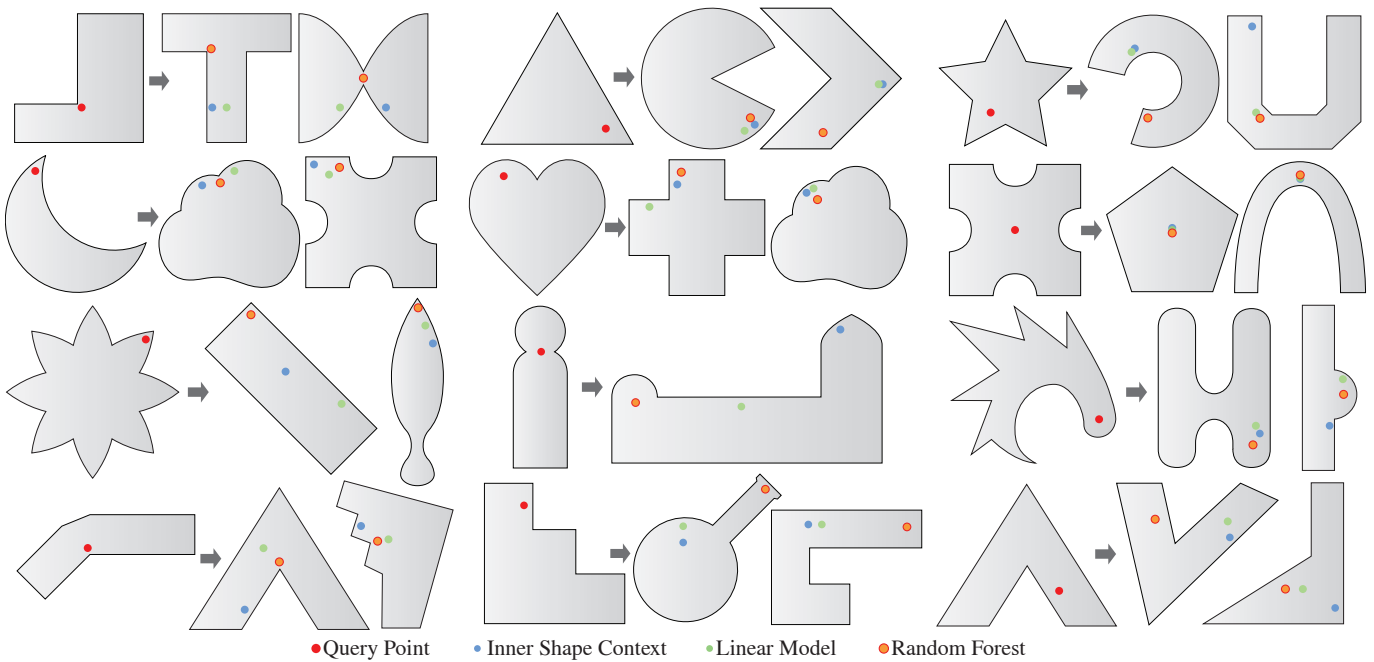


Fig. 8. A comparison of point mappings using different approaches. The top two rows show results for shapes that were also part of the user study, while the lower two rows consist of novel shapes. Only the random forest model consistently maps points to plausible locations, while other models often lead to unintuitive placements.



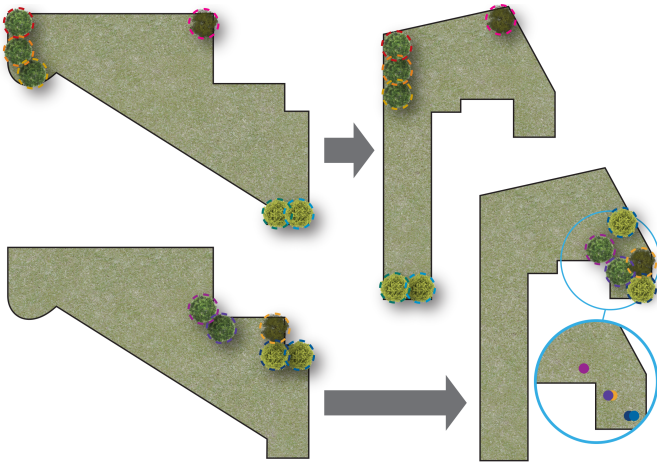


Fig. 10. Mapping trees between garden layouts, keeping a minimum distance between trees (indicated as dashed circles). Objects are mapped to their likelihood maxima only if there is sufficient distance to previously placed points (top), otherwise they are placed at the next best position (bottom). The inset shows the actual maxima locations.

are either taken from the set of shapes used in the user study (the two top rows), or novel shapes are used (the two lower rows). Note how the positions found by the random forest model are typically more intuitive than positions found by the other two models.

This is consistent with the quantitative results described in Section 5.1. We ascribe this superior performance to the ability of the random forest model to capture dependencies between feature values (see Section 4.2.4). For example, we can observe that the importance of global relationships increase as the distance of the query point from shape parts used in local relationships, like corners, increases (for example, compare the results for the first and last column).

In Figure 9, we transfer points to three shapes that vary from similar on the left to dissimilar on the right. We can observe that points are placed at locations with similar local and global relationships on the left, and rely more on local relationships towards the right (for example, the distance of the red and blue points from a corner or the boundary), as locations with similar global relationship become unavailable.

## 6 Applications

In the content-creation industry, reusing assets like meshes and textures is common practice. However, given an object layout, texture or material created for a model, retargeting them successfully to other models with different shapes is usually done manually and requires a significant amount of work. In this section, we show how our findings can be applied to simplify this process. In particular, we use our trained random-forest model to transfer scene layouts in rooms or parks to rooms or parks of different shapes (Section 6.1), and to transfer *control maps* between different shapes (Section 6.2). These control maps are parameters defined over the shape that can be used to guide procedural generation of textures, materials or geometry.

### 6.1 Object Transfer

Our models can be used to automate the transfer of objects between different shapes. We perform several experiments where we retarget scene layouts to new bounding shapes. Objects are transferred between the shapes consecutively, one at a time. For each object in the source shape, our random-forest model computes a *placement likelihood map* in the target shape. High likelihoods correspond to locations that are predicted to be intuitively similar to the position of the original object.

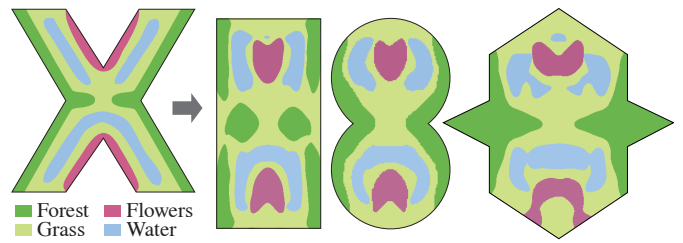


Fig. 11. Mapping a park layout defined from an “X”-shaped park to three other parks of different shapes. Labeled regions in the park are shown in different colors. Each region is defined and transferred using a binary control map that is 1 at the region and 0 everywhere else.

Transferred objects should not intersect, and in many cases, objects need to be placed at a minimum distance from each other. For example, trees that are transferred between two garden layouts cannot occupy the same space, so we need to rearrange them in a way that is plausible to the user. Figure 10 shows an example where we constrain target positions to have a minimum distance to previously transferred objects. We greedily position objects with higher likelihoods first, and place the remaining objects at locations that do not violate the minimum distance constraint. This greedy approach is simple, but gives good results as long as the layout is not populated too densely. For dense layouts, using non-linear optimization to maximize the placement likelihood of objects while respecting the minimum distance constraints would yield a better, albeit computationally more expensive solution.

### 6.2 Control Map Transfer

Another application is automating the transfer of *control maps* for procedural content. We define control maps as 2D functions on a surface with values between zero and one. Each map controls a parameter of procedural content on the surface. In large collections of objects like houses, parks or rooms, procedural details like ponds, plant growth, decay or dust might need to be added. In these large-scale settings, transferring control maps manually (e.g., by painting) can be very time consuming or even impossible to do in a reasonable time frame.

To transfer a control map automatically, we have two options to use the placement likelihoods predicted by a model: In the forward-mapping approach, we predict the likelihood maps from each point in the source shape to the target shape. We can then splat the control map value of each point in the source shape to high-likelihood regions in the target shape. This approach has the main disadvantage that some regions that are not covered by the splats may be left with undefined values. Instead, we opt for a backward-mapping approach. We compute the likelihood maps from each point in the target shape to the source shape. The control map value at each point in the target shape is then gathered from high-likelihood regions of the source shape.

More specifically, we first sample source- and target shapes densely in regular grids. For each point in the target shape, we compute the likelihood to all points in the source shape using one of the models described in Section 4.2. The control map value at a point in the target shape is then the average of all control map values in the source shape, weighed by their likelihood values.

Our method is agnostic to the procedural generation algorithm used, so we transfer each control map separately and assume there are no dependencies between control parameters.

Figure 11 shows how we can apply this method to predict the mapping of labeled regions, like forests, ponds and flower beds, between parks of different shapes, using the random-forest model. The regions are defined through binary control-map layers (which may be obtained through automatic or man-

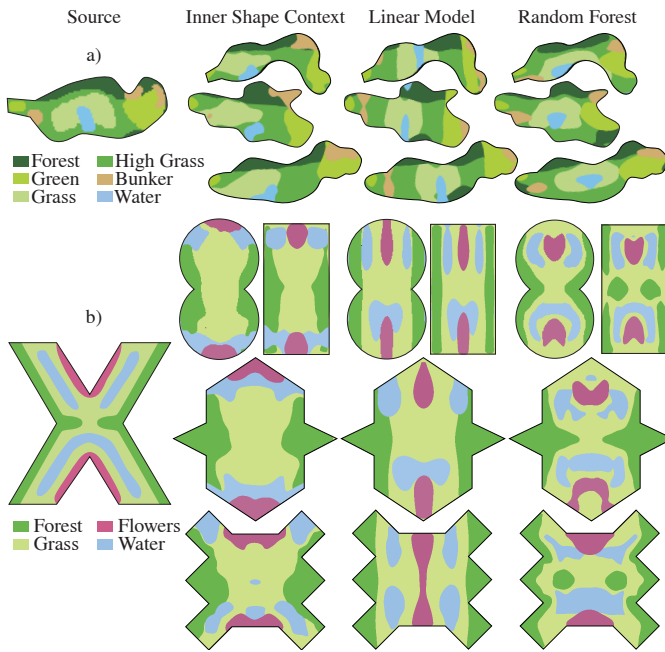


Fig. 12. Comparing statistical models for control-map transfer. Since the random-forest model more accurately predicts user mappings than the other two models, the colored regions are mapped to intuitively more similar parts of the target shapes.

ual segmentation), one layer for each label. Transferring these control maps using the method described above gives us a class membership likelihood for each label at each target point. We choose the label with highest likelihood. While this leads to plausible results, optimization strategies may also be used to allow the addition of complex constraints.

The results in Figure 11 were generated with the random-forest model; Figure 12 shows a comparison to other models. The superior predictive power of the random-forest model results in more intuitive mappings.

Our technique produces discontinuous mappings, which are preferable for our use cases, because we want to support one-to-many and many-to-one mappings. For example, we would like the corners of a five-sided star to be mapped to the corners of a four-sided star. Examples of one-to-many and many-to-one mappings can be observed in Figure 12.

## 7 Limitations and Future Work

In our study, we focused on simple shapes without high-frequency detail or semantics. Further investigation is needed to generalize our findings to more complex shapes. Also, we focus our evaluations on 17 simple geometric relationships of which we used 15 in our models. Adding additional geometric information could possibly improve the performance of our models and algorithms. We would also like to experiment with neural networks (e.g. CNNs) as an alternative approach to model the mapping between shapes based on our study data.

Our method can be applied to 2D manifolds that can be mapped to a plane. In that case, the boundary of the manifold would provide all the features. In future work, we could use additional features of a manifold, such as differential properties of the surface.

With our model, it should be relatively straightforward to combine the placement likelihoods predicted from different source shapes or query points in a single target shape (e.g., taking the average, multiplying). This would allow combining the layout of multiple source shapes in a single target shape, a direction that might be interesting to explore in future work. We would also like to explore mappings based on our model

that are constrained by additional desirable properties, such as smoothness constraints; one possible approach is to use our control maps and the smoothness constraints as energy terms in a Markov Random Field optimization. When placing multiple points, it may be beneficial in some cases to introduce constraints between these points, like maintaining their relative distances. As future work, it would be interesting to explore optimization algorithms to maximize pointwise similarity while respecting given constraints between points.

Lastly, even though our models can describe and predict user mappings, we do not claim to know the actual reasoning made by users to pick particular target points.

## 8 Conclusions

In this paper, we have presented results of a study that investigates how people map points between dissimilar shapes. Participants were asked to map a point in a source shape to a target shape. Our findings suggest that even though the shapes are dissimilar, users agree in their point placements. Furthermore, we investigated simple geometric relationships between the points placed by users and the containing shape and found a strong correlation between these relationships in the target shape and corresponding relationships in the source shape. We proposed two models based on these relationships, which been used to successfully predict user mappings with a higher accuracy than well-known shape-matching methods and even achieved results similar to actual participants when including dependencies between geometric relationships in the model. We demonstrated how these models can be used to develop new algorithms that allow users to map procedural content between shapes with low similarity.

## Acknowledgements

This research was partially financed by the Austrian Science Fund project Nr. FWF P24600-N23.

## References

- [1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE PAMI*, 24(4):509–522, 2002.
- [2] I. Biederman. Recognition by components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.
- [3] X. Chen, A. Sapiro, B. Pang, and T. Funkhouser. Schelling points on 3d surface meshes. *ACM TOG*, 31(4):29:1–29:12, July 2012.
- [4] F. Cole, A. Golovinskiy, A. Limpaecher, H. S. Barros, A. Finkelstein, T. Funkhouser, and S. Rusinkiewicz. Where do people draw lines? *ACM TOG*, 27(3):88:1–88:11, Aug. 2008.
- [5] F. Cole, K. Sanik, D. DeCarlo, A. Finkelstein, T. Funkhouser, S. Rusinkiewicz, and M. Singh. How well do line drawings depict shape? *ACM TOG*, 28(3):28:1–28:9, July 2009.
- [6] A. Emilien, U. Vimont, M.-P. Cani, P. Poulin, and B. Benes. Worldbrush: Interactive example-based synthesis of procedural virtual worlds. *ACM TOG*, 34(4):106:1–106:11, July 2015.
- [7] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.
- [8] M. S. Floater. Mean value coordinates. *Comput. Aided Geom. Des.*, 20(1):19–27, Mar. 2003.
- [9] J. D. Gibbons and S. Chakraborti. Nonparametric statistical inference. *Biometrics*, 67(3):1182–1183, 2011.
- [10] K. Grill-Spector, T. Kushnir, S. Edelman, Y. Itzhak, and R. Malach. Cue-invariant activation in object-related areas of the human occipital lobe. *Neuron*, 21:191–202, 1998.
- [11] P. Guerrero, T. Auzinger, M. Wimmer, and S. Jeschke. Partial shape matching using transformation parameter similarity. *CGF*, 34(1):239–252, 2015.
- [12] P. Guerrero, S. Jeschke, M. Wimmer, and P. Wonka. Edit propagation using geometric relationship functions. *ACM TOG*, 33(2):15:1–15:15, Apr. 2014.
- [13] J. H. Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Commun. ACM*, 7(12):701–702, Dec. 1964.

- [14] J. V. Haxby, E. A. Hoffman, and M. Gobbini. The distributed human neural system for face perception. *Trends in Cognitive Sciences*, 4(6):223 – 233, 2000.
- [15] T. K. Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition*, pages 278–282, 1995.
- [16] K. Hormann and M. S. Floater. Mean value coordinates for arbitrary planar polygons. *ACM TOG*, 25(4):1424–1441, Oct. 2006.
- [17] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. *ACM TOG*, 26(3), July 2007.
- [18] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3d mesh segmentation and labeling. *ACM TOG*, 29(4):102:1–102:12, July 2010.
- [19] Z. Kourtzi and N. Kanwisher. Representation of perceived object shape by the human lateral occipital complex. *Science*, 293:1506–1509, 2001.
- [20] P.-E. Landes, B. Galerne, and T. Hurtut. A shape-aware model for discrete texture synthesis. *CGF*, 32(4):67–76, 2013.
- [21] Y. Lipman, D. Levin, and D. Cohen-Or. Green coordinates. *ACM TOG*, 27(3):78:1–78:10, Aug. 2008.
- [22] J. Ma, J. Zhao, J. Tian, Z. Tu, and A. L. Yuille. Robust estimation of nonrigid transformation for point set registration. In *CVPR*, June 2013.
- [23] R. Malach, I. Levy, and U. Hasson. The topography of high-order human object areas. *Trends in cognitive sciences*, 6(4):176–184, 2002.
- [24] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London*, 200:269–294, 1978.
- [25] J. McCrae, K. Singh, and N. J. Mitra. Slices: A shape-proxy based on planar sections. *ACM TOG*, 30(6):168:1–168:12, Dec. 2011.
- [26] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM TOG*, 25:560–568, 2006.
- [27] A. Rosenberger, D. Cohen-Or, and D. Lischinski. Layered shape synthesis: Automatic generation of control maps for non-stationary textures. *ACM TOG*, 28(5):107:1–107:9, Dec. 2009.
- [28] M. J. Tarr and S. Pinker. Mental rotation and orientation-dependence in shape recognition. *Cognitive Psychology*, 21(2):233 – 282, 1989.



**Michael Hecher** is a Ph.D. student at TU Wien, Vienna, Austria since October 2012 and received his M.Sc. in 2012 from the same institution. He is also a member of the Center for Geometry and Computational Design at TU Wien in the Rendering and Modeling research group. His research focuses on psychophysiological methods in computer graphics, real-time rendering and applications of machine learning.



**Paul Guerrero** is a post-doctoral researcher at the Smart Geometry Processing Group, UCL, London, UK. He received his M.Sc. and Ph.D. in computer science from the TU Wien, Vienna, Austria in 2008 and 2014, with several research stays at the Visual Computing Center in KAUST, Saudi-Arabia. His research interests include geometry processing, computer vision, computer graphics, image processing and machine learning.



**Peter Wonka** received his Ph.D. in computer science and his MS in urban planning from the TU Wien, Vienna, Austria, in 2001 and 2002, respectively. He was a postdoctoral researcher at the Georgia Institute of Technology, Atlanta, GA, for two years. He is currently a professor at the Computer, Electrical and Mathematical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. His research interests include computer graphics, visualization, computer vision, remote sensing, image processing, and machine learning.



**Michael Wimmer** is an associate professor at the Institute of Computer Graphics and Algorithms at TU Wien, Austria, where he received an M.Sc. in 1997 and a Ph.D. in 2001. His current research interests are real-time rendering, computer games, point-based rendering, procedural modeling and shape modeling. He has coauthored many papers in these fields, and was papers co-chair of EGSR 2008, Pacific Graphics 2012, and Eurographics 2015, and is associate editor of *Computers & Graphics*, *IEEE Transactions on Visualization and Computer Graphics*, and *Computer Graphics Forum*.