9-2018

A vector field design approach to animated transitions

Yong WANG
Singapore Management University, yongwang@smu.edu.sg

Daniel ARCHAMBAULT

Carlos E. SCHEIDEGGER

Huamin QU

A Vector Field Design Approach to Animated Transitions

Yong Wang, Daniel Archambault, Carlos E. Scheidegger and Huamin Qu

Abstract—Animated transitions can be effective in explaining and exploring a small number of visualizations where there are drastic changes in the scene over a short interval of time. This is especially true if data elements cannot be visually distinguished by other means. Current research in animated transitions has mainly focused on linear transitions (all elements follow straight line paths) or enhancing coordinated motion through bundling of linear trajectories. In this paper, we introduce animated transition design, a technique to build smooth, non-linear transitions for clustered data with either minimal or no user involvement. The technique is flexible and simple to implement, and has the additional advantage that it explicitly enhances coordinated motion and can avoid crowding, which are both important factors to support object tracking in a scene. We investigate its usability, provide preliminary evidence for the effectiveness of this technique through metric evaluations and user study and discuss limitations and future directions.

Index Terms—Information Visualization, Animated Transitions, Vector Field Design.

1 INTRODUCTION

ANIMATED transitions, whether interactive or not, are pervasive in information visualization and visualization in general. In particular, they are important for the visualization of dynamic data [1] and for representing changes in the data [2], [3]. As these transitions are in widespread use, techniques have been developed and studies have been run to understand and leverage when they are effective for the visualization of data [4], [5], [6], [7]. Thus, we are beginning to understand where and when to use animated transitions.

Animated transitions have been shown to be useful in situations where there are large changes in the spatial positions of the data over a very short period of time[1] and when such scenes cannot be linked using other means [8]. Despite the fact they are slower to use in general, even the strongest critics of animation for the purposes of visualization admit that animations can be useful in this context: *"At this point then, the most promising uses of animation seem to be to convey real-time changes and reorientations in time and space."* [9]. Both interactive and non-interactive forms of animation can help users follow these large changes in non-spatial data. Given this particular benefit, it is interesting to study methods for facilitating short transitions that support the user's orientation in datasets.

A number of techniques have focused on linear paths as a basis for animated transitions. In particular, linear extrusion into the third dimension [7], distortion of data point speed [5], and staggered linear transitions [4] have been explored. These techniques have been shown to produce effective animated transitions with varying degrees of success. However, these techniques consider the motion of all points independently in the scene and do not explicitly leverage coordinated motion (Gestalt law of common fate) to enhance the maintenance of perceptual grouping, which has shown to be advantageous for object tracking [10].

Recent work has looked at the motion of groups of points by bundling their linear trajectories together in order to increase the coordinated motion of the points between the initial and final positions [6]. This work does leverage coordinated motion of the data points by bundling the motion of these points together. It can increase the tracking accuracy if only the group identities are required. However, it does so at the expense of *crowding* [11] or placing the objects in the same group very close together, where the identities of objects can be confused during the animated transition. But animated transitions enabling better track of individual objects can more clearly convey the transformation between different visual states and help users well understand the exact correlation between different scatterplot views, which is also discussed in previous work [4], [5].

In this paper, our goal is to design effective animated transitions between two spatial positions of data with the following two desirable properties:

1) *Data points should not pass too close to each other in the transition to avoid confusion of their identities. That is, transitions should reduce crowding [11].*
2) *The motion of data points within the same group should follow the Gestalt law of common fate to improve the performance of multiple object tracking [10]. That is, points in the same group should move together with similar trajectories. We call this* ***coordinated motion*** *in this paper.*

These properties have been shown to help with object tracking [10], [11]. As for transition trajectories, prior animation research indicates that straight-line transitions tend to result in crowding [12], [13]. Arcs provide a method for smooth and natural motion [14] and studies in vision science [15], [16], [17] have found that non-linear paths have only a minor influence on performance. Therefore, in contrast to the majority of animation techniques using linear paths, we consider using non-linear paths for animated transitions.

One way to achieve these desirable properties is to exploit the

- *Yong Wang and Huamin Qu are with the Hong Kong University of Science and Technology.*
  E-mail: {ywangct,huamin}@cse.ust.hk
- *Daniel Archambault is with Swansea University.*
  E-mail: d.w.archambault@swansea.ac.uk
- *Carlos Scheidegger is with the University of Arizona.*
  E-mail: cscheid@email.arizona.edu

1. http://research.microsoft.com/en-us/um/redmond/events/fs2010/presentations/Robertson_Vizualization_and_Interaction_RFS_071210.pdf

vector field design literature [18], [19], [20]. Vector field design considers the problem of designing vector fields for the purposes of flow simulation. If one were to closely examine the problem of designing animated transitions in this light, we could consider the constraints placed on the animated transition as constraints placed on the flow. These constraints can be either designed by the user or computed by an automated algorithm, ensuring that the constraints preserve the desirable properties needed for a better animated transition. The data points involved in the transition can be treated as particles advected by this flow in order to achieve the desired animated transition. This transition can now follow any linear or non-linear path, allowing for greater flexibility in avoiding crowding and increasing coordinated motion at the expense of linear paths of the data elements.

Using this idea, we present a technique to enable users to freely design animated transitions for clustered data between two scenes, given a pre-defined clustering. Similar to prior work [6], we also regard the points with spatial proximity of starting and ending positions (thus, similar moving directions) to be the same group. We provide users with the flexibility to add constraints for better transitions through either manual sketching or an automated approach which have comparable effectiveness. We first conduct an informal user interview to evaluate the usability of the proposed technique. Then, we evaluate the transitions that we design against approaches such as trajectory bundling [6] and straight line transitions through both metric evaluations and a user study. The metric study demonstrates that the proposed transition design technique decreases occlusion and deformation, providing a good compromise between crowding and coordinated motion. However, bundled transition has lower dispersion, meaning that the proposed technique may not be a choice as good as bundled transition when only group identities are required. The user study shows our transition design technique helps improve target tracking performance in transitions of high occlusion and maintains or improves response time when compared with straight line transition.

2 Related Work

The related work of this paper includes three areas: animation-related techniques, object tracking studies and vector field design.

2.1 Animation-related Techniques

Animation is frequently used in the design of interactive visualization systems because animations provide an engaging method for presentation, can visually attract attention [9], [21], [22] and provide potential benefits for particular tasks. On the other hand, designers must be cautious when using animations as animations can also be a source of distraction [3], [23] and impose a greater cognitive demands [24]. Because of these reasons, many animation-related techniques have been proposed, which can be classified into two types in terms of whether manual effort is needed or not: ***automated animation techniques*** and ***animation authoring techniques***.

Automated animation techniques generate animated transitions without user involvement. Dragicevic et al. [5] introduced animated transitions based on different temporal distortions and concluded that slow-in/slow out performs the best. Staggered animated transitions [4] delay the starting time of elements individually but seem to have negligible or even negative effects on multiple object tracking performance. Other automated techniques use non-linear transition paths instead of straight lines. For example, Yee et al. [13] interpolated polar coordinates to generate non-linear transition path for the animation of radial graph layouts. Dragicevic et al. [25] used curved trajectories to clarify text animations. Schlienger et al. [26] proposed conveying data information through different motion trajectories. Trajectory bundling [6] of objects in the same group has also been considered. The technique helps in tracking groups of points but suffers from crowding.

Animation authoring techniques require user interaction for the design of animation. For example, motion sketching is used by researchers in computer graphics area to generate 3D animation conforming to physical and geometric constraints [27], [28]. However, these algorithms do not apply to coordinated animated transitions. Animation sketching systems [29], [30], [31] are developed in HCI community, focusing on improving the usability and convenience of freely creating animations by incorporating different types of interaction operations. Our technique enables animated transition design between a pair of scatterplots.

User involvement can be both an advantage and disadvantage for animation-related techniques. For example, it takes less time for users to use automated animation technique to generate animated transitions, but also limits the flexibility for users to specify animations in order to make them satisfy their specific animation requirements. Taking this issue into account, the proposed technique provides users with the flexibility to have both. Similar to Draco [31], the proposed technique is also based on vector fields and enables motion sketching. However, the animated transition design in this paper enforces further constraints, such as fixed starting and ending positions, crowding avoidance, and coordinated motion.

2.2 Object Tracking Studies

The goal of animated transitions is often to support the ability of the user to follow data which is changing spatially. Thus, object tracking studies, in psychology, are highly relevant to this area. These studies are a source of inspiration for designers and evaluators of animated transitions in the visualization field. Pylyshyn and Storm [17] showed that humans can track up to five objects simultaneously against a field of identical and randomly-moving distractors. Tracking accuracy degrades as the number of tracked targets increases. A number of studies that followed in this area further confirmed tracking performance is not only adversely affected by the target number but also object speed [10], [32], [33] and tasks itself [34], [35].

Crowding, or indistinguishable targets passing too close to each other, is another important factor influencing tracking performance [36], [37], [38]. If the targets and distractors are visually similar and pass very close to each other, they become difficult to distinguish and therefore track. Franconeri et al. [11], [37] claimed that object tracking performance is limited only by crowding and the influence of speed may actually just be crowding from another perspective. Other factors such as, occlusion with boundaries [39], trajectory changes or curved paths [15], [16], surprisingly seem to have only minor effects on tracking multiple objects. In addition, Yantis [10] showed that we form perceptual groups of moving targets. When attending targets, we mentally form a polygon comprising of these targets. This finding is consistent with the *Gestalt law of common fate*, as a perceptual group, consisting of the multiple targets moving as one, can be more easily followed during the animation.

In our paper, we draw on this research to help with our animated transition design technique. In particular, we aim to avoid crowding and encourage coordinated motion. As non-linear paths seem to have little negative effect, we relax this constraint to achieve our two objectives.

2.3 Vector Field Design

A *vector field* refers to a function where each point in a space is assigned a vector – often, the space is a surface and the vectors are tangent to the surface. A vector field is frequently required as input in many applications including texture synthesis, non-photorealistic rendering, anisotropic shading and fluid simulation, to control surface appearance [40]. Due to these applications in computer graphics, *vector field design*, or methods to create vector fields that satisfy user constraints, has been an important area of research [41], [42]. Theisel [19] presented a two-dimensional vector field design system, allowing a user to specify a set of control polygons to describe a topological skeleton. Zhang et al. [20] proposed an approach that is based more directly on vector field topology to enable users to create continuous vector field on curved surfaces.

Vector fields have been applied to information visualization. For example, to conduct trajectory clustering, Ferreira et al. [43] adopted vector fields to delineate similar trajectories and cluster them by iteratively assigning the "nearest" label similar to a K-means algorithm. To support flow visualization with bounded spatial and temporal errors, a new method called *edge maps* [44] is proposed to represent vector fields. Vector fields, in particular curve shortening flow, are also applied to improving the appearance of Euler diagrams [45]. Inspired by previous studies especially [43], we apply vector field design techniques in the design of effective animated transitions.

3 Animated Transition Design

The proposed technique can be divided into three building blocks: path generation, vector field computation and point advection. As input, the proposed technique also requires a pre-defined clustering of the points as in [6]. This clustering is mainly based on the similarity of spatial positions and moving directions and can be achieved by any method. Such a clustering requirement does not necessarily limit the application of the proposed technique in common animated transitions. In an extreme case where no points share spatial proximity and movement similarity, each point can be regarded as a single group. We first generate a path for each group of points to represent its overall movement. This path is well designed to avoid occlusion between groups during the transition. The paths are used to generate a vector field for each group. Using this vector field, points can be advected creating a smooth and coordinated transition.

3.1 Path Generation

The initial path for each group can be computed from a manual sketch or an automatic algorithm as we describe below.

3.1.1 Manual Sketching

We provide an interface allowing for the flexible design of animated transitions that can have non-linear paths. As shown in Fig. 1, the initial positions of the data points are solid and the final positions of the points are hollow. Users can specify an initial path for each group of points with the help of the following interactions:

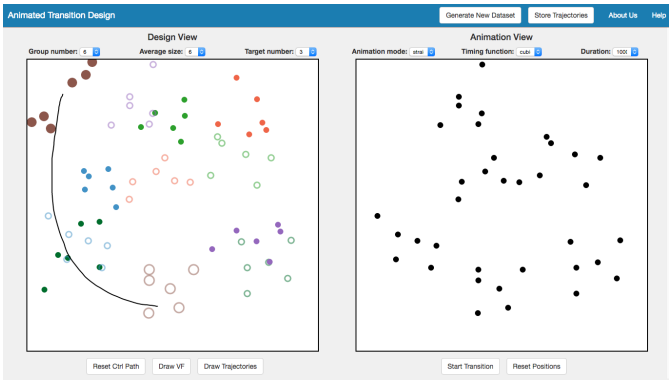


Fig. 1: Animated transition design interface. The animated transition design window overlays the initial (solid) and final (hollow) positions of the points in the same window. Groups are indicated with color and can be defined using any desired method. For each group, the user selects a set of points and draws a path for them from solid to hollow (black line). By specifying this path, the user can avoid other points in the scene. Crowding can be reduced and coordinated motion is enhanced in the designed transition.

Point Group Selection: Users can lasso select a group of initial (solid) points to highlight them.

Path Sketching: Users can sketch a path from the selected group to their final positions by brushing a curve between the centers of them, avoiding other points in the transition to reduce crowding. Once complete, the selection is removed.

Undo: Users can cancel last operations by simply pressing "ESC" key on the keyboard when necessary.

There are also other related interactions including viewing the actual path of each point, testing the generated transitions, changing animation timing, etc.. Through these interactions, users can conveniently sketch their desired path for each point group.

3.1.2 An Automated Approach

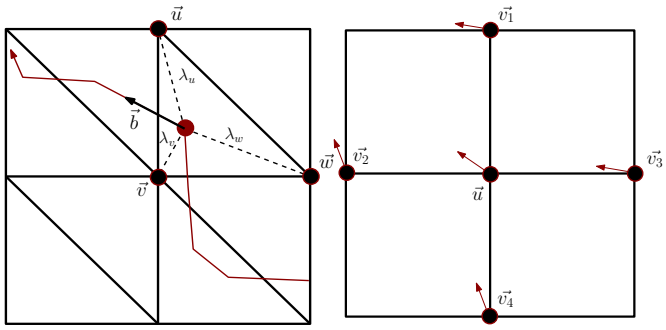
Manual sketching provides users with the flexibility to design any animated transitions they want, but it may not always be preferred as manual interaction is required. Thus, we propose an automated approach based on a force model to generate an initial path for each point group. We regard the path generation in this section as a path planning problem in 2D space, where we sample a fixed number of time steps on the path of each point group. Thus, the goal here is quite intuitive: we should avoid occlusion of data point at each time step and make position transition between adjacent time steps smooth. This can be achieved by introducing three different forces:

Repulsion Force aims at avoiding the trajectory overlapping between different groups at the same time step, so it is computed between any two groups only at exactly the same time step. The repulsion force that group q exerts on group p at the same time step i is defined as follows:

$$F_p^r = \frac{c_p}{\|q_i - p_i\|} \tag{1}$$

where c_p is the coefficient directly proportional to the length of straight-line trajectory of group p.

Attraction Force is defined for the same point between adjacent time step. It is introduced to avoid highly curved and long



(a) Path constraint **(b)** Smoothing constraint

Fig. 2: Two constraints for vector field computation. (a) Path constraint: let $\vec{b}$ be one of the vectors defined by line segments of initial group path, then we can use barycentric coordinates to define constraints to impose on $\vec{u}$, $\vec{v}$ and $\vec{w}$ of our linear system. (b) Smoothing constraint: it corresponds to Laplacian smoothing in this system. Let $\vec{z}$ correspond to the vector at a grid corner. We define a constraint on $\vec{z}$ so that it is the average of its neighbors $\vec{v_1}$, $\vec{v_2}$, $\vec{v_3}$ and $\vec{v_4}$.

trajectories. The attraction force on group p at time step i is as follows:

$$F_p^a = k_p((p_{i-1} - p_i) + (p_{i+1} - p_i)) \tag{2}$$

where the coefficient k_p is negatively proportional to the length of straight-line trajectory of group p.

Smoothening Force is also defined for the same point between adjacent time steps. The objective of this force is to encourage smooth trajectories. The smoothening force on group p at time step i is defined as the weighted average of its current and adjacent time steps:

$$F_p^s = c_s((p_{i-1} + p_{i+1})/2 + (1 - c_s) * p_i) \tag{3}$$

where c_s is a constant smoothening coefficient.

By iteratively applying the above forces to all the point groups, initial paths with good occlusion avoidance and smoothness are generated.

3.2 Vector Field Computation

In this phase of the algorithm, we construct our overconstrained system and it describes how a vector field is generated for a given group based on the initial path generated in Section 3.1.

To begin, we overlay an $n \times n$ regular grid over the scene and use it to define our vector field. To compute the appropriate vector field, we will construct a system of linear equations in the standard form:

$$A\mathbf{x} = \mathbf{b} \tag{4}$$

where A is an $m \times n^2$ matrix with $m > n^2$. The value of $\mathbf{b}$ is given by our constraints so we only need to solve for $\mathbf{x}$. As the system is overconstrained, no solution exists. Thus, an $\mathbf{x}$ is found that minimizes the error in the system. This solution can be solved for both x and y components independently, defining a vector field over the grid that can be used to advect the points in the transition.

In this system of linear equations, we define two types of constraints: ***a path constraint*** that constrains vector field to follow the transition trend of each point group and ***a smoothing constraint*** that smooths vector field outwards from the initial path of each point group, defining a smooth vector field that is wide and correctly influences the points in the group.

3.2.1 Path Constraint

The path constraint for each point group is specified by the polyline l generated from either manual sketching by users or the automated approach described above. We can interpret each segment of l as a vector in the direction that the segment was drawn. This vector is located within a cell of the regular grid superimposed on the scene. We can decompose the cells of this grid into triangles as shown in Fig. 2a.

Let us assume that the tail of $\vec{b}$ lies within the triangle with vertices u, v, and w. The linear system wishes to find a vector field at all corners of the grid. Let $\vec{u}$, $\vec{v}$, and $\vec{w}$ be the vectors at these grid corners. To constrain this vector field at u, v, and w, we impose constraints based on the barycentric coordinates of the tail of $\vec{b}$. Let λ_u correspond to the barycentric coordinate of the tail with respect to u. As λ_u is a barycentric coordinate, $\lambda_u \in [0, 1]$ with 1 corresponding to the point having the exact same position as u and 0 corresponding to when the point is collinear with the edge $\overline{vw}$. The same definition exists for λ_v and λ_w with $\lambda_u + \lambda_v + \lambda_w = 1$. We can define a constraint for each $\vec{b}$ (or each segment of p) as follows:

$$\lambda_u\vec{u} + \lambda_v\vec{v} + \lambda_w\vec{w} = \vec{b} \tag{5}$$

For each line segment of l, we add one of these constraints to the matrix A. Through these constraints, the polyline (i.e., the initial path of each point group) will have an influence on the vector field.

3.2.2 Smoothing Constraint

The smoothing constraint extends the influence of path constraint and helps ensure a smooth vector field. The smoothing constraint is equivalent to Laplacian smoothing of the vector field and is imposed on the grid as defined in Fig. 2b.

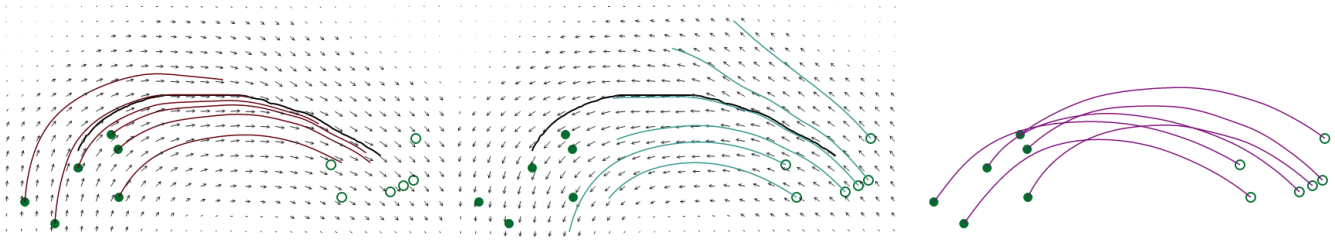
When defining this constraint, let $\vec{z}$ be the vector at a grid corner z. Let the points v_1, v_2, v_3, and v_4 correspond to the four adjacent grid corners to u with the corresponding vectors $\vec{v_1}$, $\vec{v_2}$, $\vec{v_3}$, and $\vec{v_4}$. Laplacian smoothing defines $\vec{z}$ to be the average of its neighbors:

$$\vec{z} - \frac{1}{4}\vec{v_1} - \frac{1}{4}\vec{v_2} - \frac{1}{4}\vec{v_3} - \frac{1}{4}\vec{v_4} = 0 \tag{6}$$

If the corner of the grid lies on a boundary, the smoothing constraint only considers the two or three values of $\vec{v_i}$ as appropriate. As a smoothing constraint is defined for each corner of the grid, it imposes n^2 constraints on the linear system. These constraints propagate the path constraints outwards into the grid and help ensure a smooth vector field.

3.2.3 Solving the Overconstrained System

The path and smoothing constraints form a linear system. In this linear system, $\mathbf{x}$ is a row vector with an entry for each corner of the grid. We can find a solution for the x and y components of the vectors at each element of $\mathbf{x}$ independently. The constraints defined above are loaded into the matrix A which has the defined coefficient when an element of $\mathbf{x}$ is involved in the constraint and 0 otherwise. The vector $\mathbf{b}$ is the component of $\vec{b}$ if the constraint is an initial group path constraint and 0 for smoothing constraints. Let D be the initial group path constraints and B the right-hand side of these constraints. Let S represents the smoothing



(a) Forward Advection (init → final) **(b)** Reverse Advection (final → init) **(c)** Final Interpolated Path

Fig. 3: Advection of points in the vector field. Both vectors and particle traces are shown in this diagram. **(a)** Points are advected from their starting positions towards their ending positions using the vector field computed in the first step. **(b)** Each vector in the field is multiplied by -1, reversing its direction. The points are then advected from their final positions towards their starting positions. **(c)** Given these two particle traces, interpolate the positions of both to obtain a smooth trajectory for each point in the group.

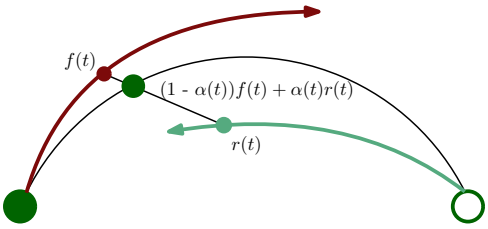


Fig. 4: Interpolating the two particle trajectories. The particle trajectory resulting from advecting the points from their initial positions ($f(t)$) is in red. The particle trajectory resulting from advecting the points from their ending positions ($r(t)$) is in green. As both $f(t)$ and $r(t)$ have the same number of time steps, linear interpolation can be used via a function $\alpha(t)$ to compute the final trajectory (the black curve).

constraints. Thus, our linear system, as a block matrix, is as follows:

$$\begin{bmatrix} D \\ \overline{S} \end{bmatrix} \mathbf{x} = \begin{bmatrix} B \\ \overline{0} \end{bmatrix} \tag{7}$$

This system, by definition, is an overconstrained system as there exists at least n^2 smoothing constraints with only n^2 elements of $\mathbf{x}$ (thus $m > n^2$ in A and A is a $m \times n^2$ matrix). As the system is overconstrained, no solution exists. However, a solution that minimizes the error in the system can be solved independently for the x and y components of the vector field using the conjugate gradients method [46]. The resulting solution will be a vector field that can be applied to move the points of the group.

3.3 Advection of the Points

Given the vector field, we treat the points of the group as particles in a flow field and advect them. If we were to naively advect the data only from the initial position of the data points towards their final positions, there is no guarantee that the points would end up at their correct final positions as the ending positions have little influence on the vector field. Thus, we employ the following algorithm to ensure that points start at their initial positions, follow the transition defined by the user, and finally come to rest at the correct ending position in the transition:

1) *Advect all points from the initial positions towards the final positions using the vector field.*
2) *Multiply all vectors in the field by -1, reversing the field direction. Advect all points from the final positions towards the initial positions using the reverse vector field.*
3) *Interpolate the two particle trajectories to form an animated transition that starts and ends at the correct positions.*

Fig. 3 shows the results of this procedure on the individual points of a group. In this section, we describe how this procedure is carried out on all points of each group.

3.3.1 Forward and Reverse Advection

The points are advected in the vector field using the standard fourth-order Runge-Kutta method [47]. An alternative is to use *edge maps* [44] which can trace paths with bounded errors, but applying this method to our problem currently remains future work. In the forward advection process, we use the vector field computed in Section 3.2 to advect points.

This process creates good paths for the points near the starting point of the process, but low-quality paths as the particle gets further away. To compensate for this tendency, we advect the particles in reverse direction from the intended final positions of the data points towards the start positions as well. In order to advect in reverse, we first need to reverse the field by multiplying it everywhere by -1. Then, we apply the same process described above from the ending positions towards the starting positions. The final result is two particle traces for each point in the group as shown in Figs. 3a and 3b. It is easy to find that the paths generated from forward and reverse advection are good near the starting position and ending position respectively. Although the reverse vector field is used exactly, the paths for forward and reverse advection computed with Runge-Kutta method can incur error, resulting in the possible intersection of these paths.

In the next section, we describe how we can combine these two paths into a single animated transition with high quality.

3.3.2 Interpolation of Particle Trajectories

The simplest way to combine these trajectories is through linear interpolation. Fig. 3c shows the result for some data points in the vector field. The forward and reverse advection process have been run for exactly the same number of time steps $0 \ldots t_f$ by definition. Let $f(t)$ represent the location of a particle at time t during the forward advection process. Similarly, let $r(t)$ represent the location of the particle at time t during the reverse advection process. We define $\alpha(t) = t/t_f$ to be a linear weighting for the positions of the two particles. Given these definitions, we can

linearly interpolate the two particle positions to obtain the final position of the point in the transition (as shown in Fig. 4):

$$(1 - \alpha(t))f(t) + \alpha(t)r(t) \tag{8}$$

The value of $\alpha(t)$ ranges from 0 to 1. Intuitively, at the starting position, the dark-green solid point in Fig. 4, $\alpha(t) = 0$, $r(t)$ has a weight of 0, and $f(t)$ has a weight of 1. As the point progressively moves further and further away from the starting position, $\alpha(t)$ increases the weight of $r(t)$. As $t \to t_f$, the position of the point is nearly entirely determined by $r(t)$. This is exactly analogous to the formulation of Bézier curves in the Bernstein basis [48]. There, the Bernstein basis $\{\alpha \mapsto t, \alpha \mapsto 1-t\}$ is used to convert two Bézier curves of degree $n-1$ (one starting at point a, the other ending at point b) into one Bézier curve of degree n starting at point a and ending at point b. We use precisely the same formulation to guarantee a trajectory that starts at $f(0)$ and ends at $r(t)$, smoothly transforming between f and r in the process of going from 0 to t.

This interpolation function produces a smooth trajectory for the point that follows a non-linear path from the starting position to the ending position. Linear interpolation of $f(t)$ and $r(t)$ is just one way of combining these two trajectories. If slow-in slow-out is desired, a non-uniform weighting of $\alpha(t)$ can be used. In fact, nearly all of the temporal distortions described in Dragicevic et al. [5] can be defined using our method by modifying $\alpha(t)$ to have the desired weighting of $f(t)$ and $r(t)$.

The approach described here is quite generic and can be applied to a variety of datasets. Figs. 5 and 6 show the trajectories of animated transitions for both randomly-generated datasets and real datasets. These trajectories can reduce the possible crowding during the animated transition and are generally smooth. Even when some point groups are not well clustered (e.g., the group of green points in Fig. 6 are sparsely distributed), the proposed transition technique is still able to enhance coordinated motion of the point groups, reduce crowding and provide a smooth transition.

4 Evaluation

Following previous work on animated transition [4], [5], [6], we evaluate the proposed approach from the perspective of both quantitative metric evaluation and object-tracking based user study. We also conducted an informal user interview at first, evaluating the usability of manual transition design and guiding further technique design and evaluations.

4.1 Qualitative User Interview

In the development of the proposed transition design technique, we first conducted an informal qualitative user interview to confirm the factors affecting object tracking, evaluate the usability of manual transition design and collect feedback about the strategy of designing animated transitions. An early prototype of the system (Fig. 1) is used in the interview. We involved 4 participants in our interview and all of the participants are young researchers or engineers with normal vision.

During the interview with each participant, we first gave participants a basic introduction about animated transition design, showed several animated transitions with crowding and asked participants to track object identities. Then we explained how to use the user interface to design transitions, and asked participants to design transitions for two types of point groups: 5 groups (10 points per group) and 10 groups (5 points per group). After that, we interviewed participants to collect feedback about factors influencing object tracking, the usability of manual transition design and their strategy for manual sketching. Each interview took 40 to 50 minutes. Basic observations from the interview are summarized as follows:

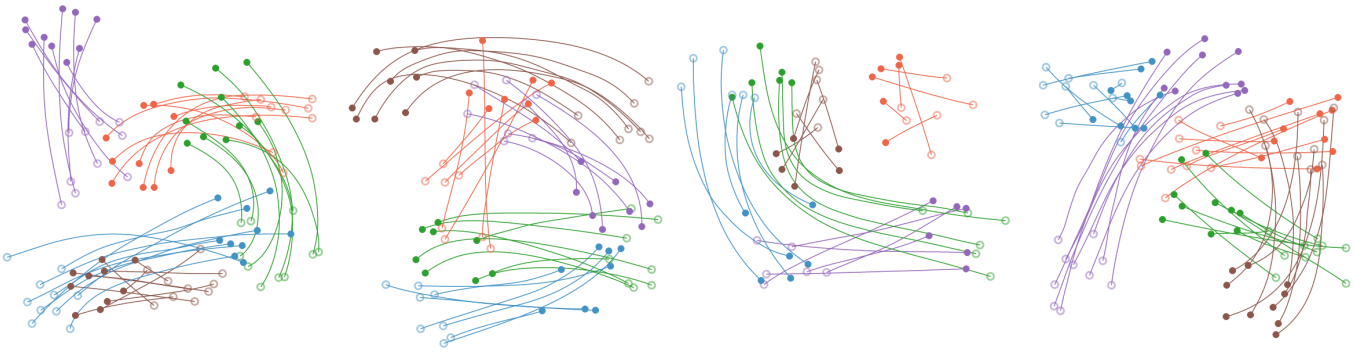
- *Crowding does affect tracking.* Almost all the participants agree that crowding during transition affects tracking accuracy, especially when several point groups cross at the same area simultaneously. Other factors, including speed, target number, and point sparsity also play an important role in the tracking.
- *Slightly curved trajectory is acceptable but highly curved trajectory is harmful.* Almost all of them agree that slightly curved trajectory will not affect their tracking accuracy, but when there is a sharp turn or the trajectory is highly curved, it will make tracking difficult.
- *Transition design strategy: plan the trajectory globally.* All the participants said that they attempt to do manual sketching and reduce crowding by relying on their intuitive estimation of time and space. One participant suggested: "the easier the first", i.e., sketching first for the groups with short transition distances or no inter-group crowding helps to find good transitions. In subsequent formal user study, we find that this strategy does help and adopt it to guide manual sketching.
- *User enjoys the flexibility of designing transition themselves,* as the desirable transition effect cannot always be achieved by existing automated animation techniques in visualization. Participants agree that the manual sketching is easy to learn.
- *More groups mean more difficulty for manual sketching.* Participants appreciate the flexibility of manual transition design. But at the same time, three of them feel that sketching trajectory for 10 groups is more difficult than 5 groups. There are two reasons for this: groups are hard to distinguish when many groups are overlaid on the screen, and optimal trajectory sketching is challenging for more groups, especially when they have long trajectories and serious crowding.

In summary, feedback from participants confirmed the important factors for animated transition design and provided us with some implications about the strategy and usability of manual transition design. These findings would be used to guide our further development of the proposed technique and its evaluation in subsequent sections.

4.2 Metric Evaluation

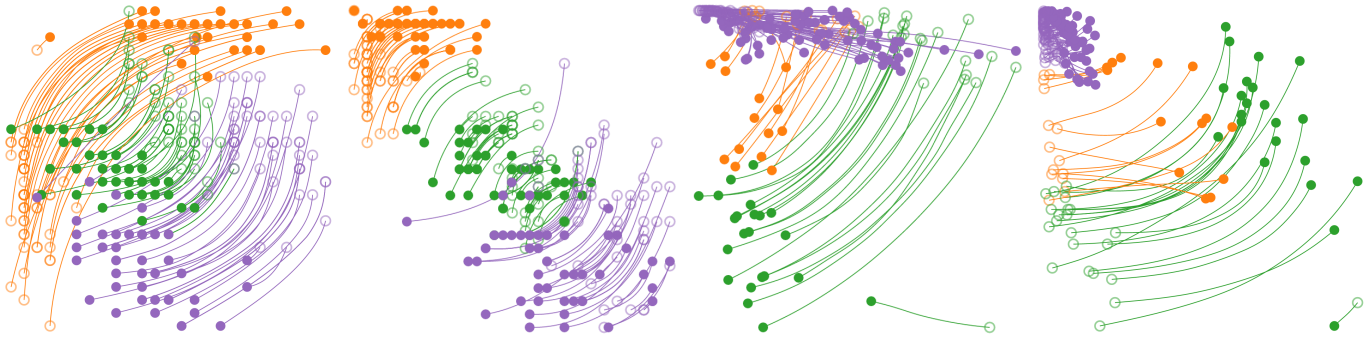
Quantitative metrics [4], [6] have been proposed to evaluate animated transitions from three perspectives: occlusion, dispersion, and deformation. **Occlusion** is a metric that quantifies the overlap between targets and distractors during the animated transition. Occlusion provides a way to quantify crowding [11], [36], [37], [38] and measures how often two objects pass close to each other, which may cause confusion in their identities. **Dispersion** is a metric that measures how compact the moving objects are to each other. By the Gestalt principle of Common fate [21], nearby objects that undergo similar operations tend to be perceived together. **Deformation** is a metric that measures the consistency of the shape of the object formed by the targets. Low deformation means that the shape of the object remains consistent, which Yantis [10] has demonstrated to be useful for object tracking.

The quantitative metrics used in this study are based on those proposed by Du et al. [6]. Du et al. [6] focus on membership in



(a) Example 1 **(b)** Example 2 **(c)** Example 3 **(d)** Example 4

Fig. 5: Trajectories of animated transitions for randomly-generated datasets. The paths of different groups of points are indicated with different colors.



(a) Example 1 of Iris Flower Dataset **(b)** Example 2 of Iris Flower Dataset **(c)** Example 1 of Breast Tissue Dataset **(d)** Example 2 of Breast Tissue Dataset

Fig. 6: Trajectories of animated transitions for real datasets such as Iris Flower Dataset [49] and Breast Tissue Dataset [50]. The paths of different groups of points are indicated with different colors.

a group of targets. However, in our work, we adapt these metrics to measure the ability to distinguish the individual identities of the targets within a group. Table 1 defines the terms used in our complexity metrics.

Notation	Definition
P	all the points in a dataset
$G \subset P$	the set of all targets
$p, q \in P$	two points in the dataset
T	the set of all time steps in the animation
t	a particular time step in the animation
K	the total number of groups in the dataset
C	a cluster of points with similar moving patterns

TABLE 1: Definition of terms used in the metric equations

We begin by describing the metrics used in our evaluation and then run a quantitative study on 50 transitions using these metrics.

4.2.1 Occlusion

Occlusion is a metric that essentially measures "crowding" or when the targets pass too close to each other and can have their identities confused [11]. Previous studies have shown that crowding impairs object tracking performance [38]. Franconeri et al. [37] claimed that crowding may be the most important factor affecting the accuracy of multiple object tracking. We propose three variants of occlusion to characterize the degree of crowding in animated transitions: **outer occlusion**, **inner occlusion**, and **overall occlusion**.

Outer occlusion quantifies the number of occlusions that occur between targets and distractors during the animated transition. This metric is exactly the same as *occlusion* as defined by Du et al. [6]:

$$outerOcclude(G) = \frac{1}{|T|} \sum_{t \in T} \frac{\sum_{p \in G, q \notin G} overlap(p, q, t)}{|G|(|P| - |G|)} \tag{9}$$

Where $overlap(p, q, t)$ is a binary function that returns 1 if there is an overlap between points p and q (i.e., $dist(p, q, t) \leq 2r$) and 0 otherwise. Outer occlusion has a maximum value of 1 when every pair of points overlaps in the animated transition.

As we would like to distinguish the individual identities of points within a group of targets, the overlap between two or more targets is also harmful. In this paper, we propose an **inner occlusion** to measure how often targets overlap in the animated transition:

$$innerOcclude(G) = \frac{1}{|T|} \sum_{t \in T} \frac{\sum_{p, q \in G, p \neq q} overlap(p, q, t)}{|G|(|G| - 1)} \tag{10}$$

Targets and distractors depend on the clustering of points in the dataset and the selection of targets. **Overall occlusion** does not distinguish between these two types of overlap. We propose

this metric to quantify any type of overlap between two points in the dataset:

$$overallOcclude(G) = \frac{1}{|T|}\sum_{t\in T}\frac{\sum_{p,q\in P, p\neq q} overlap(p,q,t)}{|P|(|P|-1)} \tag{11}$$

As in outer occlusion, inner and overall occlusion have a maximum value of 1 when all measured pairs overlap.

4.2.2 Dispersion

Dispersion has been proposed to capture the compactness of the moving group of targets. In our paper, we directly adopt the definition of Du et al. [6] and use it in our study. However, instead of computing dispersion for one group of points, we consider the dispersion of all K groups in the dataset. Therefore, the **average dispersion** of an animated transition is as follows:

$$avgDisperse(G) = \frac{1}{|K|}\sum_{1\leq i\leq K} disperse(C_i) \tag{12}$$

where $disperse()$ is the metric of Du et al. [6] and C_i is the i-th group of grouped points.

4.2.3 Deformation

Deformation measures the stability of the shape formed by the targets or the degree of coordinated motion. Yantis [10] provides evidence that we form perceptual groups, such as moving polygons, of these targets to track them, improving accuracy. We use the metric defined by Du et al. [6] directly and define **average deformation** for a dataset of K groups as follows:

$$avgDeform(G) = \frac{1}{|K|}\sum_{1\leq i\leq K} deform(C_i) \tag{13}$$

where $deform()$ is defined in [6].

4.2.4 Metric Experiment on Generated Data

Given these five metrics, we compare the performance of our animated transition design technique to both straight trajectories and bundled trajectories [6], since straight-line transition are commonly used and trajectory bundling transition is a representative technique that is also designed for point group transitions like the proposed technique. To comprehensively evaluate the proposed technique, both manual transition design and automated transition design are tested.

Dataset Generation: The testing datasets were generated in the following way. First, we randomly generated K virtual centers for each group and further generated random point positions using these centers. Points are guaranteed to be within a threshold radius of its virtual center with no overlap between points or groups. In each *randomly-generated dataset*, a point group is randomly selected and all the points within this group are chosen as the targets. As manual transition design requires a user to manually specify constraints, it is prohibitive to specify a very large number of datasets. We, therefore, generated 50 randomly-generated datasets: one half consisting of 80 points (8 groups with 10 points/group) and the other half consisting of 36 points (6 groups with 6 points per group). An author of the paper as an expert user sketched the manual transitions for this condition. During the sketching, the sketching strategies suggested by general users in Section 4.1 were followed, e.g., avoiding highly curved trajectory, sketching first for the point groups with short movements, etc..

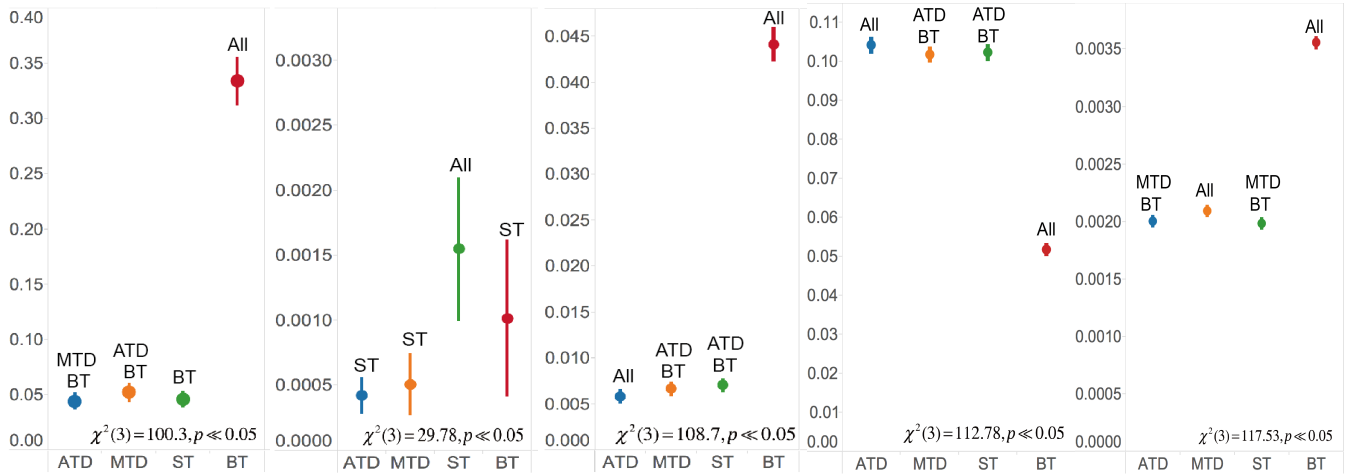
Procedures: We measured the metric scores on the 50 randomly-generated datasets. All the distances in the metric calculation are normalized by the size of the animation window. We first ran a Shapiro-Wilk test on each distribution to check for normality. Usually, the data was not normally distributed. Thus, we use a non-parametric Friedman with a Nemenyi-Damico-Wolfe-Dunn for post-hoc analysis. In the one case that all the distributions were normal (average dispersion), we checked our results with an ANOVA and Tukey post-hoc analysis and confirmed that the results did not change. All the tests were conducted with a standard significance level of $\alpha = 0.05$. Means and 95% confidence intervals for each transition technique are shown in Fig. 7. Statistically significant results are also indicated.

Results: Fig. 7 shows the metric evaluation results on the randomly-generated dataset: 1) For the occlusion metrics, the proposed transition design technique consistently has good performance. More specifically, Fig. 7a shows that automated transition design (ATD) and manual transition design (MTD) are comparable to straight trajectory (ST) in terms of inner occlusion, and all of them are significantly better than bundled trajectory (BT). For overall occlusion, ATD performs significantly better than all the other techniques. MTD is similar to ST and all of them are significantly better than BT (Fig. 7c). For outer occlusion, both ATD and MTD are significantly better than ST and have lower mean values than BT (Fig. 7b), where the means of ATD and MTD (about 0.0005) is only $1/3$ of ST's (about 0.0015) and $1/2$ of BT's (about 0.0010). 2) BT has the lowest dispersion, as it bundled trajectories of points in the same group at the cost of the high inner occlusion and deformation (Fig. 7d). 3) For deformation, BT performs worst, as its trajectories are highly curved. The deformation of our transition design is similar to or slightly more than ST (Fig. 7e).

4.2.5 Metric Experiment on Real Data

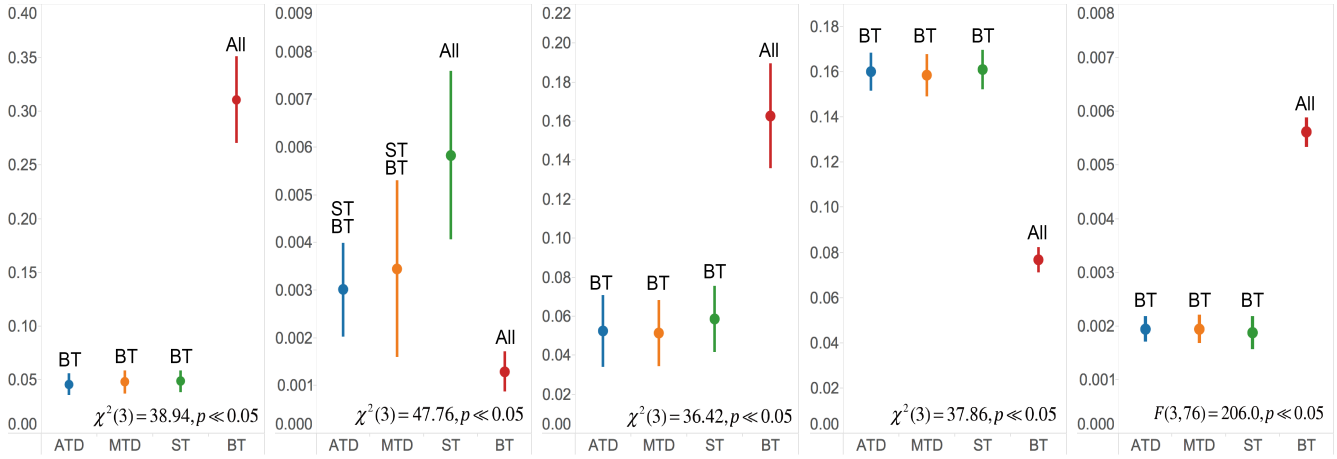
Apart from the randomly-generated dataset, we also conducted a metric experiment on *real datasets* in order to further evaluate the proposed technique. We chose Iris Flower Dataset [49] and Breast Tissue Dataset [50]. Iris Flower Dataset is a benchmark dataset commonly used in the visual analysis of multivariate data. It consists of 150 point samples from 3 species of iris flower (50 samples per species) and each species is regarded as one point group. Breast Tissue Dataset contains 106 samples of impedance measurements of breast tissue and we merged four of the original six tissue classes into one class (i.e., the non-fatty tissues [50] including carcinoma, fibro-adenoma, mastopathy and glandular tissues), as they share similar attributes. We regard the sample points within the same class as one group. Thus, the data is grouped into three groups, having 14, 22 and 70 point samples in each group respectively. For the two real datasets, we use the proposed technique to design the transitions between two scatterplots showing their different attributes. Also, since some scatterplots look similar and the transitions between them are inappropriate for comparing the effectiveness of transition techniques, we chose 10 transitions with a noticeable movement distance of point groups from each dataset by visual inspection, consisting of 20 transitions in total. Similar to the randomly-generated dataset, we also tested both manual transition design and automated transition design on the real datasets.

As in Section 4.2.4, we also conduct statistical testing on the real datasets. The Shapiro-Wilk tests show that the data was generally not normally distributed, so we ran a non-parametric Friedman



(a) Inner Occlusion **(b)** Outer Occlusion **(c)** Overall Occlusion **(d)** Average Dispersion **(e)** Average Deformation

Fig. 7: Metric evaluation on randomly-generated datasets: the metric means of the transition techniques, including Automated Transition Design (ATD), Manual Transition Design (MTD), Straight Trajectory (ST), Bundled Trajectory (BT). The error bars are 95% confidence intervals. The transition techniques that have statistically significant differences are indicated above the bars. Friedman test statistics are reported at the right bottom corner of each figure.



(a) Inner Occlusion **(b)** Outer Occlusion **(c)** Overall Occlusion **(d)** Average Dispersion **(e)** Average Deformation

Fig. 8: Metric evaluation on real datasets: the metric means of the transition techniques, including Automated Transition Design (ATD), Manual Transition Design (MTD), Straight Trajectory (ST), Bundled Trajectory (BT). The error bars are 95% confidence intervals. The transition techniques that have statistically significant differences are indicated above the bars. Friedman test statistics are reported at the right bottom corner of (a-d). For average deformation with normal distribution, the ANOVA test statistics are also reported.

with a Nemenyi-Damico-Wolfe-Dunn for post-hoc analysis and report the results in Fig. 8. For the average deformation, where all the distributions are normal, we used an ANOVA and Tukey post-hoc analysis and the result is reported in Fig. 8e.

Results: Fig. 8 shows the results on the real datasets and the results are almost the same: 1) For occlusion metrics, the experiment result is generally similar to the findings on the randomly-generated dataset. For example, the proposed technique (ATD and MTD) has similar or slightly lower inner and overall occlusions than straight line transition and all of them perform significantly better than bundled transition (Fig. 8a and Fig. 8c). For outer occlusion, our transition design technique still has significantly better performance than straight line transition, where the means of the proposed technique are about $1/2$ of the mean of straight line transition. However, bundled transition has the lowest outer occlusion (Fig. 8b), which differs from our results on randomly-generated datasets (Fig. 7b). 2) For the average dispersion and deformation (Figs. 8d and 8e), we have almost the same results as the randomly-generated datasets. Our transition design technique and straight line transition perform similar to each other, while bundled transition has the least dispersion and highest deformation.

4.2.6 *Discussion*

Given the results of our metric experiments, we discuss the advantages and disadvantages of our transition design technique.

Balance in reducing occlusions. For outer occlusion, Figs. 7b and 8b show that our transition design technique (both ATD and MTD) has significantly fewer outer occlusions than straight line transition, as the non-linear paths can move around groups.

Dataset	Attributes		Time Cost (seconds)			
	Groups	Points	Path Generation	VF Computation	Point Advection	All
Generated Dataset 1	8	80	0.143	0.448	0.102	0.693
Generated Dataset 2	6	36	0.125	0.382	0.048	0.556
Iris Flower [49]	3	150	0.026	0.132	0.130	0.288
Breast Tissue [50]	3	106	0.023	0.143	0.104	0.270

TABLE 2: Average running time of the proposed technique when testing on both the randomly-generated datasets and real datasets.

This finding is consistently observed in both metric experiments. When compared with bundled transition, our proposed technique outperforms bundled transition on the randomly-generated dataset, but performs worse than bundled transition on the real datasets. We conjecture that such a difference may be related to the number of point groups, as the real datasets have fewer point groups than the randomly-generated datasets (3 vs. 6 or 8) and a small number of groups can reduce the chance of crowding between the bundled trajectories. Also, this reduction of outer occlusion in the bundled transition is likely at the cost of inner occlusion and deformation. For inner occlusion (Figs. 7a and 8a) and overall occlusion (Figs. 7c and 8c), ATD and MTD perform similarly to straight line transition and all of them have significantly better performance than bundled transitions. Overall, our transition design technique performs better and strikes a good balance in reducing occlusions when comprehensively considering all the three occlusion metrics. This provides evidence that our technique incurs less crowding, meaning that the chance of confusing the identities of similarly looking objects in the transition is lower.

Preservation of low deformation. As shown in Figs. 7e and 8e, though our transition design technique also uses non-linear paths like bundled transition, both automated transition design and manual transition design have low deformation like straight-line transition and perform significantly better than bundled transition. Such a finding is consistent in both metric experiments. Thus, our technique preserves the shape of moving groups better than bundled trajectories, helping users keep track of the group of points, as they move. This result confirms, from a metric sense, our second desired property.

Relatively high dispersion. Figs. 7d and 8d show that both manual transition design and automated transition design perform similarly to straight line trajectories in terms of dispersion. All of them were significantly less compact than bundled transition. Bundled transition is the most compact as the moving points in a group are bundled very close together. When dispersion is low, points act like a single object, making group identities easier to track [6]. Thus, bundled trajectories have an advantage when only group identities are required. However, low dispersion is often at the expense of crowding and deformation, which can make the individual point identities difficult to track and will be shown in the tracking accuracy results in Section 4.3.

In conclusion, animated transition design strikes a balance that allows for low crowding and low deformation transitions, supporting the tracking of groups of points through the scene. Compared with straight line transition, the proposed technique preserves the advantages of straight line transitions, including low deformation, low inner occlusion and low overall occlusion, but our transition technique also has significantly fewer outer occlusions. Compared with bundled transition, the metric evaluation also provides evidence that our technique has significantly fewer inner and overall occlusions and less deformation. Since points of the same group are using the same vector field, the proposed technique also has the benefit of coordinated motion like bundled transition.

When comparing automated transition design with manual transition design, Figs. 7 and 8 show that automated transition design has almost the same or slightly better performance than manual transition design for the occlusion metrics. These findings are consistent on both randomly-generated dataset and real datasets. Manual and automated transition design also have similar performance on dispersion and deformation. It makes sense, since the same principles in the path generation for each point group are used for both manual transition design and automated transition design. Therefore, the metric evaluation demonstrates that automated transition design is a comparable alternative to manual transition design in terms of performance, which would be helpful when manual sketching is not preferred by users.

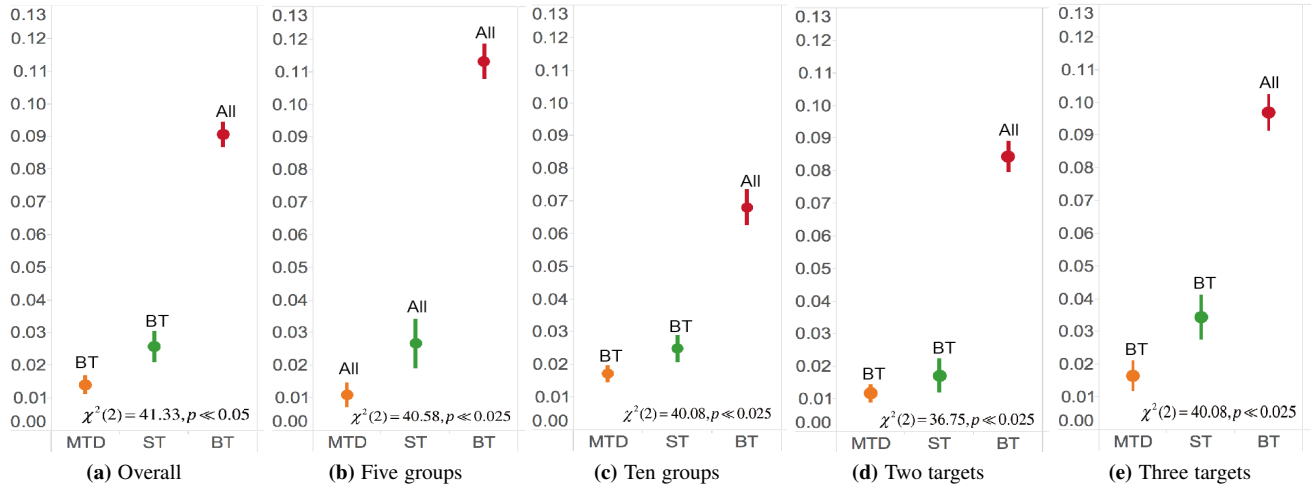
Considering the choice of dataset, Figs. 7 and 8 demonstrate that almost all the performances of all the transition techniques are consistent on both randomly-generated dataset and real datasets. The only exception is that bundled transition has relatively fewer outer occlusions on the real datasets than it has on randomly-generated dataset (Figs. 8c and 7c), when compared with other transition techniques. It may be due to the fewer point groups of the real datasets, reducing the possibility of crowding.

The running time of the proposed technique is also evaluated, as shown in Table 2. We recorded the average running time of each algorithm component when testing on both the randomly-generated datasets and real datasets. It was implemented in Javascript and run on a Mac laptop with an Intel Core i5 CPU and 8GB of memory. Only automated transition design is tested, since manual sketching is required for manual transition design. Table 2 shows that the average running time of the proposed technique on each dataset is less than 1 second, which is fast enough for designing animated transitions. Among the three stages, vector field computation is the most costly. With an increasing number of groups, more time is needed as more paths and vector fields need to be generated.

4.3 User Study

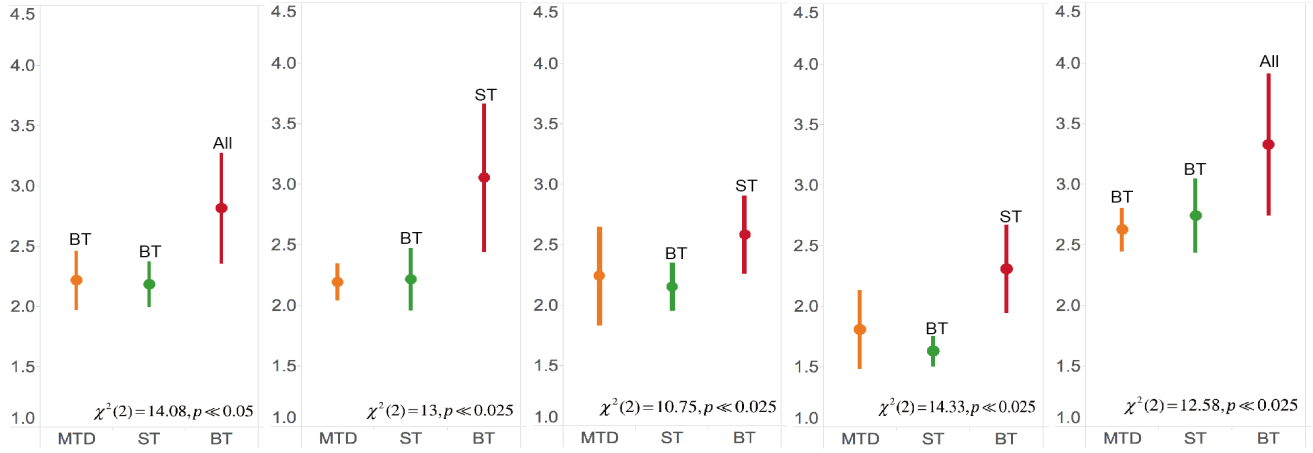
To further investigate the effectiveness of the proposed animated transition technique, we performed a formal user study. The formal user study also compared our animated transition design technique to both trajectory bundling and straight line transitions, because they are either widely used in transitions or a representative transition technique for point groups. Since the above metric evaluation indicates that our proposed technique has an advantage of reducing outer occlusions, the user study here focuses on testing its effectiveness in helping object tracking in transitions of high occlusion.

In this user study, only manual transition design is tested, because of the similar performance between automated transi-



(a) Overall **(b)** Five groups **(c)** Ten groups **(d)** Two targets **(e)** Three targets

Fig. 9: The means of tracking accuracy as measured by the normalized distance/error between the correct and entered answer. The evaluated techniques include Manual Transition Design (MTD), Straight Trajectory (ST), Bundled Trajectory (BT). The error bars are 95% confidence intervals. Techniques with a pairwise statistically significant difference are listed above each bar. Friedman test statistics are reported at the right bottom corner of each figure.



(a) Overall **(b)** Five groups **(c)** Ten groups **(d)** Two targets **(e)** Three targets

Fig. 10: The mean response time (seconds) recorded in the user study. The error bars are 95% confidence intervals. The evaluated techniques include Manual Transition Design (MTD), Straight Trajectory (ST), Bundled Trajectory (BT). Techniques with statistically significant differences are listed above the bars. Friedman test statistics are reported at the right bottom corner of each figure.

tion design and manual transition design (Section 4.2). Some guidelines based on the observations in Section 4.1 are followed to generate path for each point group. For example, *draw short transition first*, as it can be helpful to globally plan the group path and avoid occlusion. In addition, *make the manual sketching as smooth as possible and avoid highly curved trajectories*, since it shows that highly curved trajectory can be misleading for object tracking. Randomly-generated datasets are used in the user study. An author of this paper manually sketched the transitions for this user study. Because the adopted sketching strategies are exactly the same, the generated manual sketching results should be comparable to what other users would draw.

4.3.1 Study Design

We adopted a within subject design. Overall, the study considered 3 transition techniques (straight line, bundled, transition design) $\times$ 2 numbers of targets to be tracked (high: 3, low: 2) $\times$ 2 different group sizes (high: 10 points/group and 5 groups, low: 5 points/group and 10 groups) $\times$ 5 repetitions. Thus, each participant undertook 60 trials in our experiment. The duration of the transition is 2 seconds. These settings are based on our observations in the informal qualitative user interview, and we want to make sure that the task we used is neither too easy nor difficult.

In each trial, we asked participants to track the precise identity of two or three randomly-chosen targets within a single group of the animated transition. Targets were visually indicated with different colors and alphabetic labels before the transition began. When the transition started, all identifying features were removed from the targets and the color of them was set to black, the same color for all other points in the transition. When the transition concluded, participants were asked to select the targets by clicking

on them in alphabetic order. The time it took the participants to answer and the distance between the entered and correct answer were measured. As the three techniques compared in the experiment are quite different, we separated them into three blocks of twenty trials each. These blocks were counterbalanced across participants through Latin squares. Within each block, the order of the twenty trials was randomized to avoid the learning effects that could arise when the same dataset is presented repeatedly.

Dataset Generation: Similar to Section 4.2.4, we also randomly generated K virtual group centers first and further generated random positions for points of each group. In order to generate our five repetitions, we randomly generated 1,000 datasets for each of the five and ten point conditions. Ten of these datasets were randomly selected from the top 25% of datasets with high *average outer occlusion* under the straight line condition. This *average outer occlusion* is computed by calculating the average outer occlusion with each point group as the targets. As we conjecture a primary advantage of our animated transition design technique is avoiding outer occlusions, the study was designed to test if this was helpful for users in such kind of situations. For each of the five repetitions per target level, a set of points (2 or 3 points) within the same point group were randomly selected as the targets, and we considered three rotations of this dataset randomized across techniques. The measure of tracking correctness is the distance between the selected and correct answer instead of binary correct-or-wrong counting, as the distance can delineate more details of object tracking accuracy.

Apparatus and Configuration: We conducted the user study on a laptop connected to a 23-inch display with a 1920×1080 pixels resolution and 60 Hz refresh rate. All transitions are shown in a square window of 640×640 pixels with a white background. Targets and other points in the scene are black circles of nine pixels in radius. Groups of ten points have a group radius of 180 pixels while groups of five points have a radius of 120 pixels. All measurements of the distance between the correct and entered answers are normalized by the size of the window.

Participants: Overall, 24 participants (10 female, age: 22 to 29 (mean: 25.6)) took part in our experiment. Participants had a computer science background and were recruited from our university.

Procedure: We first explained the user study system and tasks to the participants. Each participant performed a training session that included all three techniques, where participants were given the correct answer once they finished each object tracking task and were provided an opportunity to ask questions. Between each block of twenty trials, participants could take a short break. After the completion of the user study, a quick interview took place where we asked the participants about how they think of the three different animated transitions and what are their advantages and disadvantages.

4.3.2 Results

Fig. 9 shows the means and their 95% confidence intervals of normalized distance between the entered and correct answers in the experiment. We first show the overall tracking accuracy (Fig. 9a) and subsequently divide by the number of groups (Figs. 9b and 9c) and the number of targets (Figs. 9d and 9e). The same analysis is performed with respect to response time (Fig. 10).

We first ran a Shapiro-Wilk test on each distribution to check for normality and found that the data is not usually normally distributed. Therefore, we further use a Friedman test with a Nemenyi-Damico-Wolfe-Dunn test for post-hoc analysis. For overall accuracy and time, a standard significance level of $\alpha = 0.05$ was chosen. When dividing by the group or target number, we apply a Bonferroni correction, reducing the significance level to $\alpha = 0.025$. Significant differences between the techniques are marked above each bar (Figs. 9 and 10).

From Fig. 9, we found that transition design performs best when comparing the means of object tracking accuracy for transitions, as measured by the distance. Such an optimality is consistent when analyzing the results from a perspective of different point groups or targets. For bundled trajectories, they have much worse object tracking accuracy than the proposed transition technique with statistical significance. The reason for it is that bundled trajectories suffer from high crowding and large deformation, which has been confirmed in Section 4.2. Straight line transitions have better tracking accuracy than bundled trajectories, but our proposed technique has better tracking accuracy than straight line transitions (though not significantly so). For instance, transition design improves the overall tracking performance on datasets (Figs 9a), where the mean error of MTD (around 0.015) is only about 60% of ST's (around 0.025). When divided by the group number, transition design has an advantage over straight line transitions. When there are five groups (Fig. 9b), transition design significantly outperforms straight line transitions and the mean error of MTD (around 0.01) is only about 40% of ST's (around 0.025). As for ten groups, transition design also has better mean accuracy than straight line transition (Fig. 9c). When divided by the target number, the same observation still holds. The mean tracking error of MTD is less than that of ST (Figs. 9d) for two targets. With a greater number of targets, such an improvement becomes greater (Fig. 9e) and the error of MTD is only about 50% of ST's. The possible reason for it is that crowding becomes a more serious issue in straight line transitions when more objects need to be tracked, but transition design can consistently avoid crowding.

Fig. 10 shows that participants take statistically significantly more time to select their answers with bundled transitions, which is consistent with object tracking accuracy (Fig. 9) and our previous metric evaluation (Figs. 7 and 8). Bundled trajectories, which suffer from high inner occlusion, are quite difficult to use for this task and thus require higher response time. On the contrary, though transition design also uses non-linear paths like bundled trajectories, its response time is still comparable to or only slightly more than straight line transitions, which means using non-linear paths in transition design is not more difficult to use. In the case with more targets (Fig. 10e), transition design even takes less response time.

In our post-study interview, almost all participants mentioned that high occlusion was noted for the bundled trajectory condition due to high inner occlusion, but as it greatly reduces dispersion, tracking each point group without point identity required is easier. For the straight line condition, participants said that occlusion was once again the major factor, especially when it occurred around targets. Most participants confirmed that transition design reduces occlusion between groups and helps them track object identities accurately. The possible drawback of the technique is that it would be not as predictable as straight lines if the trajectory is highly curved. However, similar to observations in our previous informal user interview, most participants of the user study did not find this a major issue, as trajectories were generally smooth and well designed.

5 Discussion and Conclusion

In this paper, we introduced a vector-field based technique for animated transition design of clustered points. This approach offers users with great flexibility by providing users with both manual and automated transition design. The generated transition can avoid occlusions while maintaining coordinated motions of clustered points. These objectives are accomplished by leveraging vector field design methods, heavily used in the area of scientific visualization, to design and utilize a vector field that can be further used to advect the points. Linear interpolation between forward and reverse advection from the starting and ending positions of the points allows the technique to produce a smooth, non-linear transition for all points in the scene.

We first conducted a user interview to confirm the usability of the proposed technique. Several findings from it are also used for guiding the subsequent design and evaluation of the proposed technique. Then a metric evaluation, consistent with previous work, was performed on both the randomly-generated datasets and real datasets. It not only demonstrates the similar performance between automated and manual transition design but also provides some evidence that the proposed animated transition technique can avoid occlusions while maintaining low deformation. The user study further demonstrates the effectiveness of the proposed transition technique in improving object tracking accuracy when high occlusion exists. It consistently outperforms both straight line transitions and bundled trajectories in tracking accuracy, showing its advantages of avoiding occlusion and striking a balance between deformation and coordinated motion. Response time analysis indicates that the non-linear path of our technique does not bring additional difficulty to users of the technique.

However, our technique is not without limitations. First, the proposed technique is able to reduce crowding and occlusion in animated transitions while preserving coordinated motion. But this crowding reduction is not unlimited, when there are an extremely large number of points on the limited screen, the proposed technique may also fail to reduce crowding. Secondly, as in bundled transition [6], the proposed technique needs the predefined clustering of points where a group of points are spatially close and move in a similar direction in the transition. A clustering of this type is required for the technique to be effective. Clusters without a similar spatial position and motion may not perform well. Finally, we follow previous work [4], [5], [6] and conducted our experiments based on scatterplots. Applying this technique in other animated transition design scenarios, such as dynamic graphs, has not been fully explored.

In future work, we will explore the application of the proposed technique in other transition scenarios (e.g., dynamic graph animation and other graphical animations). It would also be interesting to extend the proposed technique to show the dynamic movement of people in the visual analytics of sparse trajectory data such as telco big data [51]. Furthermore, we would like to investigate methods for relaxing our clustering constraints and conduct more user studies with general users being involved in designing animated transitions to further evaluate the effectiveness of our technique.

Acknowledgments

The authors would like to thank all the participants for their participation in the user studies and interviews, Dr. Conglei Shi for the constructive discussions, and all the anonymous reviewers for their valuable comments and suggestions. This work is partially supported by RGC GRF 16241916, AT&T Labs and NSF under award III-1513651.

References

[1] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale, "A descriptive framework for temporal data visualizations based on generalized space-time cubes," *Computer Graphics Forum*, 2016.

[2] D. Fisher, "Animation for visualization: opportunities and drawbacks," in *Beautiful visualization: looking at data through the eyes of experts*, J. Steele and N. Iliinsky, Eds. O'Reilly Media, Inc., 2010, ch. 19, pp. 329–352.

[3] J. Heer and G. Robertson, "Animated transitions in statistical data graphics," *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, vol. 13, no. 6, pp. 1240–1247, 2007.

[4] F. Chevalier, P. Dragicevic, and S. Franconeri, "The not-so-staggering effect of staggered animated transitions on visual tracking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2241–2250, 2014.

[5] P. Dragicevic, A. Bezerianos, W. Javed, N. Elmqvist, and J. Fekete, "Temporal distortion for animated transitions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 2009–2018.

[6] F. Du, N. Cao, J. Zhao, and Y. Lin, "Trajectory bundling for animated transitions," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 289–298.

[7] N. Elmqvist, P. Dragicevic, and J. Fekete, "Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1539–1148, 2008.

[8] D. Archambault and H. C. Purchase, "Can animation support the visualization of dynamic graphs?" *Information Sciences*, vol. 330, pp. 495–509, 2016.

[9] B. Tversky, J. B. Morrison, and M. Betrancourt, "Animation: can it facilitate?" *International Journal of Human-computer Studies*, vol. 57, no. 4, pp. 247–262, 2002.

[10] S. Yantis, "Multielement visual tracking: Attention and perceptual organization," *Cognitive Psychology*, vol. 24, no. 3, pp. 295–340, 1992.

[11] S. L. Franconeri, J. Y. Lin, J. T. Enns, Z. W. Pylyshyn, and B. Fisher, "Evidence against a speed limit in multiple-object tracking," *Psychonomic Bulletin & Review*, vol. 15, no. 4, pp. 802–808, 2008.

[12] C. Friedrich and M. E. Houle, "Graph drawing in motion II," in *Proceedings of the 9th International Symposium on Graph Drawing*. Springer, 2001, pp. 220–231.

[13] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst, "Animated exploration of dynamic graphs with radial layout," in *Proceedings of the IEEE Symposium on Information Visualization*, 2001, pp. 43–50.

[14] J. Lasseter, "Principles of traditional animation applied to 3D computer animation," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, vol. 21, no. 4, 1987, pp. 35–44.

[15] S. L. Franconeri, Z. W. Pylyshyn, and B. J. Scholl, "A simple proximity heuristic allows tracking of multiple objects through occlusion," *Attention, Perception, & Psychophysics*, vol. 74, no. 4, pp. 691–702, 2012.

[16] P. Jolicoeur, S. Ullman, and M. Mackay, "Curve tracing: A possible basic operation in the perception of spatial relations," *Memory & Cognition*, vol. 14, no. 2, pp. 129–140, 1986.

[17] Z. W. Pylyshyn and R. W. Storm, "Tracking multiple independent targets: Evidence for a parallel tracking mechanism," *Spatial Vision*, vol. 3, no. 3, pp. 179–197, 1988.

[18] G. Chen, K. Mischaikow, R. S. Laramee, P. Pilarczyk, and E. Zhang, "Vector field editing and periodic orbit extraction using morse decomposition," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 769–785, 2007.

[19] H. Theisel, "Designing 2D vector fields of arbitrary topology," *Computer Graphics Forum*, vol. 21, no. 3, pp. 595–604, 2002.

[20] E. Zhang, K. Mischaikow, and G. Turk, "Vector field design on surfaces," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 4, pp. 1294–1326, 2006.

[21] S. E. Palmer, *Vision science: Photons to phenomenology*. MIT press Cambridge, MA, 1999.

[22] M. Wattenberg and J. Kriss, "Designing for social data analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 549–557, 2006.

[23] P. Baudisch, D. Tan, M. Collomb, D. Robbins, K. Hinckley, M. Agrawala, S. Zhao, and G. Ramos, "Phosphor: explaining transitions in the user interface using afterglow effects," in *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, 2006, pp. 169–178.

[24] B. S. Hasler, B. Kersten, and J. Sweller, "Learner control, cognitive load and instructional animation," *Applied Cognitive Psychology*, vol. 21, no. 6, pp. 713–729, 2007.

[25] P. Dragicevic, S. Huot, and F. Chevalier, "Gliimpse: Animating from markup code to rendered documents and vice versa," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, 2011, pp. 257–262.

[26] C. Schlienger, P. Dragicevic, C. Ollagnon, and S. Chatty, "Les transitions visuelles différenciées: Principes et applications," in *Proceedings of the 18th Conference on L'Interaction Homme-Machine*, 2006, pp. 59–66.

[27] J. Popović, S. M. Seitz, and M. Erdmann, "Motion sketching for control of rigid-body simulations," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 4, pp. 1034–1054, 2003.

[28] M. Gleicher, "Motion path editing," in *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, 2001, pp. 195–202.

[29] R. M. Baecker, "Picture-driven animation," in *Proceedings of the May 14-16, 1969, Spring Joint Computer Conference*, 1969, pp. 273–288.

[30] R. C. Davis, B. Colwell, and J. A. Landay, "K-sketch: a 'kinetic' sketch pad for novice animators," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 413–422.

[31] R. H. Kazi, F. Chevalier, T. Grossman, S. Zhao, and G. Fitzmaurice, "Draco: bringing life to illustrations with kinetic textures," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, pp. 351–360.

[32] C. S. Feria, "Speed has an effect on multiple-object tracking independently of the number of close encounters between targets and distractors," *Attention, Perception, & Psychophysics*, vol. 75, no. 1, pp. 53–67, 2013.

[33] G. Liu, E. L. Austen, K. S. Booth, B. D. Fisher, R. Argue, M. I. Rempel, and J. T. Enns, "Multiple-object tracking is based on scene, not retinal, coordinates." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 31, no. 2, p. 235, 2005.

[34] T. S. Horowitz, S. B. Klieger, D. E. Fencsik, K. K. Yang, G. A. Alvarez, and J. M. Wolfe, "Tracking unique objects," *Perception & Psychophysics*, vol. 69, no. 2, pp. 172–184, 2007.

[35] Z. W. Pylyshyn, "Some puzzling findings in multiple object tracking (MOT): II. inhibition of moving nontargets," *Visual Cognition*, vol. 14, no. 2, pp. 175–198, 2006.

[36] G. A. Alvarez and S. L. Franconeri, "How many objects can you track? evidence for a resource-limited attentive tracking mechanism," *Journal of Vision*, vol. 7, no. 13, pp. 14–14, 2007.

[37] S. Franconeri, S. Jonathan, and J. Scimeca, "Tracking multiple objects is limited only by object spacing, not by speed, time, or capacity," *Psychological Science*, vol. 21, no. 7, pp. 920–925, 2010.

[38] W. M. Shim, G. A. Alvarez, and Y. V. Jiang, "Spatial separation between targets constrains maintenance of attention on multiple objects," *Psychonomic Bulletin & Review*, vol. 15, no. 2, pp. 390–397, 2008.

[39] B. J. Scholl and Z. W. Pylyshyn, "Tracking multiple items through occlusion: Clues to visual objecthood," *Cognitive Psychology*, vol. 38, no. 2, pp. 259–290, 1999.

[40] F. de Goes, M. Desbrun, and Y. Tong, "Vector field processing on triangle meshes," in *SIGGRAPH Asia 2015 Courses*, 2015, p. 17.

[41] M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe, "Design of tangent vector fields," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, 2007.

[42] G. Chen, V. Kwatra, L.-Y. Wei, C. D. Hansen, and E. Zhang, "Design of 2D time-varying vector fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 10, pp. 1717–1730, 2012.

[43] N. Ferreira, J. T. Klosowski, C. E. Scheidegger, and C. T. Silva, "Vector field K-means: Clustering trajectories by fitting multiple vector fields," in *Proceedings of the 15th Eurographics Conference on Visualization*, vol. 32, no. 3. Wiley Online Library, 2013, pp. 201–210.

[44] H. Bhatia, S. Jadhav, P.-T. Bremer, G. Chen, J. A. Levine, L. G. Nonato, and V. Pascucci, "Flow visualization with quantified spatial and temporal errors using edge maps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1383–1396, 2012.

[45] P. Simonetto, D. Archambault, and C. Scheidegger, "A simple approach for boundary improvement of Euler diagrams," *IEEE Transactions on Visualization and Computer Graphics (InfoVis 2015)*, vol. 22, no. 1, pp. 678–687, 2016.

[46] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Pittsburgh, PA, USA, Tech. Rep., 1994.

[47] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. New York, NY, USA: Cambridge University Press, 2007.

[48] G. E. Farin, *Curves and surfaces for CAGD: a practical guide*, 5th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.

[49] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

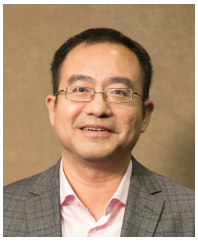
[50] J. E. Da Silva, J. M. De Sá, and J. Jossinet, "Classification of breast tissue by electrical impedance spectroscopy," *Medical and Biological Engineering and Computing*, vol. 38, no. 1, pp. 26–30, 2000.

[51] Y. Zheng, W. Wu, H. Zeng, N. Cao, H. Qu, M. Yuan, J. Zeng, and L. M. Ni, "Telcoflow: Visual exploration of collective behaviors based on telco data," in *Proceedings of 2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 843–852.

Yong Wang is currently a PhD candidate in the Department of Computer Science and Technology at the Hong Kong University of Science and Technology (HKUST). He received his B.E degree in Automation from Harbin Institute of Technology, China in 2011 and M.E. degree in Pattern Recognition and Intelligent system from Huazhong University of Science and Technology, China in 2014. His research interests include data visualization, visual analytics and image processing.

Daniel Archambault received his PhD from the University of British Columbia in 2008 and is a Senior Lecturer at Swansea University. His principal area of research has been network visualization often for social networks. In particular, he has focused on the development and evaluation of techniques for visualizing dynamic graphs and scalable graph visualizations.

Carlos Scheidegger is an assistant professor in the Department of Computer Science at the University of Arizona since 2014. He holds a PhD in Computing from the University of Utah, where he worked on software infrastructure for scientific collaboration. His current research interests are in large-scale data analysis, information visualization and, more broadly, what happens "when people meet data". His honors include multiple best paper awards and nominations, and an IBM student fellowship.



Huamin Qu is a professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His main research interests are in visualization and computer graphics, with focuses on urban informatics, social network analysis, e-learning, and text visualization. He obtained a BS in Mathematics from Xi'an Jiaotong University, China, an MS and a PhD in Computer Science from the Stony Brook University. For more information, please visit http://www.huamin.org.