A Deformation Framework for Focus+Context Flow Visualization

Jun Tao, *Student Member, IEEE,* Chaoli Wang, *Member, IEEE,*
Ching-Kuang Shene, *Member, IEEE Computer Society*, and Seung Hyun Kim

Abstract—Striking a careful balance among coverage, occlusion and complexity is a resounding theme in the visual understanding of large and complex three-dimensional flow fields. In this paper, we present a novel deformation framework for focus+context streamline visualization that reduces occlusion and clutter around the focal regions while compacting the context region in a full view. Unlike existing techniques that vary streamline densities, we advocate a different approach that manipulates streamline positions. This is achieved by partitioning the flow field's volume space into blocks and deforming the blocks to guide streamline repositioning. We formulate block expansion and block smoothing into energy terms and solve for a deformed grid that minimizes the objective function under the volume boundary and edge flipping constraints. Leveraging a GPU linear system solver, we demonstrate interactive focus+context visualization with 3D flow field data of various characteristics. Compared to the fisheye focus+context technique, our method can magnify multiple streamlines of focus in different regions simultaneously while minimizing the distortion through optimized deformation. Both automatic and manual feature specifications are provided for flexible focus selection and effective visualization.

Keywords—Flow visualization, focus+context visualization, optimized deformation.

1 Introduction

For large and complex 3D flow fields, the presence of potentially dense particle traces or field-lines easily leads to visual occlusion and clutter in rendering. This issue poses a fundamental challenge to generating informative images for effective understanding. Simply filtering streamlines for feature highlighting reduces occlusion and clutter. However, maintaining a good balance of densities among different flow regions could be tricky as many streamlines may pass regions with various degrees of importance. Showing only portions of streamlines may solve the problem, but the continuity of streamlines is broken. Rather than varying streamline *densities* through seed placement or streamline selection, we seek a different approach that manipulates streamline *positions*. The benefit of doing so is that since the content of streamlines remains unchanged, the viewer's attention will be readily directed to the deformed streamlines for purposeful observation. In this paper, we present a novel focus+context (F+C) solution for effective clutter reduction and visual exploration. F+C visualization stems from the need to show both overview (context) and detail information (focus) simultaneously within a limited display area. It has long been used in visualization to show details of selected features together with the context in a full view.

Instead of deforming each individual streamline directly, we advocate a grid-based space deformation scheme to maintain the coherency among deformed streamlines. By applying a coarse grid over the domain, we partition the domain into blocks and derive each individual block's importance. We then formulate block expansion and block smoothing into energy terms and search for a deformed grid that minimizes the objective energy function. With the fixed volume space, important blocks are magnified while unimportant ones are shrunk. The deformed grid guides streamline repositioning. As a result, streamline segments passing important blocks will enjoy larger space and thus get less cluttered while other streamline segments passing unimportant blocks will occupy less space and thus get more compact, achieving F+C visualization. At first glance, this idea of streamline deformation seems to go against the principle of maintaining the accuracy of streamlines. Nevertheless, leveraging the GPU we are able to perform the deformation and recovery process in real time. The user can therefore intuitively understand how the deformation or recovery progresses without losing the reference to the original representation.

• *J. Tao, C. Wang, and C.-K. Shene are with the Department of Computer Science, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931. E-mail: {junt, chaoliw, shene}@mtu.edu.*
S. H. Kim is with the Department of Mechanical Engineering-Engineering Mechanics, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931. E-mail: shkim@mtu.edu.

2 Related Work

Examples of F+C visualization include the fisheye view [6], [14], [7] for text, image and graph visualization, and the magnification lens [10], [18] and conformal magnifier [22] for image and volume visualization. Unlike fisheye view or magnification lens, our method leverages an optimized deformation to minimize the global distortion, which can magnify multiple streamlines in different focus regions simultaneously.

In flow visualization, Fuhrmann and Gröller [5] presented magic lenses and magic boxes to examine the region of interest with greater detail by showing denser streamlines. This technique was later extended to magic volumes of varying focal regions such as cubes, prisms and spheres [13]. Laramee et al.

[11] leveraged feature-based techniques [4] to extract interesting flow regions, such as stagnant flow, reverse-longitudinal flow and regions of high pressure gradient, as the focus. They achieved F+C rendering through interactive thresholding that reduces flow complexity and resulting visualization. Correa et al. [3] introduced physical and optical operators to intuitively visualize the internal 3D flow through illustrative deformation. By cutting along flow traces, they allowed a clear observation of the internal 3D flow through optical transformation and elastic deformation. To explore blood flow in cerebral aneurysms, Gasteiger et al. [8] proposed an interactive 2D widget for flexible visual filtering and visualization of the F+C pairs (i.e., relevant hemodynamic attributes). Their widget supports local probing and conveys changes over time for the lens region. van der Zwan et al. [17] modeled several visualizations of a data set as abstractions which represent the information on different levels of details. Given the user-selected level of focus, F+C visualization is achieved by manipulating the transparency of each abstraction. All these methods, however, do not shrink the context of the flow field while magnifying a specific focal region in order to best utilize the available volume space. In contrast, we devise a continuous deformation solution based on an energy optimization model to achieve F+C streamline visualization.

Closely related to our work are the F+C techniques presented by Wang et al. [19], [20] for surface models and volumetric data. In [19], they presented a F+C technique for surface models that magnifies a region of interest for closer examination while deforming other regions without perceivable distortion. In [20], a similar technique was presented for volumetric data for feature-preserving data reduction and F+C visualization. To the best of our knowledge, the full benefits of such a deformation framework have not been demonstrated for interaction and visualization of flow fields. Several challenges exist when applying this grid-based deformation framework to flow fields. Unlike surface models and volumetric data where the visualization is 2D or 3D continuous, streamlines are only 1D entities and therefore, their F+C effect may not be readily perceivable yet the distortion could be much easier to spot. Furthermore, we need to carefully design our solution in order to maintain key physical properties for streamlines during the deformation process. Finally, to make this deformation framework truly useful, we should incorporate the unique features of flow field and streamlines for both automatic and manual focus identification and F+C visualization.

3 Our Deformation Framework

The basic idea of our F+C flow visualization is to partition the flow field's volume space into blocks and deform the blocks to guide streamline repositioning. Given a vector field, we uniformly partition it into a grid space, $\mathbf{G} = \{\mathbf{V}, \mathbf{E}, \mathbf{B}\}$, where $\mathbf{V}$ is the set of all vertices $\mathbf{V} = \{\mathbf{v}_0^T, \mathbf{v}_1^T, \ldots, \mathbf{v}_n^T\}$, and $\mathbf{E}$ and $\mathbf{B}$ are the sets of all edges and all blocks, respectively. During the deformation, we compute a new set of vertex positions $\mathbf{V}' = \{\mathbf{v}_0'^T, \mathbf{v}_1'^T, \ldots, \mathbf{v}_n'^T\}$, with the intention that the deformed blocks under focus will grow while others blocks will shrink. Clearly, some distortion will be introduced in this deformation process. By minimizing the energy function described in Section 3.3, we aim to spread the unresolved distortion to the blocks according to their importance values, so that interesting blocks of focus can maintain their shapes while less interesting blocks and empty blocks can absorb more distortion and even be squeezed excessively into a plane. Our deformation framework consists of four key steps: *block importance evaluation*, *manual feature specification*, *grid space deformation* and *streamline repositioning*. The user can choose automatic block importance evaluation and/or manual feature specification for F+C visualization. Note that although the outputs of some steps depend on how the streamlines are placed or selected, our deformation framework can work with any streamline placement and selection algorithm. In our experiment, we apply random seed placement and streamline selection algorithm described in Tao et al. [16]. In the following, we describe each step in detail, followed by a description of evaluating the errors introduced in the deformation process.

3.1 Block Importance Evaluation

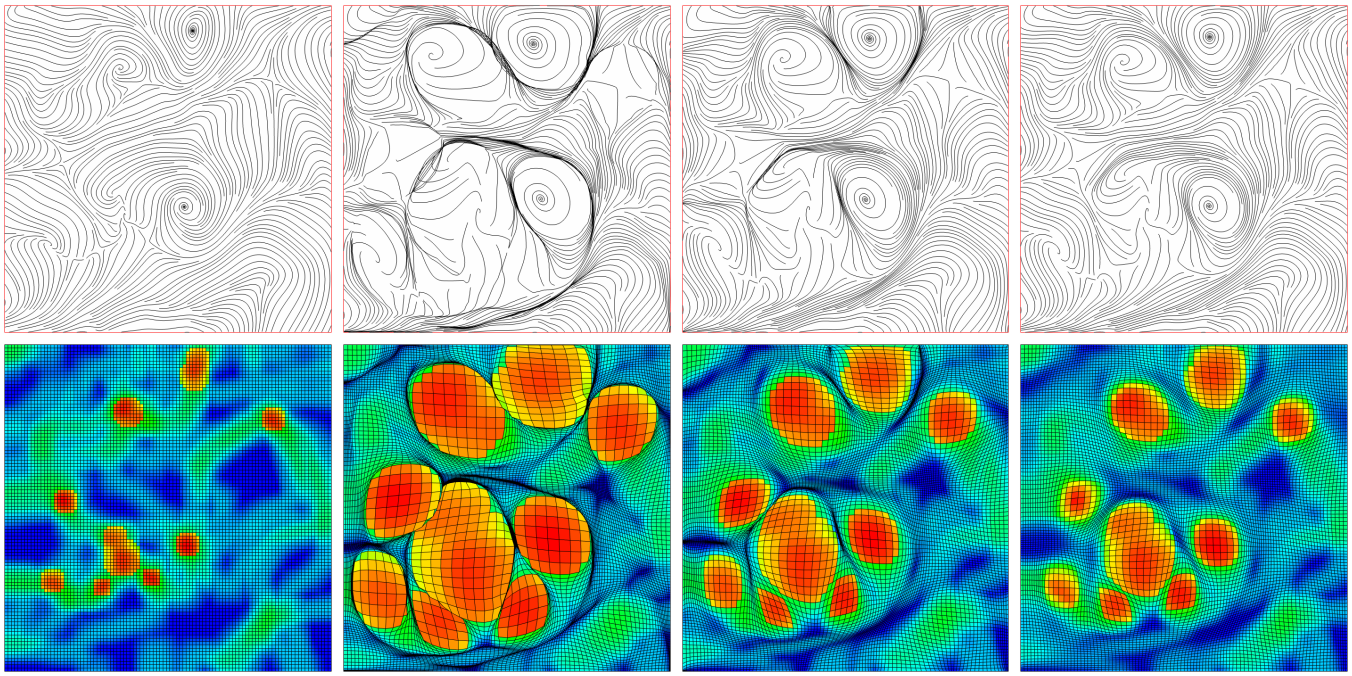
We propose two different ways to define the importance of a block. One is to automatically derive block importance based on flow information, such as the flow entropy [2], [21]. The other way is to manually decide block importance by incorporating user input (refer to Section 3.2). Once we derive the importance values of all blocks, we normalize them to $[0,1]$ and use them as the weighting factors for individual block expansion (refer to Section 3.3). We note that our deformation framework does not depend on any specific approach for importance evaluation. Thus, other importance evaluation techniques could also be applied.

Automatic Importance Computation. For the automatic importance evaluation, we measure the importance of a block using its entropy. Let X be a discrete random variable with alphabet $\mathscr{X}$ and probability mass function $p(x)$, $x \in \mathscr{X}$. The *entropy* of X is defined as

$$H(X) = -\sum_{x \in \mathscr{X}} p(x) \log p(x), \tag{1}$$

where $p(x) \in [0.0, 1.0]$, $\sum_{x \in \mathscr{X}} p(x) = 1.0$. In our case, the outcome of the random variable X is decided by both the directions and magnitudes of vectors, as described by Tao et al. [16]. Intuitively, by considering these two factors, the blocks that contain simple flow patterns will have small entropy values, since the vectors in those blocks are similar; while the blocks that contain complicated flow patterns will have large entropy values, since the vectors in these blocks might vary in both direction and magnitude.

We give two methods to calculate the entropy of a block. The first method is to compute the entropy of a block over the vectors at all points sampled along the streamlines passing through the block. This method considers only the current streamline pool, and will give a high entropy value for blocks that are intersected by the generated streamlines and zero for blocks that do not contain any streamline. The drawback is that the entropy field will need to be recomputed if the streamline pool changes, e.g., adding more streamlines or removing some existing streamlines. Furthermore, due to the discontinuity



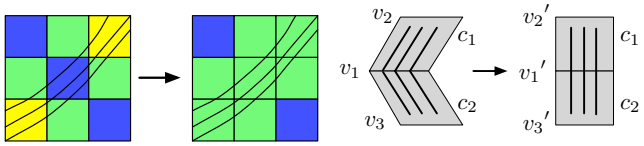
(a) (b) (c) (d)

Fig. 1. Automatic multi-F+C visualization on a 2D flow field and the corresponding block importance grid. (a) is the original streamline visualization. (b) is the naïve deformation that only considers individual block expansion. (c) is the deformation with adding neighboring block smoothing. (d) is the deformation with considering both neighboring block smoothing and flow-aware adjustment.

between the streamlines, the derived entropy field might lose some continuity, e.g., the entropy of neighboring blocks might vary greatly.

The second method is to compute the entropy for each voxel by considering a local window centered at the voxel (we use a $9 \times 9 \times 9$ window). For each window, we compute the entropy based on the vectors' direction and magnitude. We then average all voxels' entropy values in a block to compute the block's importance value. Figure 1 (a) shows an example of the resulting block importance grid for a 2D flow field. This method produces a stable entropy field, which is independent of the generated streamlines. However, with this method, the resulting F+C visualization cannot fully utilize the empty regions, i.e., the regions that contain no streamline segments, which can be deformed into any shape without being noticed. Thus, we add an additional step to calculate the number of streamlines in each block, and set the entropy value of a block to zero if it is empty.

In our experiment with 3D flow fields, we apply the first method, since it is more closely related to the streamlines that are visualized. On the contrary, the entire volume of vector data itself is not directly visible to the user. This implies that the features can only be observed if they are captured by the streamlines. Thus, by applying the first method, we consider those regions that contain interesting patterns and are well captured by the streamlines to be more important. The flow patterns that are more complicated and difficult to predict are considered to be more interesting. Note that the importances of blocks also depend on the tasks of users. In some scenarios, the users might prefer to enlarge a region even if there are only a few streamlines passing through it. We enable this through manual feature specification (refer to Section 3.2).



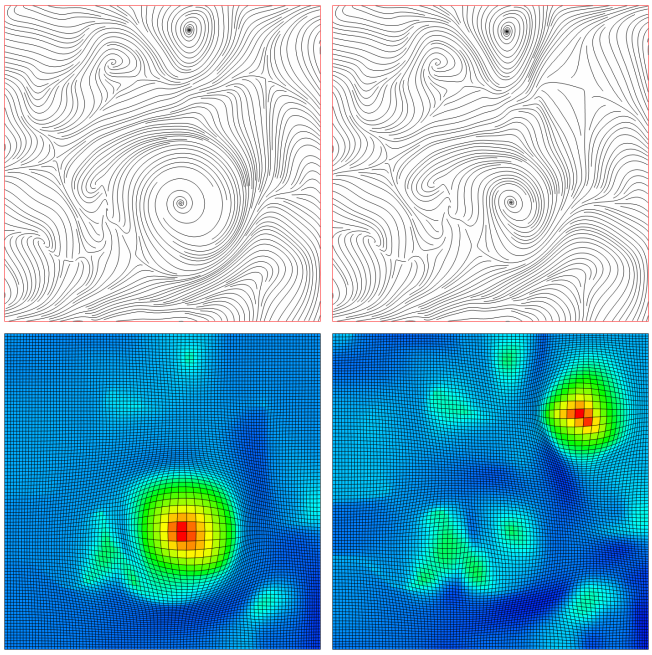
(a) (b)

Fig. 2. (a) flow-aware adjustment. (b) flow-aware smoothing.

Flow-Aware Adjustment. We introduce flow-aware adjustment to smooth the importance values between neighboring blocks that share a large number of streamlines. We notice that preserving the shape of featured blocks does not guarantee the shape of deformed streamlines in the blocks, since each streamline may cross multiple blocks of varying importance values. The distortion will be obvious when a less important block is surrounded by the important ones, as shown in Figure 1 (c). We solve this problem by considering the importance value as some energy term: when the flow moves from an important block to a less important one, it will also carry the energy along with it. In other words, neighboring blocks sharing more streamlines in common should have more similar importance values, as illustrated in Figure 2 (a). This can be formulated as minimizing the following term

$$D_d = \sum_{\mathbf{b}_i} \sum_{\mathbf{b}_j \in B(\mathbf{b}_i)} w_{ij} \left(I_i' - I_j'\right)^2 + \sum_{\mathbf{b}_i} \beta \left(I_i' - I_i\right)^2, \qquad (2)$$

where $B(\mathbf{b}_i)$ is the set of neighboring blocks of block $\mathbf{b}_i$, $w_{ij} = n_{ij}/n_{\max}$ is a normalized weight, n_{ij} is the number of



(a) (b)

Fig. 3. Manual focus selection and F+C visualization. (a) and (b) are the results with two different user-specified centers of focus, respectively.

streamlines shared by blocks $\mathbf{b}_i$ and $\mathbf{b}_j$, $n_{\max}$ is the maximum number of streamlines shared by two neighboring blocks, β is a user-specified weight, I_i is the importance value of $\mathbf{b}_i$ before the adjustment, and I'_i and I'_j are the importance values of $\mathbf{b}_i$ and $\mathbf{b}_j$ after the adjustment, respectively. This is similar to filtering the importance values along the streamlines. Note that the adjusted field will be closer to the original one if β is larger, and the effect of this flow-aware adjustment will be weaker. The adjusted field will be almost unchanged when $\beta = 1.0$. Since Equation 2 is a quadratic function, the method of least squares can be used to solve for the importance values.

3.2 Manual Feature Specification

Besides computing block importance automatically, we can also evaluate it based on the focal region specified by the user. In this paper, we present two different options: *block focus* and *streamline focus*.

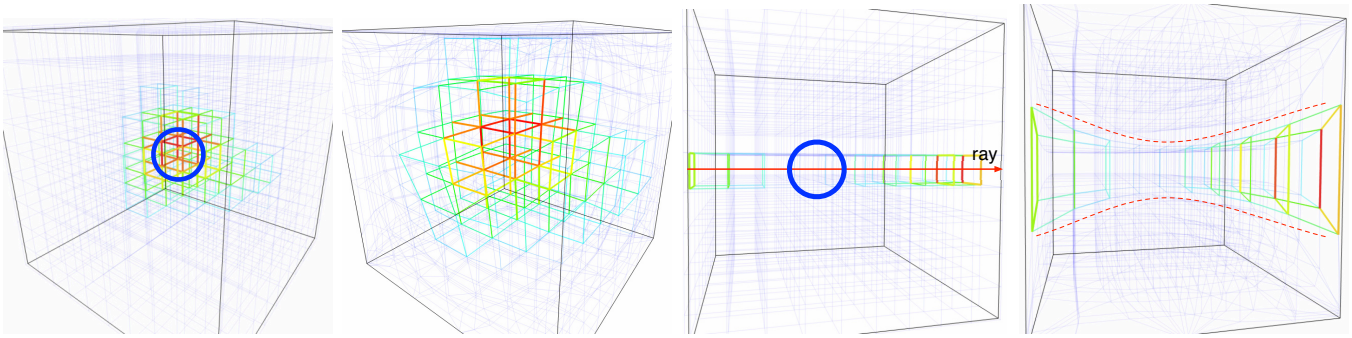
Block Focus. For this option, the user simply clicks on the visualization result to specify a block as the center of focus. Figure 3 shows such an example with the same 2D flow field shown in Figure 1 (a) as the input. For a 3D flow field, the visualization result is a 2D projection, so we still need to estimate the depth value to pinpoint the focal point. A straightforward solution is to follow the first hit on the streamlines displayed in the projection, but the focal region selected in this way may miss internal flow features. A better solution which we will use is to identify the most prominent feature along the direction of projection. For instance, Lee et al. [12] presented the concept of maximal entropy projection (MEP). For each pixel on the screen, a ray is cast into the entropy volume and the z-value is given by that of the voxel with the maximal entropy. In this way, we will select the most important block along the ray specified by the user. We can use this manual feature selection to modulate the automatic importance evaluation and modify the automatic focus result.

We can also use the manual focus independently, starting from uniform importance values for all blocks and modulating those values according to some predefined templates. We design two templates for exploring small or occluded flow features: *spherical block focus* and *hourglass block focus*. Examples are shown in Figure 4. The spherical block focus assigns the largest importance value to the specified block, and the importance values gradually decrease for blocks further away from the focus. This template is suitable for magnifying small features, since the center region, which contains the features, will grow as other regions shrink. This template, however, could be ineffective when the features are hidden by other streamlines due to occlusion, since the blocks located at the outer rings or layers could be denser than those at the center. The hourglass block focus is designed to solve this problem. Instead of magnifying the feature region, we enlarge the blocks along the user-specified ray except the feature region. It assigns larger importance values for the blocks that are closer to the ray and gradually decreases the values for those further away. Furthermore, it assigns smaller importance values for the blocks whose depth are close to the depth of the specified block. In this way, the blocks with large importance values form the shape of an hourglass, whose axis is along the ray and whose center is the block of focus. By magnifying the blocks that occlude the block of focus, we are now able to see through and observe the flow features that are previously occluded.

Streamline Focus and Animation. Another useful way of manual focus specification is to allow the user to select a streamline of interest through first hit. We then perform F+C visualization on the entire streamline by assigning larger importance values for the blocks that the streamline goes through and smaller values for other blocks. An animation of F+C visualization can also be generated by moving the focal point along a streamline from end to end. To produce smoother animation, we insert additional frames, in which the grid vertices are linearly interpolated from the grid vertices of the two neighboring frames and the streamlines are repositioned according to the intermediate grids. In cases where there are no desired streamlines going through the regions to be explored, a user-drawn path could be used instead for an effective exploration of interesting regions.

3.3 Grid Space Deformation

The grid space deformation should serve the following purposes: the blocks in focus regions should be magnified while pushing the context regions to be shrunk; the streamlines in focus regions should keep their shape unchanged while the distortion for other streamlines should be minimized; and the relative position among streamlines should not change dramatically. To perform grid space deformation, we consider *individual block expansion*, *neighboring block smoothing*, and *flow-aware smoothing*, inspired by Wang et al. [20]. Individual block expansion allows each block to resize independently based on a global scaling factor and its weighting factor.



(a) (b) (c) (d)

Fig. 4. The grids before and after the deformation using block focus. (a) and (b) show original and deformed grids for the spherical block focus, where the focus is at the center of the volume with the largest importance value. (c) and (d) show original and deformed grids for the hourglass block focus, where the focus is highlighted with the blue circle in the first row and the deformed shape is enhanced by the red dashed lines for clearer exposition.

Neighboring block smoothing preserves the continuity of neighboring blocks while flow-aware smoothing preserves the shape of the streamlines. We add *edge flipping constraints* to avoid neighboring block intersection and *volume boundary constraints* to retain the size and shape of the bounding space. We formulate these considerations into energy terms and search for a deformed grid that minimizes the objective function under the edge flipping and volume boundary constraints. To achieve this, we transform the objective function into a linear system and solve for the unknown vertex positions in a least-squares sense.

Individual Block Expansion. We introduce this energy term to preserve the cube shape of the blocks. Given a block $\mathbf{b}_i$, ideally, its deformed version $\mathbf{b}'_i$ can be obtained by applying scaling, rotation and translation to the original block, i.e., $\mathbf{b}'_k = s_{\mathbf{b}_k}\mathbf{R}_{\mathbf{b}_k}\mathbf{b}_k + \mathbf{t}_{\mathbf{b}_k}$. Note that by multiplying the rotation term $\mathbf{R}_{\mathbf{b}_k}$, we allow the block to rotate in order to better utilize the space and incur less distortion. Denoting the set of edges of block $\mathbf{b}_k$ as $E(\mathbf{b}_k)$, we express the energy term of block deformation as

$$D_f(\mathbf{b}_k) = \sum_{\mathbf{e}_{ij}\in E(\mathbf{b}_k)} w_{\mathbf{b}_k} \|\mathbf{e}'_{ij} - s_{\mathbf{b}_k}\mathbf{R}_{\mathbf{b}_k}\mathbf{e}_{ij}\|^2, \tag{3}$$

where $w_{\mathbf{b}_k}$ is the normalized importance value of block $\mathbf{b}_k$, $\mathbf{e}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ and $\mathbf{e}'_{ij} = \mathbf{v}'_i - \mathbf{v}'_j$ are the edges before and after the deformation, respectively. The translation $\mathbf{t}_{\mathbf{b}_k}$ is canceled out due to the simple fact

$$\mathbf{e}'_{ij} = \mathbf{v}'_i - \mathbf{v}'_j = s_{\mathbf{b}_k}\mathbf{R}_{\mathbf{b}_k}\mathbf{v}_i + \mathbf{t}_{\mathbf{b}_k} - (s_{\mathbf{b}_k}\mathbf{R}_{\mathbf{b}_k}\mathbf{v}_j + \mathbf{t}_{\mathbf{b}_k}) = s_{\mathbf{b}_k}\mathbf{R}_{\mathbf{b}_k}(\mathbf{v}_i - \mathbf{v}_j). \tag{4}$$

Initially, we set $s_{\mathbf{b}_k}$ to a user-defined scaling factor s_f and $\mathbf{R}_{\mathbf{b}_k}$ to an identity matrix for all blocks to solve for a new set of vertex positions $\mathbf{V}'$. Then, for each block, we compute $s_{\mathbf{b}_k}$ and $\mathbf{R}_{\mathbf{b}_k}$ from the deformed vertices, and apply the updated $s_{\mathbf{b}_k}$ and $\mathbf{R}_{\mathbf{b}_k}$ to solve for another set of vertex positions. This procedure is repeated for several iterations, until the system converges or a certain number of iterations is reached. Although we assign the same scaling factor s_f for all the blocks, only the blocks with larger importance values can get the chance to be enlarged, since they will receive larger penalty (i.e., increase $D_f(\mathbf{b}_k)$) for not approaching the target scaling factor. In contrast, those less important or trivial blocks, with very small or even zero importance values, can be squeezed substantially without receiving much penalty.

Neighboring Block Smoothing. We introduce this smoothing term in order to reduce the size difference between neighboring blocks. Since the importance values of neighboring blocks may vary dramatically, only considering individual block expansion deforms neighboring blocks to have very different sizes, which easily makes neighboring blocks intersect each other. As shown in Figure 1 (b), such a naïve deformation distorts streamlines that span across multiple blocks and lead to pronounced artifacts along block boundaries. To avoid this, we preserve the Laplacian coordinates [15] of the deformed vertices $\mathbf{v}'_i$ by minimize the following energy term

$$D_\ell = \sum_{\mathbf{v}_i\in\mathbf{V}} \|\mathbf{L}(\mathbf{v}'_i) - s_{\mathbf{v}_i}\mathbf{R}_{\mathbf{v}_i}\mathbf{L}(\mathbf{v}_i)\|^2, \text{ where}$$

$$L(\mathbf{v}_i) = \frac{1}{|V(\mathbf{v}_i)|}\sum_{\mathbf{v}_j\in V(\mathbf{v}_i)} (\mathbf{v}_i - \mathbf{v}_j),$$

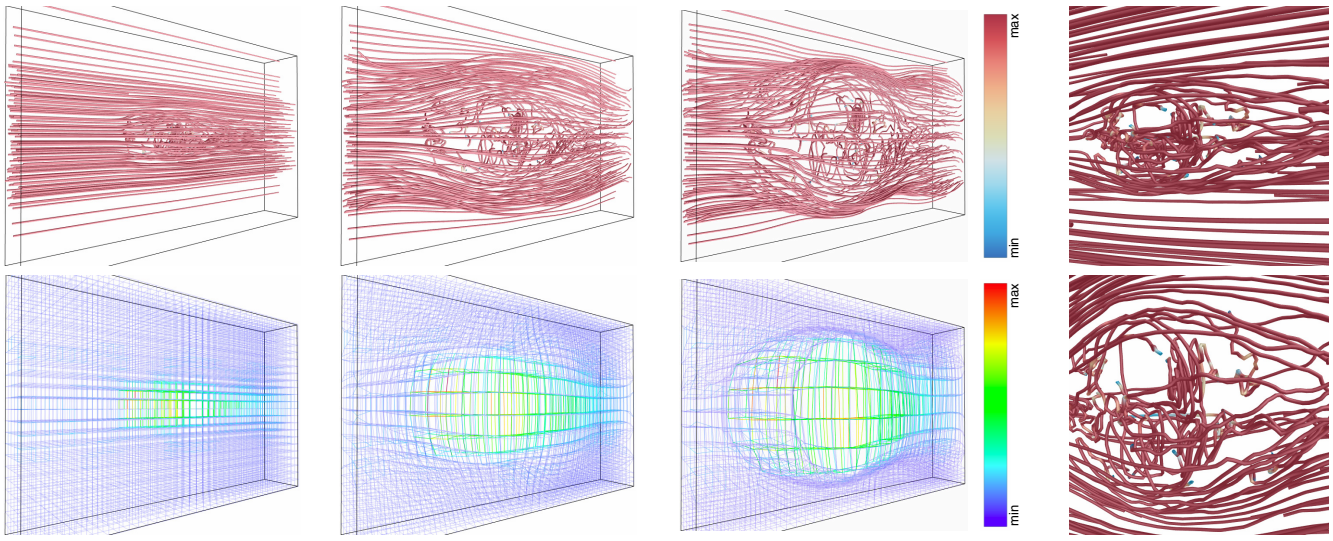
$$s_{\mathbf{v}_i} = \frac{1}{|B(\mathbf{v}_i)|}\sum_{\mathbf{b}_k\in B(\mathbf{v}_i)} s_{\mathbf{b}_k}, \text{ and } \mathbf{R}_{\mathbf{v}_i} = \frac{1}{|B(\mathbf{v}_i)|}\sum_{\mathbf{b}_k\in B(\mathbf{v}_i)} \mathbf{R}_{\mathbf{b}_k}. \tag{5}$$

In Equation 5, $s_{\mathbf{v}_i}$ and $\mathbf{R}_{\mathbf{v}_i}$ are the scaling factor and rotation matrix for the Laplacian coordinates of vertex $\mathbf{v}_i$, respectively, $V(\mathbf{v}_i)$ and $B(\mathbf{v}_i)$ are the sets of neighboring vertices and blocks of $\mathbf{v}_i$ respectively, and $|V(\mathbf{v}_i)|$ and $|B(\mathbf{v}_i)|$ are the numbers of neighboring vertices and blocks of $\mathbf{v}_i$ respectively. Since the Laplacian coordinates are zero vectors for all the inner vertices, we can actually simplify Equation 5 to

$$D_\ell = \sum_{\mathbf{v}_i\in\mathbf{V}} \|\mathbf{L}(\mathbf{v}'_i)\|^2, \tag{6}$$

and only add these constraints to the inner vertices. To simplify the calculation for the boundary vertices, we apply a similar approach that only considers the inner vertices on each boundary face. For each vertex, the Laplacian coordinates are computed from the four neighboring vertices that are also located on the boundary face. In this way, the Laplacian coordinates are still zero vectors and therefore we do not need to apply the scaling and rotation.

Flow-Aware Smoothing. Flow-aware smoothing serves a similar purpose as flow-aware adjustment in the block im-



(a) (b) (c) (d)

Fig. 5. F+C visualization of the car flow data set with automatic importance evaluation. (a) shows the original streamlines and the grid. (b) and (c) show the deformed streamlines and grids with two different sets of parameter setting. (d) shows a zoom-in result around the features with the original streamlines in the first row and the deformed ones in the second row.

portance evaluation step. By introducing this term into the deformation process, we are able to reduce the difference between the transformations of two neighboring blocks that share a large number of streamlines. In the left side of Figure 2 (b), we illustrate an example of severe streamline distortion, where straight streamlines are deformed into polylines due to the different orientations of the two neighboring blocks $\mathbf{b}_1$ and $\mathbf{b}_2$. To reduce this kind of distortion, we drag $\mathbf{v}_1$ shared by $\mathbf{b}_1$ and $\mathbf{b}_2$ back to the center of the two adjacent vertices $\mathbf{v}'_2$ and $\mathbf{v}'_3$, as shown in the right side of Figure 2 (b). The flow-aware smoothing can be achieved by minimizing the energy term

$$D_s(\mathbf{b}_i,\mathbf{b}_j) = \sum_{\mathbf{v}_i \in V(\mathbf{b}_i)\cap V(\mathbf{b}_j)} w_{ij}\|\mathbf{v}'_j + \mathbf{v}'_k - 2\mathbf{v}'_i\|^2, \qquad (7)$$

where $V(\mathbf{b}_i)\cap V(\mathbf{b}_j)$ is the set of vertices shared by blocks $\mathbf{b}_i$ and $\mathbf{b}_j$, w_{ij} is defined in the same way as in Equation 2, and $\mathbf{v}'_j$ and $\mathbf{v}'_k$ are the two neighboring vertices adjacent to $\mathbf{v}'_i$.

Edge Flipping Constraints. Edge flipping can only be found for edges that belong to less important blocks. However, such a happening is still not desirable, since it might lead to the change of relative positions of blocks that are incident to the flipped edge. We detect edge flipping by computing the angle formed by the deformed edge and the original one. If the angle is larger than 90°, we consider the edge as flipped. Note that a flipped edge indicates that it has a negative scaling factor. Therefore, we enforce the flipped edge to be aligned with its original direction, but with a very small scaling factor. This can be achieved by adding the following energy term

$$D_{\mathbf{e}_{ij}} = \alpha\|\mathbf{e}'_{ij} - \delta\mathbf{e}_{ij}\|^2, \qquad (8)$$

where $\mathbf{e}_{ij}$ is a flipped edge, α is a large constant to enforce the constraints, and δ is a small constant to preserve the block from being shrunk to zero size or even being negatively scaled. In this paper, we set $\alpha = 50.0$ and $\delta = 0.1$ for all data sets we experimented with. Since the flipped edge will be short after being flipped back, we ignore the rotation matrix for efficiency.

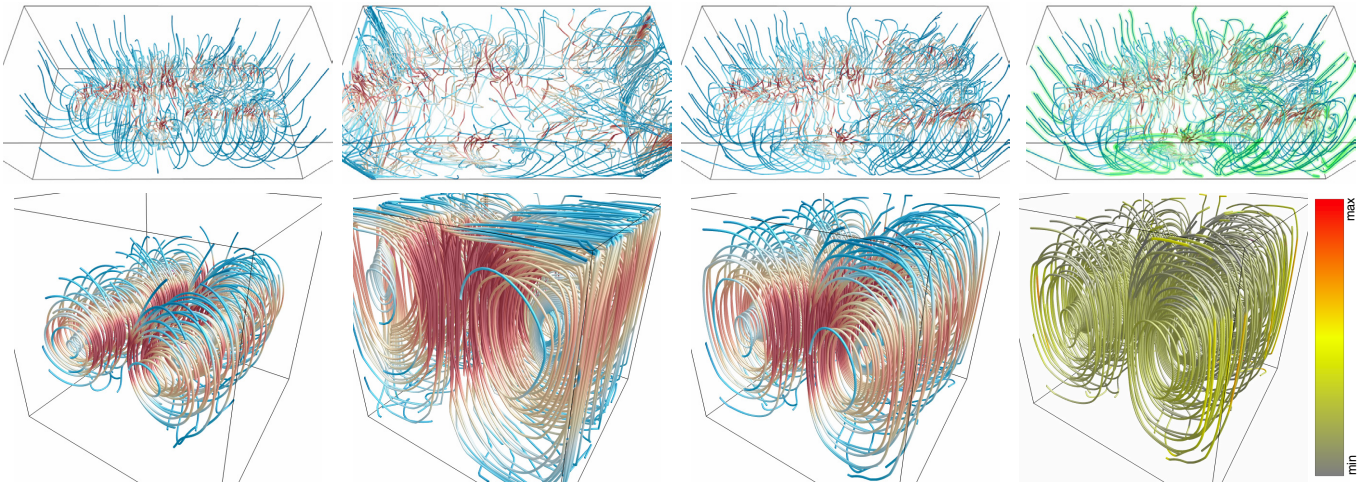
Volume Boundary Constraints. In order to retain the size and shape of the volume bounding space, we add the boundary constraints to ensure that the vertices in the grid are always placed within the boundary throughout the deformation process. Written in equation,

$$\begin{cases} \mathbf{v}'_{i,x} = \mathbf{v}_{i,x} & \text{if } \mathbf{v}_{i,x} \text{ is on the } yz \text{ boundary plane,} \\ \mathbf{v}'_{i,y} = \mathbf{v}_{i,y} & \text{if } \mathbf{v}_{i,y} \text{ is on the } xz \text{ boundary plane,} \\ \mathbf{v}'_{i,z} = \mathbf{v}_{i,z} & \text{if } \mathbf{v}_{i,z} \text{ is on the } xy \text{ boundary plane.} \end{cases} \qquad (9)$$

Solving Linear System. The energy function that we would like to minimize is

$$D = \sum_{\mathbf{b}_k \in \mathbf{B}} D_f(\mathbf{b}_k) + w_\ell D_\ell + w_s \sum_{\mathbf{b}_i,\mathbf{b}_j \in \mathbf{B}} D_s(\mathbf{b}_i,\mathbf{b}_j) + \sum_{\mathbf{e}_{ij} \in \mathbf{E}^*} D_{\mathbf{e}_{ij}}, \qquad (10)$$

where $\mathbf{b}_i$ and $\mathbf{b}_j$ are neighboring blocks, $\mathbf{E}^*$ is the set of flipped edges, and w_ℓ and w_s are parameters to adjust the weights of the two smoothing terms. Each term in this energy function is converted to a row in the linear system $\mathbf{AV}' = b(\mathbf{V})$, where $\mathbf{A}$ represents the coefficients of unknown vertex positions $\mathbf{V}'$, and $b(\mathbf{V})$ are the vectors in the right-hand side of simultaneous equations. Note that the boundary constraints are set when configuring the solver, which do not affect the construction of coefficient matrix $\mathbf{A}$. Each dimension of the vertex coordinates can be solved independently in multiple passes. In each pass, we solve the linear system to obtain a new set of vertex positions $\mathbf{V}'$, and update the scaling factor $s_{\mathbf{b}_k}$ and rotation matrix $\mathbf{R}_{\mathbf{b}_k}$ for each block to better estimate the desired transformation. We use the method described by Horn [9] to calculate the scaling factors and rotation matrices. From the corresponding coordinates of the vertices of one block before and after the deformation, a 4×4 matrix for that block can be constructed. Then, the rotation matrix is represented by a unit quaternion, which is the eigenvector associated with the most positive eigenvalue of this matrix. To achieve interactive deformation, we leverage a GPU implementation



(a) (b) (c) (d)

Fig. 6. F+C visualization results with automatic importance evaluation. First row: the crayfish data set. Second row: the two swirls data set. (a) is the original visualization result. (b) is the naïve deformation that only considers individual block expansion. (c) is the deformation with adding neighboring block smoothing, flow-aware adjustment and flow-aware smoothing. (d) shows deformed streamlines with block errors mapped to colors along the points on each streamline.

of the concurrent number cruncher (CNC) sparse solver [1] to solve the linear system. By increasing or decreasing the global scaling factor s_f, the user is able to observe the deformation or recovery process in a progressive manner for intuitive understanding.

3.4 Streamline Repositioning and Error Evaluation

After grid deformation, we reposition the streamlines by computing each point along the line as a linear combination of its corresponding eight block vertices in the deformed grid. To measure block distortion, we transform each original block to have the same size and orientation as the deformed one and compare their difference

$$D_{err}(\mathbf{b}_k) = \sum_{\mathbf{e}_{ij} \in E(\mathbf{b}_k)} \frac{w_{\mathbf{b}_k} \|\mathbf{e}'_{ij} - s_{\mathbf{b}_k} \mathbf{R}_{\mathbf{b}_k} \mathbf{e}_{ij}\|^2}{s^2_{\mathbf{b}_k}}. \qquad (11)$$

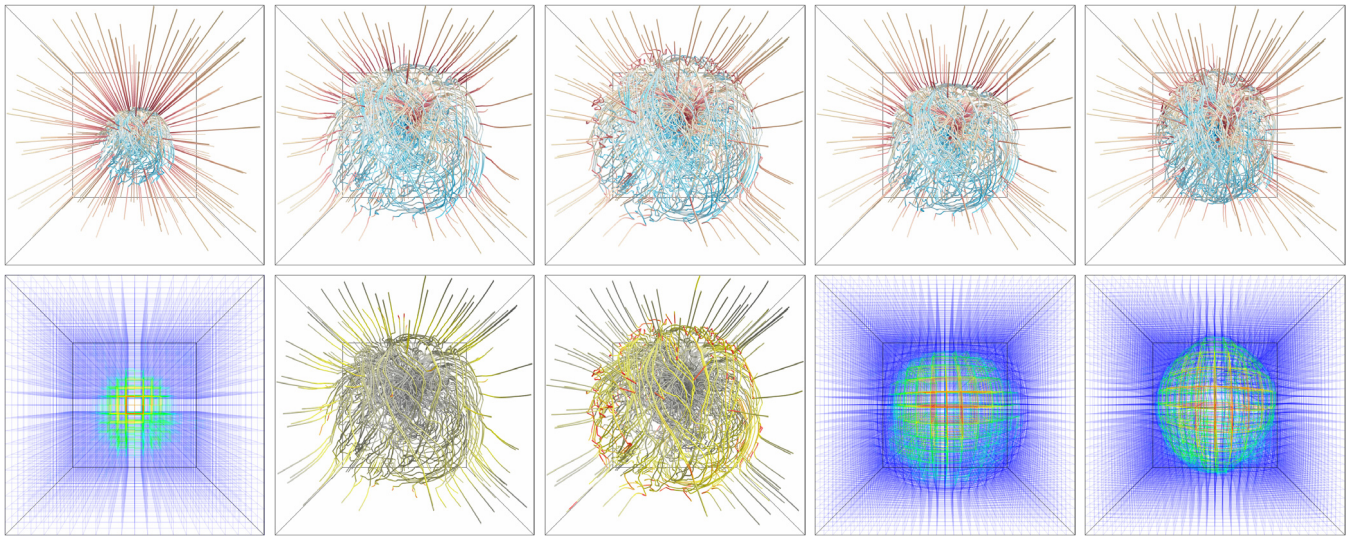
Note that Equation 11 is the block deformation term (Equation 3) divided by the square of the scale, which normalizes the error for blocks with different sizes. We measure the distortion whenever we update the scaling factor and stop the magnification when increasing the scaling factor does not further magnify the flow features of interest.

Evaluating the distortion also serves two other important purposes. First, it provides a quantitative tool for us to compare our approach to others. Second, it informs the users how well the original flow patterns are preserved. Since the distortion are inevitable in the deformation process, it will be helpful to inform the users *where* the distortion exists and *how* severe the distortion each block suffers, so that they will not be misled by some abnormal patterns created. We visualize the distortion with two methods. One method is to map error values to streamline colors, as shown in the second row of Figure 6 (d). With this method, error values can be better revealed while the information (e.g., velocity magnitude) about the flow field shown by the original color map is lost. The other method is to draw green semitransparent tubes over the streamlines with higher opacity values indicating larger distortion. With this method, the error values only provide a rough idea about which regions suffer larger distortion. In the first row of Figure 6 (d), we can observe that the grids with larger distortion are mostly located around the volume boundary. We suggest to use the first method when analyzing the distortion from deformation, and to use the second method when investigating the flow field itself with the distortion as an additional visual hint.

4 Results and Discussion

4.1 Performance and Parameter Settings

Table 1 shows the data sets we experimented with and the timing results for the block importance evaluation, flow-aware adjustment, block deformation and streamline repositioning steps. The timing was collected on a PC with an Intel Core i7-960 CPU running at 3.2GHz, 24GB main memory, and an nVidia GeForce GTX 580 graphics card with 1.5 GB graphics memory. As expected, the deformation time was proportional to the grid dimension, since a grid with larger resolution would have more vertex positions to solve which means the linear system has more variables and equations. Since our framework is designed to visualize the entire flow field in a F+C manner, the grid resolution is not supposed to grow significantly even with larger data sets. Nevertheless, the time cost for streamline repositioning appeared to be similar. Although the total number of streamline points for the supernova data set is much larger than the others, the repositioning time was only slightly longer. Our implementation utilized the CUDA OpenGL interoperability so that the repositioning was performed in parallel on the GPU and there was no need to transfer the rendering data between main memory and graphics memory. For block importance evaluation and flow-aware adjustment, although they took a longer time for the supernova data set, there was no clear pattern in timing among the grid resolution, number of streamlines, and total number of points on streamlines at the scales of other data sets we explored. The timing was dominated by the deformation time.



(a) (b) (c) (d) (e)

Fig. 7. F+C visualization results with automatic importance evaluation for different scaling factors s_f and grid resolutions. (a) is the original visualization result of the supernova data set. (b), (c), (d) and (e) are deformation results produced with s_f and grid resolution as 5.0 and $20 \times 20 \times 20$, 10.0 and $20 \times 20 \times 20$, 5.0 and $30 \times 30 \times 30$, and 10.0 and $30 \times 30 \times 30$, respectively. The second row shows the corresponding grid, block distortions mapped to streamline colors, and deformed grids.

TABLE 1

The flow data sets and their timing results. We run ten iterations for the deformation step. All the timing results are calculated by averaging the results gathered from 100 runs. The evaluation time is for block importance evaluation and the adjustment time is for flow-aware adjustment.

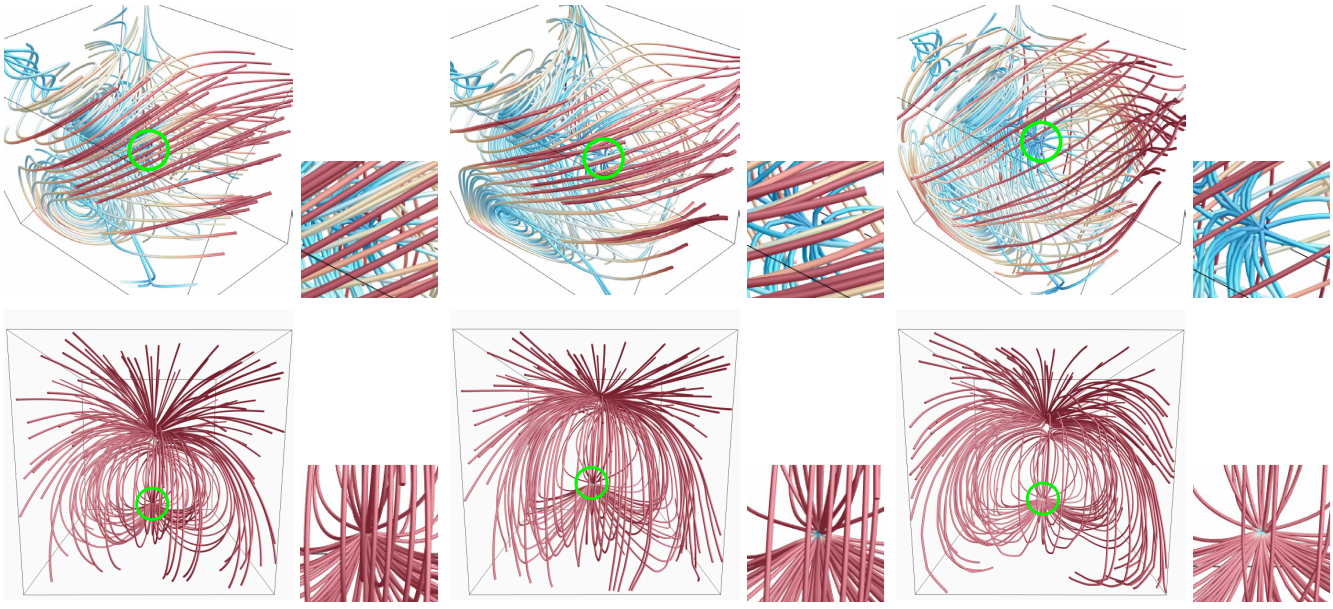
data set	dimension	grid dimension	evaluation time	adjustment time	deformation time	# lines	avg. # pts. per line	repositioning time
five critical points	$51 \times 51 \times 51$	$10 \times 10 \times 10$	0.772 ms	8.996 ms	0.238 sec	140	58.8	0.165 ms
tornado	$64 \times 64 \times 64$	$12 \times 12 \times 12$	1.267 ms	12.089 ms	0.317 sec	60	365.4	0.179 ms
two swirls	$64 \times 64 \times 64$	$12 \times 12 \times 12$	1.465 ms	11.568 ms	0.303 sec	100	236.7	0.185 ms
electro	$64 \times 64 \times 64$	$12 \times 12 \times 12$	1.077 ms	8.351 ms	0.290 sec	200	52.6	0.162 ms
car flow	$368 \times 234 \times 60$	$36 \times 23 \times 5$	2.107 ms	11.845 ms	0.493 sec	140	198.7	0.180 ms
crayfish	$322 \times 162 \times 119$	$21 \times 10 \times 7$	1.736 ms	8.826 ms	0.284 sec	100	248.8	0.167 ms
computer room	$417 \times 345 \times 60$	$27 \times 22 \times 3$	2.578 ms	9.645 ms	0.288 sec	200	182.7	0.174 ms
hurricane	$500 \times 500 \times 100$	$24 \times 24 \times 4$	2.712 ms	11.820 ms	0.362 sec	140	346.7	0.177 ms
supernova	$864 \times 864 \times 864$	$20 \times 20 \times 20$	9.521 ms	23.763 ms	0.908 sec	200	692.4	0.243 ms

Except for the large supernova data set, the overall time to update the F+C visualization results was less than 0.5 second, which makes our deformation approach interactive.

The parameters used include a user-defined scaling factor s_f, a user-specified weight β in the flow-aware adjustment (Equation 2), and the two weighting factors w_ℓ and w_s for smoothing the energy terms (Equation 10). In our experiments, we usually used a fixed scaling factor $s_f = 5.0$ and set $\beta = 0.1$, $w_\ell = 2.5$, $w_s = 3.0$. The scaling factor could be a bit larger than the actual scaling that can be achieved, since the blocks would stop growing up to a certain degree. Normally, we found that $\beta = 0.1$ was appropriate for most cases, and $\beta = 0.04$ for a few cases where neighboring blocks sharing many streamlines are of different sizes. For the Laplacian smoothing, we might increase w_ℓ to 3.0, if obvious size change between neighboring blocks can be found. We might decrease w_ℓ to 2.0 to obtain a larger scaling when the change in size was already smooth. For the flow-aware smoothing, $w_s = 3.0$ was good for most cases. However, if many streamlines were distorted to polylines, we would increase w_s to 6.0. All visualization results we present in this section were generated with these parameter settings.

4.2 Focus+Context Visualization Results

Automatic Importance Evaluation. Figure 5 and Figure 6 show the F+C visualization results with automatic importance evaluation. Figure 5 clearly shows the effect of F+C visualization which allows us to observe internal streamline features along the car which were previously occluded by the surrounding straight streamlines. In Figure 5 (d), we remove some context streamlines that occlude the features from the original streamline display, and compare the resulting streamlines with the deformed ones in the two zoom-in images. Although the deformed streamlines are stretched in the vertical direction for a clearer view, the spatial relationships among streamlines of focus and the shapes of streamlines are well preserved in the F+C deformation. As shown in Figure 6 (b), without adding any smoothing term, the regions that are evaluated as the more important ones occupy most of the space, while less important ones are squeezed into thin layers which creates serious distortion. By adding neighboring block smoothing, flow-aware adjustment and flow-aware smoothing



(a) (b) (c)

Fig. 8. F+C visualization results that reveal hidden features for the five critical points (top) and electro (bottom) data sets. The user-specified regions are highlighted in the green circle, together with the enlarged images to the right side of each visualization result. (a) shows the original streamlines. (b) shows the deformed streamlines with the spherical block focus. (c) shows the deformed streamlines with the hourglass block focus.

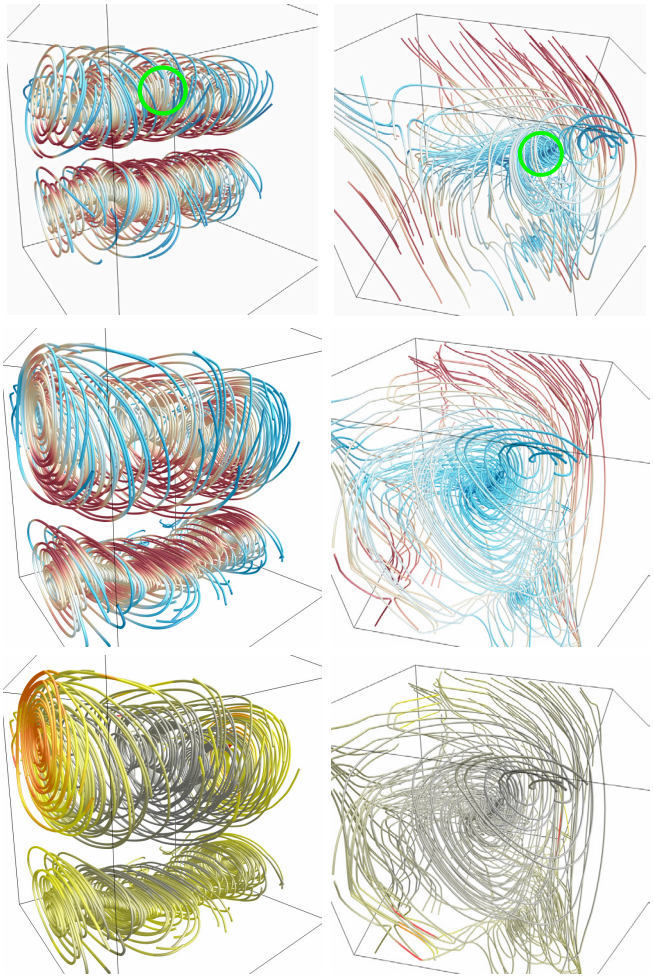
terms, we can observe from Figure 6 (c), that important regions are still well magnified and the volumes are almost filled with streamlines everywhere. Meanwhile, perceptually, those less important streamlines suffer from less distortion. Error results shown in Figure 6 (d) also demonstrate that important regions almost keep their original shapes and less important ones are not seriously distorted either. Although the measured block distortion seems to be large for the two swirls data set, the swirl patterns are still clear in the deformation result. This is due to the fact that the distortion mainly comes from block stretching, which only slightly changes the perceived shape of the streamlines.

In Figure 7, we compare the F+C results with different grid resolutions and scaling factors using the supernova data set. Compared with using $30 \times 30 \times 30$ grid resolution, the important region at the center is expanded to a larger degree using $20 \times 20 \times 20$ grid resolution. This is true for both cases under different scaling factors: 5.0 and 10.0. Even though less important blocks would shrink by only applying the individual block expansion term, the other smoothing terms still maintain the shape of those blocks to some degree. With a higher grid resolution, the number of less important blocks surrounding the important region becomes larger, which increases the resistance and prevents important blocks from growing further. Meanwhile, using a scaling factor of 10.0 does not further magnify the important region, which is also due to the fact that the smoothing terms stop less important blocks from being further squeezed. From deformed grids and evaluated error results shown in the figure, we can observe that using a scaling factor of 10.0 shrink the blocks around the boundary of the important regions, leading to larger distortion. Therefore, in practice, we need to carefully select the appropriate grid resolution and scaling factor instead of simply aiming for larger grid resolutions and higher scaling factors.

Spherical Block Focus. In Figure 9, we show the F+C visualization results with the user-specified spherical block focus. For the two swirls data set, the focus is at the center of the upper swirl. Before the deformation, the two swirls occupy similar space in the volume. By applying the spherical block focus on the upper swirl, the focal region grows, pushing the lower swirl and the two ends of the upper one to the boundary. From the error image, we can observe that the focal region does not suffer much distortion, as the distortion is mainly distributed to the squeezed regions. The five critical points data set also shows a similar result, where the focus is the spiral located at the upper right conner. After the deformation, that spiral is magnified and shifted closer to the center, while the other regions shrink and absorb most of the distortion.

Hourglass Block Focus. Figure 8 demonstrates the effectiveness of our hourglass block focus. For the five critical points data set, the source located at the center of the volume is occluded by some less interesting streamlines with a similar pattern, as shown in Figure 8 (a). After the spherical block focus is applied, the source still cannot be observed clearly as shown in Figure 8 (b), although the source itself has been magnified. This is because the density of streamlines in the front does not change significantly. The hourglass block focus is used to specify the same source as the focus. In Figure 8 (c), the source becomes clearly visible since the streamlines at the outer ring are much sparser. A similar result can be found for the electro data set as the streamlines occluding the source become sparser with the hourglass block focus.

Streamline Focus. In Figure 10, we show the streamline focus results. Unlike the block focus, the streamline focus



(a) (b)

Fig. 9. F+C visualization results with the user-specified spherical block focus for (a) two swirls and (b) five critical points data sets, respectively. The focus is highlighted in the green circle. First row: the original streamlines. Second row: the deformed streamlines. Third row: the deformed streamlines with block errors mapped to colors along the points on each streamline.

treats all the blocks into two different categories: the blocks that the focal streamline does and does not pass through, respectively. Blocks in the same category are assigned the same importance value, so that the transformation for the blocks that contain the focal streamline will be similar and the distortion will be distributed to the rest of blocks more evenly. From the visualization results for both data sets, we can see that the shape of the focal streamline is almost the same as the original one after the deformation, while no obvious change can be observed for other surrounding areas. This implies that the streamlines in the context are more stable and the relationships between the focal streamline and the rest of streamlines are easier to interpret.

Streamline Animation. In Figure 11, we show selected snapshots of streamline animation results where the focal point moves along a user-specified streamline or a user-drawn path. As shown in Figure 11 (a), the animation helps the user explore the regions that a streamline passes through. This is different from our previous approach that magnifies the entire

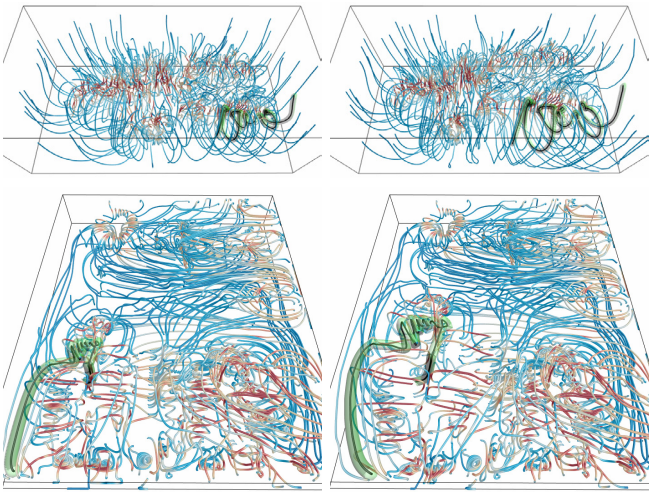
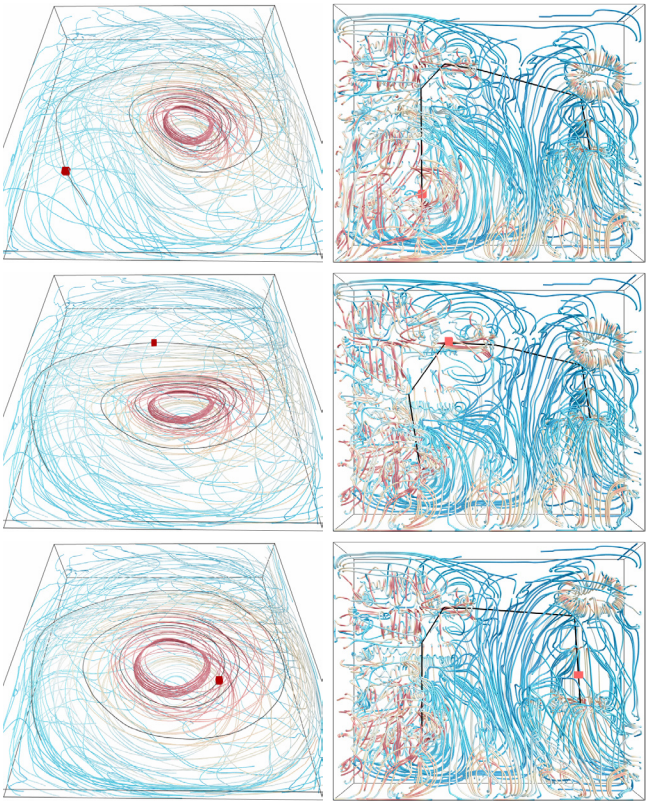


Fig. 10. F+C visualization results with the user-specified streamline focus for the crayfish (top) and computer room (bottom) data sets, where the focal streamline is highlighted in black and surrounded by green semitransparent tubes.



(a) (b)

Fig. 11. Snapshots of F+C animation results. (a) shows a user-specified streamline for the hurricane data set and (b) shows a user-drawn path for the computer room data set. The focal streamline/path is highlighted in black and the current focal point is marked with a red cube.

streamline simultaneously. Since the space in the volume is limited, enlarging multiple regions could either decrease the scaling factor that can be achieved or result in more serious distortion. Using the animation to move the focal point will be more efficient to magnify the regions consecutively. As

TABLE 2
Block distortion evaluated using Equation 11. The blocks are grouped by the distances (in voxels) from their center to the focus.

dist. to focus	# blocks	spherical focus		fisheye focus	
		avg.	max	avg.	max
tornado					
< 10	27	1.641	2.665	13.572	15.987
< 20	251	17.145	37.969	88.950	843.750
< 40	1283	49.445	505.412	66.443	843.750
all	1728	49.434	505.412	49.333	843.750
electro					
< 10	27	2.332	4.219	18.382	43.927
< 20	230	19.749	55.511	95.026	844.575
< 40	1283	42.603	191.895	64.261	844.575
all	1728	40.833	434.848	42.841	844.575

such, we should select those long streamlines that pass through different regions over short ones. Figure 11 (b) shows another example where we explore F+C animation using the user-drawn path.

4.3 Comparison with Fisheye View

For comparison, we implemented the fisheye view F+C technique presented by Sarkar and Brown [14]. For each vertex within the user-specified focal region with radius r_{focus}, we transform the vertex based on the polar coordinate system originated at the center of focus. This maps a vertex with the original coordinates $(r_{ori}, \theta, \gamma)$ to the fisheye coordinates $(r_{feye}, \theta, \gamma)$, where r_{feye} is given by

$$r_{feye} = r_{focus} \frac{(d+1)\frac{r_{ori}}{r_{focus}}}{d\frac{r_{ori}}{r_{focus}} + 1} = r_{focus} \frac{d+1}{d + \frac{r_{focus}}{r_{ori}}}. \qquad (12)$$

Here, d is a constant distortion factor and a larger value of d results in a higher degree of magnification. In this paper, we set $d = 3.0$.

In Table 2 and Figure 12, we show quantitative and qualitative results comparing block distortions for our spherical grid focus and the fisheye focus. For the tornado data set, the focus is set at the center of the volume. We can observe from Figure 12 that the focal region has much less distortion using our method. For the surrounding regions, although both methods lead to some distortion, our method does not gather the distortion around the boundary of the focus. The measured errors shown in Table 2 are consistent with the error image results. The radius of focal region for our spherical block focus is 20 voxels, and that of the fisheye focus is 35 voxels in order to achieve a similar scaling effect. The overall errors are close for both methods, but our method has much smaller average and maximum errors near the focus. For our spherical block focus, the errors are mainly distributed to the blocks that are at least 20 voxels away. This is not the case for the fisheye focus. The blocks located outside of the focal region will not deform at all. All the errors are accumulated within the focal region, especially for the blocks around the boundary of the focal region, which results in the undesired ringing artifact.

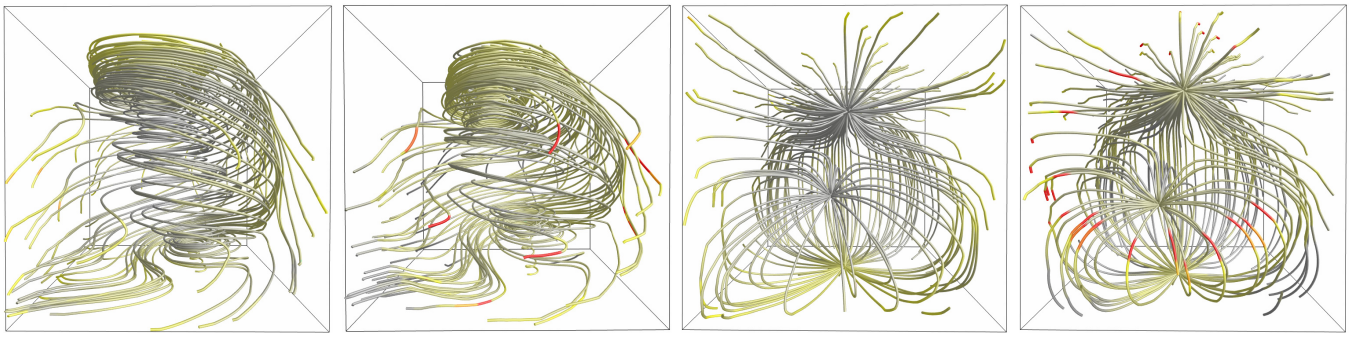
5 Evaluation

5.1 Empirical Expert Evaluation

To evaluate our deformation framework, we collaborate with a domain expert in fluid mechanics (Professor Seung Hyun Kim) whose research focuses on the modeling of multiscale and multiphysics problems in relation to energy science and technology. After learning the framework and using our program multiple times with various data sets, he provided the following feedback. In general, the use of deformation for F+C visualization in flow field exploration is novel and effective. Having multiple methods developed for users to select the focus is a significant advantage. This allows users to determine the best method in their respective cases or even apply multiple methods in a certain order to achieve more desired results. In terms of distortion, both the spherical focus and streamline focus provide better F+C visualization effects than the fisheye view. When the interesting region is at the corner, focusing on a streamline in that region might be more effective than the spherical focus, since the spherical region might go out of bound and lead to inevitable distortion on the boundary. In many cases, he found it very useful to explore the data sets with multiple focus selection methods. For example, the users could first use the streamline focus to enlarge the regions that a streamline goes through. Based on the deformation result, the users might be able to find some small features in those regions and apply the spherical focus to further enlarge the small features. It is also beneficial to use the hourglass focus on the features that are hidden in the central region of volume, e.g., the source around the center in the five critical points data set. That is, the users can use the hourglass focus to push away the streamlines that occlude the interesting features, and then apply the streamline focus or spherical focus to further explore that region of interest.

With the GPU implementation, our program is fairly interactive which allows the users to fine tune the parameters to achieve satisfactory deformation results on the fly. According to the suggestions of the expert, we also modified our initial single view interface to support multiple views, which benefits parameter tuning by eliminating the need to switching between different views back and forth. We allow the users to freely select any three items for simultaneous display. Our experience shows that visualizing the deformed streamlines, original streamlines and deformed grid simultaneously is particularly useful. The connection among these three views is helpful for the users to determine the actual scaling obtained in focused regions and fine tune the distortions accordingly. Even if some distortions are inevitable, they could be easily identified under multiple views and with error indications (introduced in Section 3.4). In addition, as the users get familiar with each parameter, they can predict the changes due to parameter tuning accordingly.

Since one of the main current areas of interest of the expert is the modeling of turbulent combustion, he further commented that the application of this deformation framework to flame visualization could be valuable for investigation. The combustion reactions can be confined into a relatively thin region and be substantially influenced by a flow field, strain



(a) (b) (c) (d)

Fig. 12. Mapping block errors to streamline colors for the tornado and electro data sets. (a) and (c) are with our spherical block focus. (b) and (d) are with the fisheye focus.

or vorticity. Using our deformation framework to emphasize the species concentrations or temperature in the region of high strain rate or vorticity would provide very useful information. The current deformation framework could also be extended to be of further use in two aspects. First, in addition to the vector field, the deformed blocks could also guide the deformation of scalar fields for a mix rendering to provide more context information or apply to F+C visualization of time-varying data sets. Second, a diverse choice for automatic evaluation could be applied to enlarge the regions with any other desired properties, e.g., high vorticity.

5.2 User Study

We also recruited five unpaid researchers for a user study: two postdoctoral scholars, one PhD student, and two master students. All of them are researchers majoring in mechanical engineering with at least one year of research experience in fluid dynamics. The user study was conducted in a lab using the same PC. The PC has a 27-inch monitor with 1920×1080 resolution, where the visualization result occupied an area of 1200×800. The users were first introduced to the concepts of automatic importance evaluation, spherical block focus, hourglass block focus, streamline focus, and error indication. Then they were given the crayfish data set for free exploration to get familiar with the system. They could perform the tasks whenever they felt comfortable. Each study took about two hours to two and a half hours to complete. Although each task could be performed in a few minutes, the users frequently returned to the interface for further verification when writing their comments, which occupied most of the time.

We designed six tasks (T1 to T6). T1 and T2 asked the user to select a deformation method and the viewing direction to best observe the source for the five critical points data set and the flow pattern at the center for the supernova data set, respectively. T3 and T4 asked the user to evaluate the distortion given a deformation result using the crayfish and two swirls data sets, respectively. T5 and T6 asked the user to select a deformation method and reproduce the deformation result given an image of deformation result using the car flow and computer room data sets, respectively. The users were informed that the tasks were not timed and their comments were of crucial importance.

Six criteria (C1 to C6) were given as guidelines for users to evaluate the deformation framework: C1 effectiveness of magnifying features; C2 effectiveness of reducing the occlusion over a feature; C3 ease of noticing the distortion; C4 ease of understanding the relationship between deformed and original streamlines; C5 ease of estimating the original pattern from the deformed one; and C6 ease of reproducing a deformed visualization result. We asked the users to comment on the effectiveness of the proposed F+C techniques for T1, T2, T5 and T6, the distortion evaluation for T3 and T4, and the ease of reproducing a deformation result for T5 and T6. Finally, a set of open questions were also presented which ask the users to provide general impression of the methods for each of the criteria. The deformation result for T3 and T4 can be found in Figure 6 (d). The images presented to the users for T5 and T6 are the bottom right image in Figure 10 and Figure 5 (c), respectively.

In terms of **C1**, the feedback was very positive. Yet the selection of methods varied due to the different foci between disciplines or personal preference. The only exception was T6, where the deformation result to be reproduced was apparently generated using streamline focus. Other than that, each user selected the deformation method to complete the tasks in a consistent way. User 1 used a combination of automatic importance evaluation and spherical block focus for T1, T2, and T5. He commented that *"automatic importance evaluation and spherical block focus combined could provide more magnifying features compared to using any single choice"*. He would finally select spherical block focus to produce the result, since *"spherical block focus does it better because the feature is within a local area"*. User 2 also preferred a combination of multiple methods. He applied automatic importance evaluation and then hourglass block focus for T1, because of *"secluding the source that is obscured by other streamlines"*. For T2 and T5, he used automatic importance evaluation to gain an overview and spherical block focus to magnify a specific region. The other three users would like to apply a single method to perform the task, User 3 preferred hourglass block focus and User 4 and User 5 mostly used spherical block focus. User 5 also used streamline focus for T2, which asked the users to observe the pattern of the supernova data set. The central region was complex and difficult to understand even if it was magnified. He said that the streamline focus *"allowed*

to better understand the features", since most streamlines shared a similar pattern and the analysis should start from understanding one of them.

Although the selections differed among the users, their comments on each method were similar. All of them rated spherical block focus to be the most effective one to magnify features, followed by automatic importance evaluation and streamline focus. They indicated that *"automatic importance evaluation method could find the region of importance"*, but spherical block focus was better, since *"user interaction is involved"* and *"in most fluid flow problems, we are interested in a region in space"*. Spherical block focus was favored over streamline focus mostly because streamline focus only had *"limited ability to magnify the adjacent fluid flow"*. But a user also mentioned that this might be discipline dependent, since *"if the particle tracing is the concern, maybe the streamline focus could do a better job"*. Hourglass block focus was considered to be the least effective one, since it did not magnify a feature.

In terms of **C2**, most users selected spherical block focus or hourglass block focus. Although spherical block focus was not designed to reduce the occlusion, users found that when the influence region was large enough, the outer layer could still get sparser. This would definitely sacrifice the quality of context streamlines, but a user stated that he *"chose to focus more in the point"* rather than *"worry much about the distortion on the boundary"*. In our observation, the users usually used automatic importance evaluation to gain an overall impression of the field, and applied spherical or hourglass block focus for further analysis, since the interaction allow them to specify a region for detail observation.

The error estimation corresponds to our criteria **C3**, **C4**, and **C5**. All users agreed that with error indication, it was easy to notice the distortion. A user even mentioned that he could notice the distortion by just moving the scale slider, and *"the error indication helped me to quantify"*. The ease of understanding the relationship between deformed and original streamlines depended on the complexity of the deformation. For the deformation result using the two swirls data set, a user stated that *"if the deformed style is a regular shape, it is easy to relate to the original streamlines, even when the fluid flow is complex"*. Most users believed it was easy to understand this relationship by moving the scale slider and observing the deformation process, even for relatively complex deformation. But a user also mentioned that the absence of error indication with a small scale value (close to 1.0) *"makes tracking all the distorted streamlines impossible"*, although he admitted that *"it is able and relative easy to track just one streamline"*. Once the relationship between deformed and original streamlines was built, the estimation of original shape of streamlines would not be an issue. Only one user mentioned that *"it is able to make relatively rough estimations, but not into details"*. This is acceptable since the distorted streamlines are mostly in the context.

In terms of **C6**, all users stated it was easy to produce a similar result with comments such as *"it is not difficult to reproduce the results, since the feature of each function is clear and easily recognized"*. The selection of parameters seemed not to be a problem. A user mentioned that *"it may take some time to select the correct cell, and select the correct parameters, but it can be done within a short time"*. Two users also commented that it might be more difficult to reproduce a result generated with hourglass block focus. This was probably because hourglass block focus is view-dependent and has more parameters to adjust.

6 Limitations and Future Work

Our F+C deformation framework has several limitations. First, although we introduce the flow-aware adjustment and smoothing terms to consider the flow pattern, the framework is still based on minimizing block distortion. In some cases, large block distortion might not lead to serious streamline distortion, e.g., when a block containing many straight streamlines is stretched. Our framework could be further improved, if we could define some streamline-based distortion measure. In this way, the minimization should not only preserve the shape of each streamline, but also maintain the perceptual patterns of the flow field. Second, another limitation of our work is parameter setting. In our current implementation, although the default parameter values could work well for most cases, the users still need to adjust those values carefully to achieve the best visualization results for some data sets. Although the user study shows that the users feel that the adjustment of parameters is easy, automatic parameter value selection is still a direction that is worth exploring. Third, our work aims at effectively visualizing the streamlines with moderate density in a limited screen space, where the F+C visualization is achieved by repositioning the streamlines. Its ability to reduce the occlusions of extremely dense streamlines is limited. To handle those cases, other properties of visualization, e.g., transparency, should be incorporated. Fourth, our current error evaluation only produces a scalar value for each block to indicate the distortion. To decide in which directions the distortion is maximal or minimal, we need to compute a tensor field from the deformation field and the resulting eigenvalues describe the amount of the principal distortions.

7 Concluding Remarks

We have presented a deformation framework for F+C streamline visualization that reduces occlusion and clutter around the focal regions for detailed examination while compacting the context region in a full view. Existing solutions for 3D flow visualization resolve the fundamental challenge of occlusion and clutter through displaying a less number of streamlines as the representatives or seeking illustrative rendering solutions using color and opacity modification. We take a very different strategy and achieve this goal through streamline repositioning. Besides automatically selecting focal regions according to derived flow information, we allow the user to manually specify a focal region by clicking a point or a streamline of interest. F+C animation can be achieved by allowing the focal point to follow a user-selected streamline or a user-drawn path from end to end. Such explorations enhance our ability to interact with streamlines for effective visual understanding, making our approach attractive to use in practice.

ACKNOWLEDGMENTS

This research was supported in part by the U.S. National Science Foundation through grants IIS-1017935 and DUE-1105047. We would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] L. Buatois, G. Caumon, and B. Lévy. Concurrent number cruncher - a GPU implementation of a general sparse linear solver. *International Journal of Parallel, Emergent and Distributed Systems*, 24(3):205–223, 2009.

[2] C.-K. Chen, S. Yan, H. Yu, N. Max, and K.-L. Ma. An illustrative visualization framework for 3D vector fields. In *Computer Graphics Forum*, volume 30, pages 1941–1951, 2011.

[3] C. D. Correa, D. Silver, and M. Chen. Illustrative deformation for data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1320–1327, 2007.

[4] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of Eurographics/IEEE TCVG Symposium on Visualization*, pages 239–248, 2003.

[5] A. L. Fuhrmann and M. E. Gröller. Real-time techniques for 3D flow visualization. In *Proceedings of IEEE Visualization Conference*, pages 305–312, 1998.

[6] G. W. Furnas. Generalized fisheye views. *ACM SIGCHI Bulletin*, 17(4):16–23, 1986.

[7] E. Gansner, Y. Koren, and S. North. Topological fisheye views for visualizing large graphs. In *Proceedings of IEEE Symposium on Information Visualization*, pages 175–182, 2004.

[8] R. Gasteiger, M. Neugebauer, O. Beuing, and B. Preim. The FLOWLENS: A focus-and-context visualization approach for exploration of blood flow in cerebral aneurysms. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2183–2192, 2011.

[9] B. K. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[10] E. LaMar, B. Hamann, and K. I. Joy. A magnification lens for interactive volume visualization. In *Proceedings of Pacific Graphics Conference*, pages 223–232, 2001.

[11] R. S. Laramee, C. Garth, H. Doleisch, J. Schneider, H. Hauser, and H. Hagen. Visual analysis and exploration of fluid flow in a cooling jacket. In *Proceedings of IEEE Visualization Conference*, pages 623–630, 2005.

[12] T.-Y. Lee, O. Mishchenko, H.-W. Shen, and R. Crawfis. View point evaluation and streamline filtering for flow visualization. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 83–90, 2011.

[13] O. Mattausch, T. Theußl, H. Hauser, and M. E. Gröller. Strategies for interactive exploration of 3D flow using evenly-spaced illuminated streamlines. In *Proceedings of Spring Conference on Computer graphics*, pages 213–222, 2003.

[14] M. Sarkar and M. H. Brown. Graphical fisheye views of graphs. In *Proceedings of ACM SIGCHI Conference*, pages 83–91, 1992.

[15] O. Sorkine, Y. Lipman, D. Cohen-Or, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 179–188, 2004.

[16] J. Tao, J. Ma, C. Wang, and C.-K. Shene. A unified approach to streamline selection and viewpoint selection for 3D flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):393–406, 2013.

[17] M. van der Zwan, A. Telea, and T. Isenberg. Continuous navigation of nested abstraction levels. In *Short Paper Proceedings of the EG/IEEE VGTC Conference on Visualization (EuroVis 2012, June 5-8, 2012, Vienna, Austria)*, pages 13–17, 2012.

[18] L. Wang, Y. Zhao, K. Mueller, and A. E. Kaufman. The magic volume lens: An interactive focus+context technique for volume rendering. In *Proceedings of IEEE Visualization Conference*, pages 367–374, 2005.

[19] Y.-S. Wang, T.-Y. Lee, and C.-L. Tai. Focus+context visualization with distortion minimization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1731–1738, 2008.

[20] Y.-S. Wang, C. Wang, T.-Y. Lee, and K.-L. Ma. Feature-preserving volume data reduction and focus+context visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):171–181, 2011.

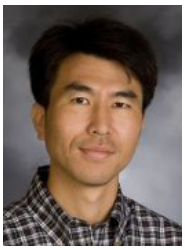
[21] L. Xu, T.-Y. Lee, and H.-W. Shen. An information-theoretic framework for flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1216–1224, 2010.

[22] X. Zhao, W. Zeng, X. Gu, A. Kaufman, W. Xu, and K. Mueller. Conformal magnifier: A focus+context technique with local shape preservation. *IEEE Transactions on Visualization and Computer Graphics*, 18(11):1928–1941, 2012.

Jun Tao is a PhD student of computer science at Michigan Technological University. His research interests include flow visualization, image resizing, and mesh editing. He received a BS degree in software engineering from Sun Yat-sen University, China, in 2008, and a MS degree in computer science from Michigan Technological University in 2010. He is a student member of the IEEE.

Chaoli Wang is an assistant professor of computer science at Michigan Technological University. His research focuses on large-scale data analysis and visualization, high-performance computing, and user interfaces and interaction. He received BE and ME degrees in computer science from Fuzhou University, China, in 1998 and 2001, respectively, and a PhD degree in computer and information science from The Ohio State University in 2006. From 2007 to 2009, he was a postdoctoral researcher at the University of California, Davis. He has served on the program committees of the IEEE SciVis, EuroVis and IEEE PacificVis. He is a member of the IEEE.

Ching-Kuang Shene is a professor of computer science at Michigan Technological University. His research interests include geometric modeling, mesh processing, software visualization, and computer science education. Shene has a PhD degree in computer science from The Johns Hopkins University in 1992, and is a member of ACM, AMS, Eurographics, IEEE/CS, MAA and SIAM.



Seung Hyun Kim is an assistant professor of mechanical engineering at Michigan Technological University. His research focuses on the modeling of multiscale and multiphysics problems in relation to energy science and technology. He received BS, MS, and PhD degrees in Mechanical Engineering from Pohang University of Science and Technology, South Korea, in 1996, 1998, and 2003, respectively. He was a postdoctoral researcher (2004-2008) and a research associate (2008-2010) at Stanford University.