Stain Formation on Deforming Inelastic Cloth

Xiaojun Wang, Shiguang Liu, *Member, IEEE,* and Yiying Tong *Member, IEEE*

Abstract—We propose a novel approach to simulating the formation and evolution of stains on cloths in motion. We accurately capture the diffusion of a pigmented solution over a complex knitted or woven fabric through homogenization of its inhomogeneous and/or anisotropic properties into bulk anisotropic diffusion tensors. Secondary effects such as absorption, adsorption and evaporation are also accounted for through physically-based modeling. Finally, the influence of the cloth motion on the shape and evolution of the stain is captured by evaluating the inertial (e.g., centrifugal and Coriolis) forces experienced by the solution. The governing equations of motion are integrated in time directly on a deforming triangle mesh discretizing the inelastic cloth for efficiency and robustness. The deformation of the cloth can be precomputed or integrated through simplified two-way coupling, by using off-the-shell cloth simulations. Finally, numerical experiments demonstrate the plausibility of our results in practical applications by reproducing the usual shape and behavior of stains on various fabrics.

Index Terms—Cloth animation, inhomogeneous and anisotropic cloth material, stain formation, deforming surfaces.

1 Introduction

Despite the large number of physically-based methods for both cloth animation and fluid simulation, very limited effort has been devoted to the simulation of stain formation on cloth in motion: the complexity in accounting for inertial forces and anisotropic effects that the absorption, adsorption, and diffusion of a liquid solution on knitted or woven textiles involve has thwarted the efficient simulation of such phenomena. However, with the rapid improvement in computing power and the advent of virtual worlds, simulating the formation and evolution of stains on moving cloth could have a wide range of applications in computer games, special effects in movies, digital arts, and augmented reality.

Stains on fabric, textile, or even paper are formed when a pigmented fluid such as ink, wine, or water gets into the fibers of the medium, starting a process of diffusion, absorption, adsorption, and evaporation. The solution, which forms stains, consists of solute and solvent, typically, pigment particles and water/alcohol respectively. When a drop of such solution falls on the surface of cloth, it begins to spread throughout the cloth involving the macroscale motion of the liquid and the diffusion of both solute and solvent. Water will evaporate into open air, in the cells with non-zero solvent density. The solution can enter the space in the fabric, i.e., become absorbed. The solute and sometimes the solvent can also form a thin layer adhered to the surface, i.e., becomes adsorbed. Since knitted or woven cloth materials are often highly anisotropic and inhomogeneous, the evolution of a stain in time depends heavily on the pattern of threads or yarns of the textile. Moreover, the motion of the cloth itself can have a dramatic effect on the stain formation: inertial forces (be they centrifugal or Coriolis) that the fluid experiences will bias the propagation of the stain, adding further complexity to the phenomenon.

We present a simple physically-based approach to capture the evolution of a fluid density (i.e., stain) over a moving and deforming cloth. Using homogenization theory, we construct the bulk anisotropic diffusion tensor based on the composition and the weaving pattern of the fabric. The resulting diffusion tensor is mapped onto the cloth by specifying the local alignment of the fabric. The fluid motion is then calculated directly on the mesh by storing the velocity field as a tangent vector field. The velocity field is influenced by the movement of the deforming cloth through locally estimated inertial forces. Interactions between stain and textile including absorption, adsorption and evaporation are also modeled by keeping track of the density fields of various components of the solute and solvent. Our approach generates realistic visual simulation of stains, which we demonstrate by showing complex stain evolutions on deforming cloth made of a variety of inhomogeneous and anisotropic materials.

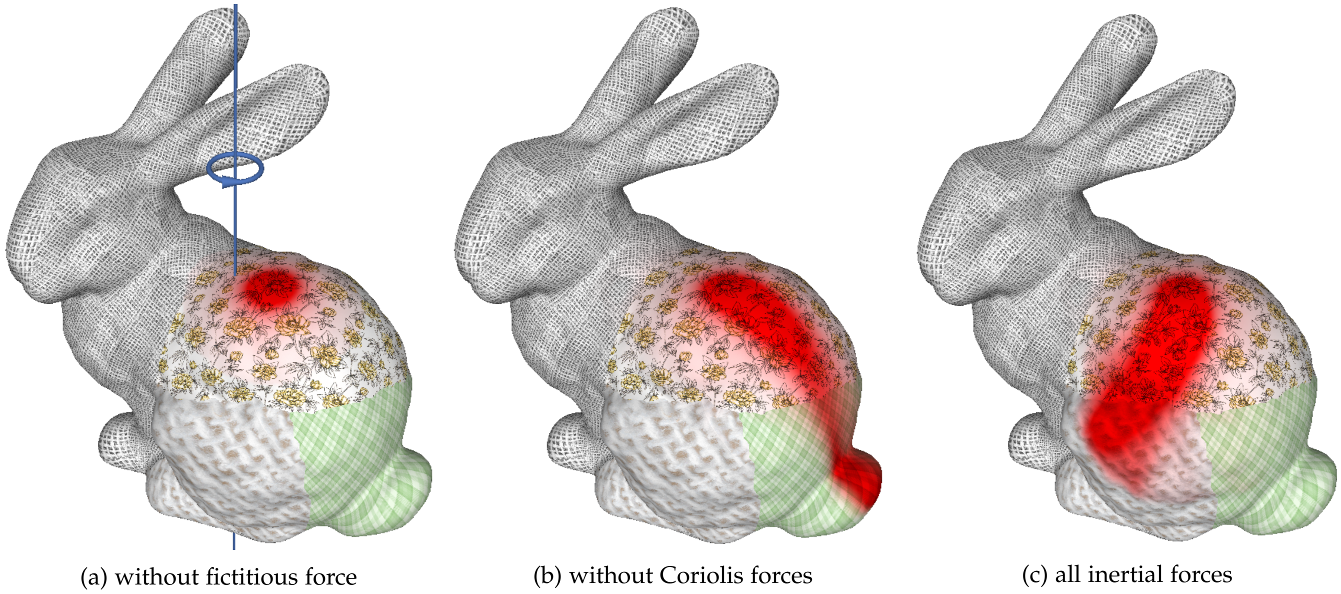
1.1 Related Work

Stain formation relates to many research efforts in computer graphics. We briefly cover the most relevant works here to motivate our technique and its various components.

Ink simulation over planar fibrous medium. Over the years, researchers have developed numerous efficient techniques for 3D fluid simulation in graphics [1]. Recently, reduced models for fluid simulation on surfaces have received increased attention due to their efficiency compared to a full-blown simulation of the 3D Navier-Stokes equations. In particular, work has been dedicated to simulate watercolor painting and Chinese ink painting, both of which involving diffusion of pigments in paper. Curtis et al. [2] simulated watercolor effects such as dry-brushing, intentional backruns, and flow patterns. Kunii et al. [3], [4] modeled the interactions between ink and paper with partial differential equations, which are essentially Fick's law of diffusion. Chu et al. [5] simulated ink dispersion in absorbent paper by solving the lattice Boltzmann equation. Van Laerhoven et al. [6], [7] presented a physically based technique for

- *S. Liu is with the School of Computer Science and Technology, Tianjin University, Tianjin, China. E-mail: lsg@tju.edu.cn*
- *X. Wang and Y. Tong are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824.*

Manuscript received Feb 14, 2017.



(a) without fictitious force (b) without Coriolis forces (c) all inertial forces

Fig. 1: Stain on a bunny rotating around a vertical axis.

creating images with watery paint in real time. Simulating Chinese calligraphy and Chinese landscape painting based on ink diffusion on paper has also been proposed by various researchers [8]–[12]. Morimoto et al. [13] presented a method for visualization of cloth dyeing. However, their method focused on simulating the dyeing process, and is not suited for the stain formation on textile surfaces. Liu et al. [14] proposed a simulation of stains on flat anisotropic media, including a simplified treatment of diffusion, infiltration, and evaporation. Liu and Chen [15] presented an extension for simulating hand-painted printing patterns on cloth based on the physical process of the ancient folk handicraft. While all these methods can generate realistic simulation of pigment distributions in flat 2D domains, they often use no or very simplified fluid advection, and cannot be directly extended to model stain evolution on moving and deforming cloth.

Fluid and droplet simulation on curved surfaces. Fluid motion on non-flat surfaces was simulated by Stam [16] through the use of 2D parameterization, which can induce noticeable artifacts. Intrinsic (i.e., parameterization-free) methods were later proposed to simulate surface flows directly on triangle meshes [17]–[20], and a model-reduced approach based on eigenvector bases has been recently offered as well [21]. Auer et al. [22] leveraged the Closet Point Method (CPM) to numerically approximate the wave equation and the incompressible Navier-Stokes equations on arbitrary surfaces in realtime on GPU. Wang et al. [23] presented a technique to simulate shallow water equations on the surfaces, targeting surface tension driven effects, after introducing a simulation of water drops on hydrophobic or hydrophilic leaves [24]. Zhang et al. [25] offered a fast Lagrangian method for such droplet simulations using triangle meshes to represent the drops, while Jung and Behr [26] introduced GPU-based real-time simulation of droplet flows. Finally, Djado et al. [27] simulated the motion of water drops on a surface, realistically capturing condensation on a surface or human sweating in real time.

Treatment of fields on non-flat surfaces was also studied extensively in geometry processing. For example, de Goes et al. [28] introduced a representation of arbitrary 2-tensor fields on triangle meshes. Azencot et al. [29] proposed a novel discretization for the covariant derivative of vector fields on discrete surfaces with various appealing properties. Liu et al. [30] presented a unified discretization of the covariant derivative that offers closed-form expressions for both local and global first-order derivatives of vertex-based tangent vector fields on triangulations. Clarenz et al. [31] provided a multi-scale method in surface processing by incorporating anisotropic diffusion equations. Singer and Wu [32] presented vector diffusion maps, an algorithmic and mathematical framework for analyzing data sets, where scalar affinities between data points are accompanied by orthogonal transformations. Diamanti et al. [33] designed N-PolyVector fields, including their topology, parallel transport, smoothness and singularities, with complex polynomials. Recently, Ren et al. [34] presented an efficient simulation approach reproducing sophisticated surface flow phenomena based on the shallow-water equations.

Flows on deforming surfaces. Recently, flow simulation on moving surfaces has started to emerge. For instance, Angst et al. [35] proposed a method for wave simulations on deforming meshes, while Neill [36] developed a framework for simulating fluid flows on deformable surfaces, where only the average acceleration of the underlying surface is considered for the calculation of inertial forces. Hegeman et al. [37] presented an approach to solving Navier-Stokes equations on surfaces based on the unique properties of conformal cube maps. Jeong and Kim [38] also introduced a combustion model of heat transfer and fuel consumption for the propagation of a fire front on a point cloud surface. They proposed angular Voronoi weights for a discrete Laplace-Beltrami operator that shows better isotropic diffusion on inhomogeneous point clouds than the cotangent or moving least-squares schemes. While all these methods treat curved, deforming surfaces composed of either homogeneous or isotropic media, they do not account for the full inertial effects of the surface motion, and are thus not directly applicable for stain formation on moving surfaces with inhomogeneous and anisotropic materials.

Dealing with inhomogeneous and anisotropic materials. In order to efficiently handle highly inhomogeneous and/or anisotropic materials without having to capture the microscopic features of a medium, homogenization theory [39], [40] offers various ways to replace the microscopic structures by an effective locally homogenous material with similar macroscopic properties. In particular, Owhadi et al. [41] proposed a general homogenization technique which optimally simulates the macroscopic effect of an elliptic equation in divergence form with highly varying microscopic material properties. This idea was extended to deal with elasticity homogenization to approximate deformable objects made of arbitrary fine structures of various linear elastic materials with a dynamically-equivalent coarse model [42] (see also [43] for a localized homogenization technique). We adopt this class of homogenization methods in our approach for efficiency.

1.2 Contributions

We offer a complete simulation pipeline for the simulation of fluid-induced stain formation and evolution on deforming inelastic cloth with inhomogeneous and/or anisotropic material properties such as weaving patterns. The complex time evolution of the pigment density over the cloth is computed through a mixed explicit-implicit integrator using a multistep splitting method. Figure 2 illustrates an overview of our pipeline: we preprocess the inhomogeneous (and possibly anisotropic) cloth material through homogenization to get effective diffusion tensors; we then specify material properties directly on a triangle mesh representing a discretization of the cloth; the fluid motion is computed in a splitting fashion as in semi-Lagrangian simulation [1], including fluid acceleration due to inertial forces (i.e., centrifugal and Coriolis forces due to the local acceleration of the cloth) and gravity, advection, diffusion, infiltration, adsorption, and evaporation. All steps are handled though explicit integration, with the exception of the (parabolic) diffusion process for which an implicit integration is used to enforce stability. Our exposition basically follows the flowchart of our pipeline.

The main contributions of our method include:

- a homogenization procedure for evaluating bulk diffusion tensors of fabrics with arbitrary knitting patterns;
- a modified anisotropic Laplacian matrix to simulate velocity diffusion directly on curved surfaces;
- an efficient evaluation of fictitious (i.e., inertial) forces induced by the coupling between surface and fluid, including the Coriolis effects based on local comoving and corotational frames;
- a framework incorporating realistic stain evolution, including absorption, adsorption, and evaporation.

Note that many of the components of our framework are not novel, but we adapted them for the specific application of stain simulation on deforming surfaces. Homogenization is extensively used and tested in other contexts in physical simulation, although not for stain formation. Anisotropic scalar field diffusion has also been developed in various forms, but our anisotropic velocity diffusion is novel to the best of our knowledge. Inertial forces were also used in fluid simulation, but the Coriolis effects have been overlooked for corotating local frames on surfaces in existing methods. We provided detailed explanation for absorption, adsorption and evaporation as implementation details, but we do not claim novelty on these procedures.

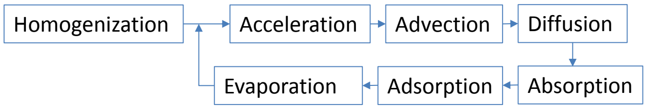
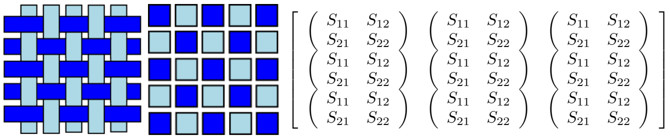


Fig. 2: Flowchart of our simulation framework.

2 Homogenizing Bulk Material Properties

We represent fabric (or any other inhomogeneous material) by a weaving pattern of cells for efficient simulation. As described in [13], there are three types of cells in a textile model including warp cell, weft cell and gap cell. Figure 3b shows an example distribution of different cells for a swatch of fabric (Figure 3a). A given weaving pattern determines whether each warp (weft) cell is orientated up or down. By changing the arrangement of warp cells and weft cells, we can approximate the structure of textiles with different weaving styles such as plain weave, satin weave and twill weave. Gap cells, instead, denote the gaps among warps and wefts. Note that the diffusion of a fluid stain can be different between different pairs of cells. For instance, stains over a fabric with hydrophobic fibers may diffuse quickly between gap cells, but slowly between warp and weft cell.

Homogenization consists in establishing a bulk diffusion tensor that effectively averages the diffusion coefficients of the different cells on a piece of fabric—so that the overall anisotropic diffusion of stains in the fabric is properly captured without resorting to the computationally-intractable modeling of each individual thread. Figure 3c gives the basic layout of an input diffusion tensor field. A 2×2-matrix is used to denote the diffusion property of each cell. Each element in the matrix can be obtained according to the layout of the cells and the diffusion characteristics of the textile such as tortuosity and porosity. One method to obtain the input tensor field is to use the Weisz-Zollinger model (see, e.g., [44]), which defines five types of connectivity based on cell positions and porosity: a) fibers in different layers, b) perpendicular fibers in the same layer, c) fiber and gap, d) gap and gap, and e) parallel fibers in the same layer. The permeability (diffusion) coefficient between two textile cells is computed as $D = K_{\text{diff}}T$, where K_{diff} is a coefficient determined by the properties of both the stain (dynamical viscosity) and textile (porosity), and T is the tortuosity, which can be different for different types of cell connectivity. We allow the user to choose between specifying a pattern of diffusion tensors, or specifying a cell pattern and D. In



(a) Pattern. (b) Cells. (c) Tensor layout.

Fig. 3: *2D homogenization: (a) Weaving pattern. Dark: Weft. Light: Warp. (b) Cell-based approximation. (c) Layout of diffusion tensor for one block of the period domain.*

the latter case, we average the boundary permeability to construct the diffusion tensor through

$$S = 1/2\ [D_{\text{left}} + D_{\text{right}}, 0; 0, D_{\text{top}} + D_{\text{bottom}}].$$

The original highly inhomogeneous material property can only be resolved with an extremely high mesh resolution. By using the homogenization theory [39], we can simulate the long-time behavior effectively and avoid the high frequency tensor field. We assume that the material pattern is periodic, and the user input swatch is a representative tile of this pattern. We construct a mesh of a topological torus with the specified tensor field. An "effective" tensor, which produces a similar diffusion in a region containing a large number of tiles, can then be evaluated as

$$S^{\text{eff}}_{ij} = \frac{1}{|\Omega|}\int_{\Omega} [S(x,y)(e_i + \nabla f_i(x,y))] \cdot e_j dxdy, \qquad (1)$$

where $|\Omega|$ is the total area of domain Ω, S^{eff} is the effective bulk diffusion tensor, $S(x,y)$ is the diffusion tensor field, e_i ($i = 0, 1$) is the unit vector in i-th direction, and f_i is the solution to the following Poisson equation

$$\nabla \cdot [S(x,y)(\nabla f_i(x,y) + e_i)] = 0. \qquad (2)$$

Roughly speaking, $(f_1 + x, f_2 + y)$ is a new (harmonic) parameterization of the domain, in which the diffusion tensor is properly stretched. As suggested in [41], the heterogeneous multi-scale structures can be accurately simulated at a coarse level under the assumptions of ergodicity and scale separation by transferring bulk (averaged) information from sub-grid scales to computational scales. The diffusion tensor is a special case of such heterogeneous structures, and can thus be averaged this way. In 1D, this is the same as taking a simple harmonic mean, but in 2D, this expression has to be evaluated numerically in general. The discretization of Equation 2 will be discussed when we address the simulation of diffusion in Sec. 3.2.

3 Stain simulation on inelastic cloth

The motion of a solution (i.e., a solute within a solvent) along the fibers of a fabric is governed by the restricted Navier-Stokes equations, in which the normal velocity is ignored; i.e., we can assume that the normal acceleration is balanced by the constraint that the fluid stays on the surface. We can thus express the motion as a set of partial differential equations:

$$\dot{u} + u \cdot \nabla u = \nabla \cdot (S_v \nabla u) - \nabla p_{\text{ext}}/\rho + a_{\text{ext}}, \qquad (3)$$
$$\dot{\sigma}_f + \nabla \cdot (\sigma_f u) = \nabla \cdot (S_f \nabla \sigma_f) + I_f, \qquad (4)$$
$$\dot{\sigma}_s + \nabla \cdot (\sigma_s u) = \nabla \cdot (S_f \nabla \sigma_s) + I_s, \qquad (5)$$

where $\dot{x} = \partial x/\partial t$ denotes the time derivative, u is the tangential velocity field, ρ is the density of the incompressible fluid, p_{ext} is the external pressure (ignored in our tests), a_{ext} is the external body force (including tangential components of the gravity and inertial forces as detailed in Sec. 4), σ_f is the solvent fluid density (per unit area), σ_s is the solute density, and I_f (I_s) are the interaction terms that we will review in Sec. 5. Aside from the solute density, the above equations are equivalent to the generalized shallow wave equations in [23] with $\sigma_f = H\rho$, where H is the height of the solvent in the normal direction. We also assume that the motion of the solvent is not influenced by the solute, and that the viscosity/permeability tensors are different for the tangent velocity field (S_v) and the densities(S_f). These tensors are evaluated through the aforementioned homogenization procedure, as we assume that the fluid is partially flowing on the surface and partially permeating through the porous media.

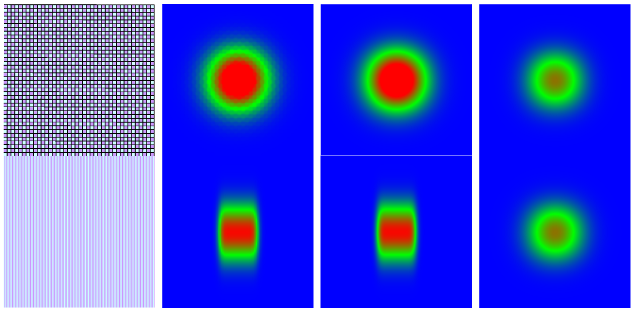


Fig. 4: From left to right: layout, diffusion with original tensor field, diffusion with effective tensor, and diffusion with direct average tensor.

As shown in Figure 2, we follow a typical splitting method for time integration as in [1]. In each iteration, we first compute the external forces (Sec. 4); then, we advect the velocity and solvent and solute densities (Sec. 3.1); next, we perform the diffusion processes (Sec. 3.2); and finally, we handle the additional interaction terms, adsorption and evaporation (Sec. 5). For boundaries between different weaving patterns, the effects are treated through the accumulation of one-ring triangles, each of which belongs to one pattern. For boundaries of the surface, we follow no-transfer and free-slip boundary conditions, and set the normal component of the velocity to 0 and allow the tangent component to slip along the boundary.

3.1 Advection

Our velocity advection essentially follows the procedure described in [36]: the surface is triangulated and the velocity field is discretized as one 2D vector $u_i = (u_i^x, u_i^y)^\top$ per vertex in the XY-plane of the tangent frame F_i stored at vertex i. F_i is defined by setting the direction z_i to be the area-weighted average unit normal of the one-ring, $y_i = (z_i \times e_{ij})/|z_i \times e_{ij}|$, where e_{ij} is one of the incident edges of vertex i, and $x_i = y_i \times z_i$.

The key difficulty in the velocity advection is the calculation of ∇u, which, on a curved surface, is the covariant derivative of u. It provides a way to compare u_i and u_j for adjacent vertices, since a direct differencing of the components would not produce the true coordinate-frame independent "vector gradient". One approximation of ∇u is to map the 2D tangent field to 3D through F_i for comparison; however, much numerical diffusion would be introduced as the edges are not infinitesimally small. Thus, we follow the approach proposed in [36] and use a 2×2 rotation matrix R_{ij} to align the vector expressed in F_j to F_i. The 2×2 rotation matrix R_{ij} can be regarded as parallel transporting a vector from vertex j to vertex i along the edge connecting the two. Thus $R_{ij} u_j - u_i$ produces the covariant derivative ∇u integrated along the edge. To obtain the rotation angle for the 2D rotation matrix, we map the 1-ring of vertex i to a flat 2D topological disk through the geodesic polar map by rescaling the sum of tip angles to 2π. If the local edge e_{ik} is the chosen edge to construct F_i, the direction of the edge e_{ij} can be represented by the angle α_{ij} that it forms with the chosen x-axis in the 2D domain. Assuming the sum of tip angles for all the triangles adjacent to vertex i is γ_i, α_{ij} is the sum of all the tip angles of the triangles between e_{ik} and e_{ij} in the counterclockwise order rescaled by $2\pi/\gamma_i$. Likewise, e_{ji} makes an angle α_{ji} with the x-axis in the local frame of vertex j. Thus, the rotation matrix R_{ij} can be expressed through the angle $\theta_{ij} = \alpha_{ij} - \alpha_{ji} + \pi$, and we have:

$$R_{ij} = \left[\begin{array}{cc} \cos\theta_{ij} & -\sin\theta_{ij} \\ \sin\theta_{ij} & \cos\theta_{ij} \end{array} \right]. \tag{6}$$

Following the semi-Lagrangian advection method [1] on the surface, we first backtrack vertex i within in its flattened one-ring using the current velocity u_i by a time step h, evaluating the barycentric coordinates $(\lambda_i, \lambda_j, \lambda_k)$ within the triangle t_{ijk}, and update the velocity u_i by:

$$u_i^{t+h} = \lambda_i u_i^t + \lambda_j R_{ij} u_j^t + \lambda_k R_{ik} u_k^t.$$

Note that one can extend this method to include the case when the backtracked point is outside of the one-ring by simply following the strip of triangles being traversed, so there is no time step size limitation to this semi-Lagrangian approach.

For density field advection, we use a finite-volume approach instead to enforce mass conservation of both solvent and solute. Since σ_f and σ_s advect in the same fashion, we use σ_s to illustrate the process. We discretize the solute density by assigning a value $\sigma_{s,i}$ for the Voronoi region of vertex i, and we update it via

$$\sigma_{s,i}^{t+h} = \sigma_{s,i}^t - h/V_i \sum_{j \in N(i)} F_{ij},$$

where $N(i)$ is one-ring of vertex i, V_i is the Voronoi region area, and F_{ij} is the flux through the interface between the Voronoi regions of i and j. This flux is evaluated through

$$F_{ij} = \sigma_{ij}^{\uparrow} (u_{ij} \cdot e_{ij}) w_{ij},$$

where $u_{ij} = 1/2(u_i^x x_i + u_i^y y_i + u_j^x x_j + u_j^y y_j)$, with (x_i, y_i) denoting the local tangent plane in frame F_i, $\sigma_{ij}^{\uparrow}$ is the upwind density (i.e., $\sigma_{s,i}$ if $u_{ij} \cdot e_{ij} > 0$ and $\sigma_{s,j}$ otherwise), and w_{ij} is the usual cotan weights, measuring the ratio between the dual Voronoi edge length and $|e_{ij}|$,

$$w_{ij} = -\frac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij}), \tag{7}$$

where α_{ij} and β_{ij} are the opposite angles of e_{ij} in the two incident triangles.

3.2 Diffusion

As our diffusion is performed directly on the curved triangulated surface representing an inelastic cloth, we need to specify the material on the mesh via a homogenized diffusion tensor per triangle. The fabric geometry can be specified by the weft and warp directions. Aside from genus-1 surfaces, the surface cannot be covered by one consistent weaving pattern, so we partition the surface into a few patches, each with a smooth weft direction field e that induces the warp direction as its 90° rotation $e^{\perp}$. In our tests, we specify these patterns manually. For more practical designs, one may use tools such as [45]. We delay the discussion on how to calculate these fields to the end of this section, since it is essentially using the same discretized operators involved in the diffusion process. The main operator in this process is the modified Laplacian operator $\nabla \cdot S \nabla$, which, using a piecewise linear finite element method, can be discretized as a linear operator L_S, i.e., a matrix with elements:

$$L_{S,ij} = \sum_{e_{ij} \in T} \int_T \nabla\phi_i \cdot (S_T \nabla\phi_j),$$

where ϕ_i is the linear basis function for vertex i, and $S_T = (e, e^{\perp}) S (e, e^{\perp})^{\top}$ is the diffusion tensor aligned to the specified direction field within triangle T. Note that for a diffusion tensor S equal to identity, L_S reduces to the usual cotan formula as expected since it leads to a uniform isotropic diffusion.

In order to remove stringent constraints on time step sizes, temporal discretization of the diffusion process is performed using an implicit integration,

$$M(\sigma^{t+h} - \sigma^t)/h = L_S \sigma^{t+h},$$

where M is the mass matrix with $M_{ij} = \sum_{e_{ij} \in T} \int_T \phi_i \phi_j$, which is often simplified through mass lumping to become just a diagonal matrix with M_{ii} being one third of the one-ring area for vertex i. Thus, the above diffusion process is turned into a linear system: $(M + hL_S)\sigma^{t+h} = M\sigma^t$.

The diffusion process of the velocity field involves the covariant derivative ∇u as the one described in the advection process. However, using the same discretization of covariant derivative with rotations R_{ij} that align the tangent vectors at vertices i and j, we can approximate the Bochner Laplacian in the metric of S_v by replacing each entry in the $N \times N$-matrix (N is the number of vertices in the patch) $L_{S,ij}$ with a 2×2-matrix $L_{S,ij} R_{ij}$ with the term R_{ii} set to the identity matrix, and obtain a $2N \times 2N$-matrix. Implicit integration results in the following linear equations

$$M_{ii}(u_i^{t+h} - u_i^t) = h \sum_{j \in N(i)} L_{S_v,ij} \left(R_{ij}\ u_j^{t+h} - u_i^{t+h}\right). \tag{8}$$

Finally, in order to obtain a smooth weft direction field e, we simply use (at least) one user specified direction for each patch, and solve the discretized version of the equation $\nabla \cdot \nabla e = 0$ while enforcing the user constraint(s).

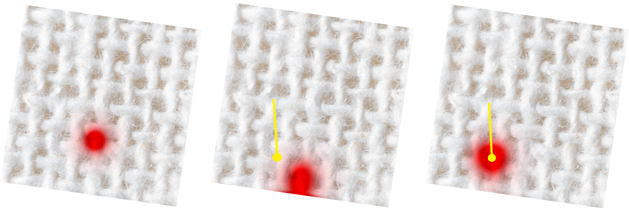


Fig. 5: The comparison shows that both translational (missing in a) and rotational (missing in a and b) inertia forces are necessary for the comoving and corotational frames. A solid ball is indicates the motion of the center of mass (without diffusion) in an inertial coordinate system.

4 Fictitious Forces

An important dynamical aspect of stain formation on cloth is the effects of inertial forces, often referred to as fictitious forces as they do not arise from any direct physical interaction, but from the acceleration the non-inertial reference frame of the cloth itself. Thankfully, we can easily evaluate these forces from the movement of its local frames in time.

For a deforming triangle mesh, each frame F_i attached to a vertex i is time dependent. It can be seen as a comoving and corotating frame for the one-ring neighborhood. Similar to [46], we compute the rotation U_i that best aligns the original frame $\bar{F}_i$ at vertex i to its current frame F_i (i.e., that minimizes the difference between F_i and $U_i\bar{F}_i$) through:

$$U_i = \operatorname*{argmin}_{R\in SO(3)} \sum_{j\in N(i)} [(v_j - v_i) - R(\bar{v}_j - \bar{v}_i)]^2,$$

where $SO(3)$ is the set of all 3D rotations, v_i and $\bar{v}_i$ are respectively the current and original locations of vertex i. Instead of performing a polar decomposition as [46], we directly find the minimum using the Kabsch procedure in [47]. Assuming that the position of vertex i is p_i, the inertial force (acceleration) experienced by a moving object at location r in the frame F_i centered at p_i can be expressed in this frame as the tangential components of

$$a_{\text{inertia},i} = -F_i^\top \ddot{p}_i - F_i^\top \ddot{F}_i r - 2F_i^\top \dot{F}_i \dot{r},$$

where the second term (the sum of the centrifugal force and the Euler force) on the right hand side vanishes (since the fluid velocity is measured at the vertex which corresponds to $r = 0$), the first term is called the linear inertia force due to the translation of the local frame, and the last term is the Coriolis force due to the rotation of the local frame and the local velocity $\dot{r} = u$. In mechanics and physics, the Coriolis force, was first described by Gustave-Gaspard Coriolis in 1835, which acts on objects that are in motion relative to a rotating reference frame. The effects of Coriolis force are to pull the trajectory to bend to the right for counterclockwise rotation of the reference frame, or to the left for clockwise rotation, in order to keep the trajectory straight in a nonrotating inertial frame. For instance, due to the Coriolis effects, large scale cyclones rotate in opposite ways in the Northern and Southern hemispheres.

The total body force is given by

$$a_{\text{ext}} = a_{\text{inertia}} + (g - (g\cdot n)n),$$

where g is the gravitational acceleration and n is the surface normal.

Given the motion of the mesh stored as a sequence of vertex locations, we can easily calculate the linear inertia force by centered differencing,

$$\ddot{p}_i^t = \frac{p_i^{t+h} - 2p_i^t + p_i^{t-h}}{h^2}.$$

As $F^\top \dot{F}$ for F in the rotation group $SO(3)$ is in the Lie algebra $\mathfrak{so}(3)$, i.e., is an antisymmetric 3×3 matrix, we evaluate the Coriolis force as $2\ \omega\times\binom{u_i}{0}$, where ω is found through

$$\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \frac{(F^t)^\top F^{t+h} - (F^{t+h})^\top F^t}{2h},$$

where we antisymmetrized the first-order estimate $(F^t)^\top(F^{t+h} - F^t)/h$ of $F^\top \dot{F}$ to get a proper antisymmetric matrix corresponding to the cross product with the angular velocity ω.

5 Stain-Surface Interaction

In addition to diffusion, the solution interacts with the fabric through penetration, adsorption and evaporation, and both solvent and solute are involved. The solute is separated into three parts: one dissolved in the solution, one absorbed by the textile fibers, and the other deposited in the fibers. The solvent is partially absorbed by the fabric fibers, and it also partially evaporates.

For the absorption process of the solution by the fabric, the penetration depth and the absorption speed can be evaluated through the Lucas-Washburn equation [48], which indicates that this speed is inversely proportional to the penetration depth. We use a simplified formula with the absorption speed dependent on how far the surface is from becoming saturated, assuming that the surface is thin and thus reaches the saturation capacity σ_{sat} quickly. We calculate the absorbed solvent density σ_p as

$$\dot{\sigma}_p = K_p(\sigma_{\text{sat}} - \sigma_p),$$

where K_p is the absorption speed constant. We can use a simple explicit scheme at each vertex by writing

$$\sigma_p^{t+h} = \sigma_p^t + \min(K_p h/(1 + K_p h)(\sigma_{\text{sat}} - \sigma_p^t), \sigma_f^t),$$

where σ_f is the maximum amount of solvent left to be absorbed. We also maintain the absorbed solute density $\sigma_{s,p}$, which evolves as

$$\dot{\sigma}_{s,p} = K_p(\sigma_{\text{sat}} - \sigma_p)\sigma_s/\sigma_f.$$

The absorbed solvent and solute go through the same adsorption and evaporation process as their free flowing counterparts, but they are assumed to be held in place and not to be diffusing much.

Adsorption is the process in which the solute is gradually deposited into the textile fibers, making it difficult to be dissolved again by the solvent. We model this effect via the Langmuir adsorption theory [49], which assumes monolayer adsorption, so the adsorbent will not adsorb further after the adsorbate is covered. Before reaching adsorption

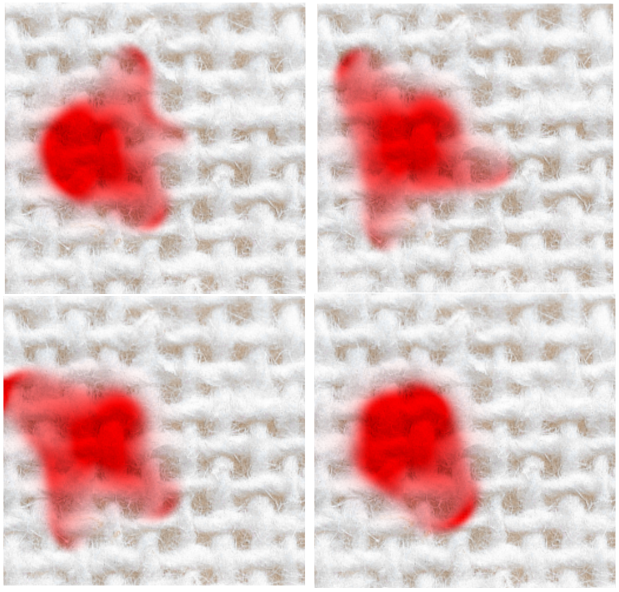


Fig. 6: The stain induced by various initial fluid velocities.

equilibrium, the adsorption rate is proportional to the area of the blank surface, while the desorption rate is proportional to the coverage. We can calculate the adsorption rate and the desorption rate as $K_a V_d (1-\theta) C$ and $K_d V_d \theta C$ resp., where V_d is the total adsorption capacity determined by the material and porosity, θ denotes the coverage of the surface, C represents the concentration of the single layer adsorbed solute, while K_a and K_d are the adsorption rate and desorption rate, respectively. When the adsorption is balanced with desorption, we obtain the maximum adsorption capacity $A_d = V_d K/(1+K)$, where $K = K_a/K_d$. We calculate the change of adsorbed solute $\sigma_{s,a}$ as the difference between adsorption and desorption, which is equal to

$$\dot{\sigma}_{s,a} = K_d (A_d - \sigma_{s,a}). \tag{9}$$

This differential equation is explicitly discretized exactly as in the absorption process.

Finally, we also must account for the fact that the solvent in the solution evaporates. The change of the amount of solvent is directly proportional to the surface area, $\dot{\sigma}_p = -K_{\text{evap}}$, where K_{evap} is the evaporate coefficient. However, the effective exposed area of the boundary cells is greater than that of the inside cells. So the evaporation of the boundary cell will be faster, and the above calculation should be changed into

$$\dot{\sigma}_f = -a_{\text{evap}} K_{\text{evap}}, \tag{10}$$

where a_{evap} is boundary coefficient (we use 1.2 in all our tests). Note that the amount evaporated in each step is bounded by the total amount left.

The combined effects can be formulated as

$$\text{interact}_f = -K_p(\sigma_{\text{sat}} - \sigma_p) - a_{\text{evap}} K_{\text{evap}} \tag{11}$$

$$\text{interact}_s = -K_p(\sigma_{\text{sat}} - \sigma_p)\sigma_s/\sigma_f - K_d(A_d - \sigma_{s,a}). \tag{12}$$

We have now reviewed all the variables involved in the entire system, including the velocity field, solvent and solute on the fabric, absorbed solvent, absorbed solute, and adsorbed solute, and all of which are updated in each time step as we described.

6 Results

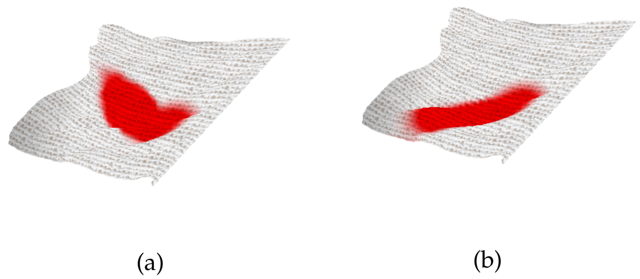
We now present a series of results of our integrator of stain evolution on inelastic cloth. We implemented our framework and performed our tests on a Windows 7 system with Intel Core i7@2.8GHz and 12GB RAM. The cloth simulation are loaded as a dynamic sequence of meshes with fixed connectivity. While our code is not optimized for speed, tests ran at interactive rates, as with most other 2D surface flow simulations.

We begin with a few tests describing the benefits of some of te key components of our approach.

Homogenization. We tested our homogenization of the diffusion/permeability tensor on an inhomogeneous rectangle. As shown in Figure 4, the homogenized diffusion tensor produces diffusion results that closely resemble the original tensor field, while the direct averaging of the tensor fields in the original domain results in a completely different stain. We used a knitting pattern to compute the tensor field in the top row with D set to 1 for gap-to-gap, 100 for weft-to-weft, 500 for warp-to-warp, 10 for gap-to-weft, 50 for gap-to-warp, and 200 for weft-to-warp. The tensor field in the bottom row has alternating columns of isotropic material, one with diffusion coefficient 1 and the other 100; in this case, the effective bulk tensor is anisotropic unlike the average tensor, properly capturing the net effect of generating a faster diffusion along the vertical direction.

Inertial forces. In the rotating planar region test (Figure 5), we show that without fictitious force, the fluid motion clearly fails to account for the motion of the underlying mesh. If we only ignore the influence of rotation through the Coriolis effects, as was often done in graphics, the fluid still deviates significantly from the physical trajectory. When the bunny model undergoes rigid motion (Figure 1c), our procedure leads to realistic fluid motion, taking inertial forces as well as gravity into account. When the underlying surface is deforming, we can still capture the translation and rotation of the local frames, as shown in Figures 7 and 8. In the static flag example (Figure 6), we also tested the influence of the initial velocity (four random outgoing velocity fields to simulate a splash) has on the stain shape. The influence of the direction is as obvious in the deforming flag in Figures 7a and 7b. Figure 8b shows the trail of the fluid turning right under the Coriolis effect as compared to the straight track in Figure 8a when only the outward going translation inertial force is used (corresponding to centrifugal force from the reference frame rotating at the center of the table).

Additional tests were performed to simulate effects when fluids are constantly added to the surface, by fixing the solvent and solute densities in a certain region. We did the tests for all the above experiments, and the results for the rotating bunny with different patterns are shown in Figure 9. By using a light textile, we can also show the effects of the solution influencing the deformation by including the mass and momentum of the liquid in the update for cloth simulation. As shown in Figure 10, this simple approach



(a) (b)

Fig. 7: Simulation on a waving flag.

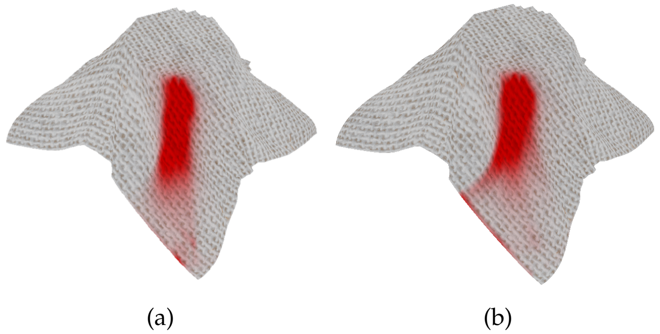
is adequate for visually plausible results, although more careful treatment of the coupling can replace it if needed.

Absorption, adsorption, evaporation, and anisotropy. In Figure 11, we can see the necessity of absorption, adsorption, and evaporation for the ring-like halo effects of stains that are common in certain materials. The halo also changes shape according to the anisotropy due to the textile patterns. Stains near a corner formed by three different textile patterns can also be handled, as shown in Figure 11d.

Effects of parameters. In the following, we assume the use of the metric system for units. We take rather large time steps (h between 0.01s and 0.025s) due to the stability from the diffusion. The gravitational acceleration is 9.8m/s^2. Note the densities that we use are not the volumetric densities (for the solvent, it is 1kg/m^3) but surface densities, and they are not considered parameters but variables. The ratio between the maximum and minimum eigenvalues of the bulk diffusion matrix in our examples can range from 1 to over 1 million. We also demonstrate the effects of the absorption speed K_p, which controls how fast the fluid is absorbed into the material, and σ_{sat}, which controls the capacity of the material to hold fluids, for example, it is small for printing paper and large for tissues. In Figure 12, we show K_p from 0.01 to 0.2, and σ_{sat} from 2 to 8. Similarly, the adsorption parameters K_d and A_d (which is computed from some other parameters, but in simulation, we manipulate A_d directly). The difference on the visual results between absorption and adsorption is that the latter only takes in the solute.

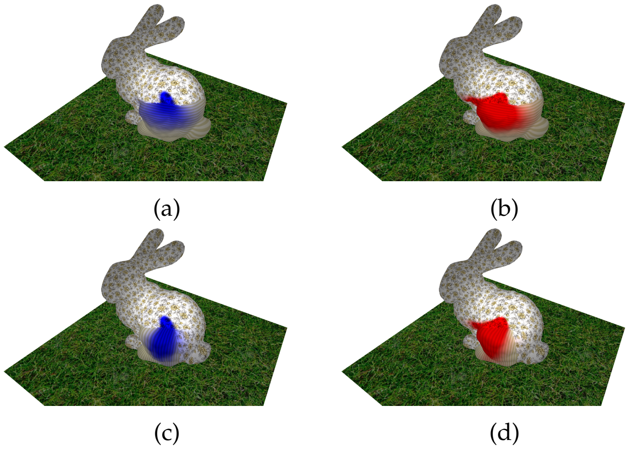
Finally, we generated a few additional examples to show the entire process of the formation of the stain in the accompanying video. Table 1 shows the breakdown of computation time.

Limitations. We did not consider the possible chemical reactions between the stain and the fabric. We did not take into account the optical properties of stain either, which may significantly influence the rendering of results. Our implementation is still a proof-of-concept, and there is much space for improvement, in particular, the inertia force computation can be parallelized in a GPU implementation. As we focus on the framework for stain evolution, in particular the homogenization procedure and the fictitious forces, we did not carefully examine the dependence of discretization of covariant derivatives on the triangulation of the surface, and alternatives to upwinding finite-volume schemes for density update. However, most of these issues are not intrinsic to our framework, and can be addressed by improving certain components in the framework.



(a) (b)

Fig. 8: Stain on a dropping tablecloth.



(a) (b)

(c) (d)

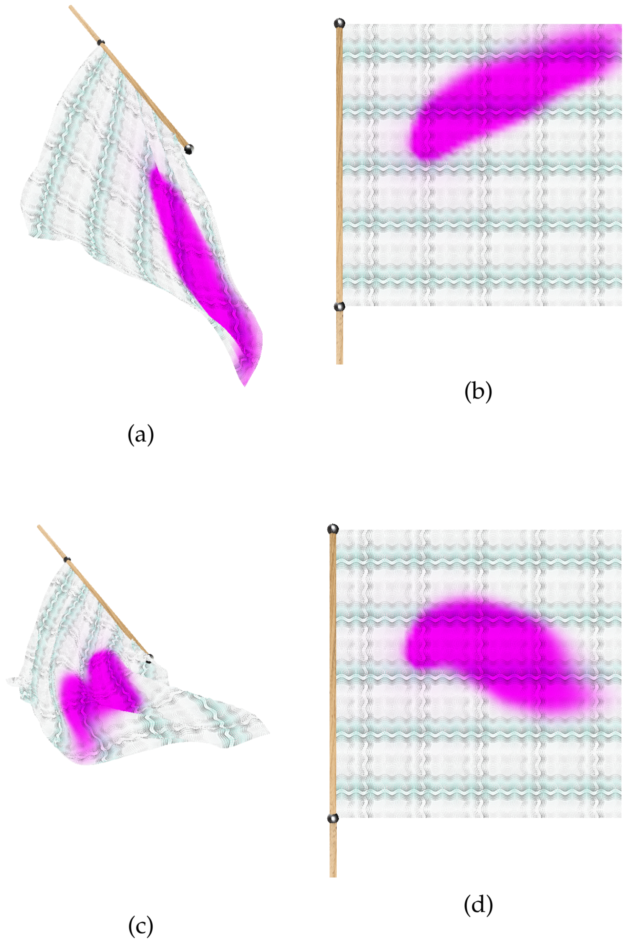
Fig. 9: Stain on a rotating bunny with different diffusion tensors (a) and (c), and when fluid is constantly added to the initial spot (b) and (d).

7 Conclusion

We have presented an efficient technique to simulate the evolution of a liquid solution staining an inelastic fabric. Instead of resorting to the computationally intensive direct simulation involving large numbers of threads or yarns, we analyze the pattern of the textile and use effective bulk diffusion tensors on the surface to accurately capture the formation and evolution of stains over highly inhomogeneous materials. Our system handles the resulting anisotropic diffusion process with a simple modification to the (Bochner) connection Laplacian. The one-way coupling from the mesh motion to the fluid is modeled by an accurate approximation

Name	#V	#F	Inert	Adv	Diff	Def
tablecloth	2.5K	4.8K	16	15	<1	NA
flag	2.5K	4.8K	15	15	<1	NA
flag(2-way coupling)	2.5K	4.8K	16	16	<1	671
Rotating cloth	10K	19.6K	94	47	<1	NA
bunny	35K	70K	359	156	55	NA
bunny(added fluid)	35K	70K	375	172	62	NA
bunny(multi tensors)	35K	70K	358	172	62	NA

TABLE 1: **Statistics.** From left to right: number of vertices, number of triangles, time for inertia force computation, time for momentum advection, time for diffusion, and time for deformation with two-way coupling. All timings are in milliseconds.



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

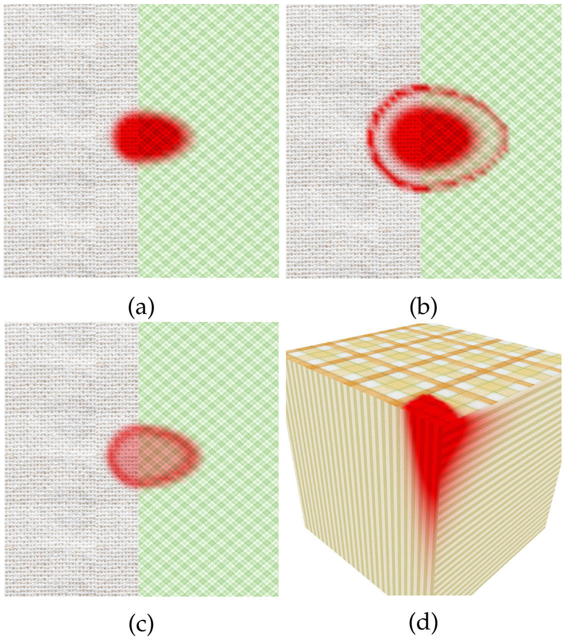
Fig. 10: Waving flag with two-way coupling between solid and fluid (a) without Coriolis force and (c) with Coriolis force. The same stains shown on the rest shape (b) and (d).

of inertial forces (including centrifugal and Coriolis effects). We also account for the absorption, adsorption, and evaporation processes, and model them as independent ODEs for each vertex. The resulting stain simulation is visually plausible and exhibits complex behaviors depending on the fabric, solution, and motion of the inelastic cloth.

In the future, we wish to explore multi-layered textile model for diffusion, the effects of possible chemical reaction, two-way coupling, wetting, possible use of dynamical texture for adding back high frequencies taken out during homogenization, and learning parameters from real stains on textile. Adding fictitious forces due to elastic stretching may also be visually interesting for very specific materials.

Acknowledgements

We thank Beibei Liu for her help on the homogenization code and Mathieu Desbrun for valuable suggestions. This project was supported in part by NSF grants IIS-0953096, CMMI-1250261, III-1302285, and CCF-1655422. The corresponding author, Shiguang Liu, was supported by the Natural Science Foundation of China under grant Nos. 61672375 and 61170118.



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

Fig. 11: Anisotropic diffusion without (a) and with (b,c) the influence of absorption, adsorption, and evaporation. (d) shows the behavior near a corner at three different tiling patterns.

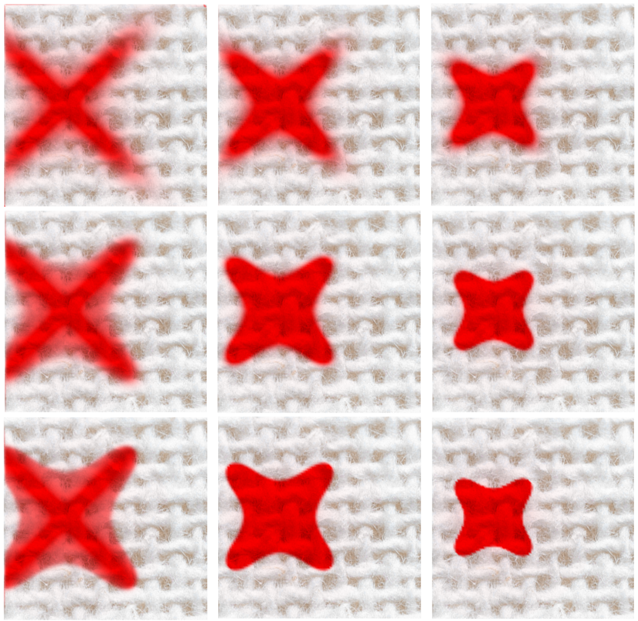


Fig. 12: Effects of different absorption speeds (increasing from top to bottom) and absorption saturation densities (increasing from left to right). Higher speed of absorption means that the fluid is absorbed fast until it is saturated, and the stain boundary is less blurry. Higher saturation means that liquid stops at a shorter distance on the surface.

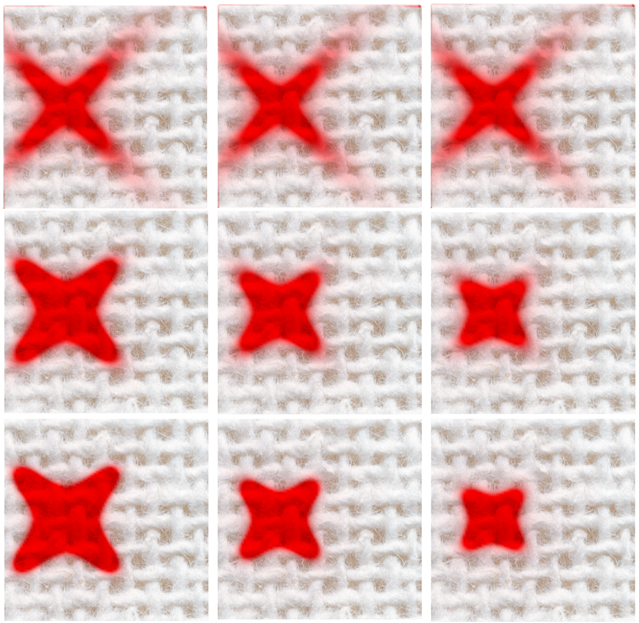


Fig. 13: Effects of different adsorption speeds (increasing from top to bottom) and adsorption saturation densities (increasing from left to right). The effects are similar to those of absorption, but the solvent is not adsorbed so it may carry the remaining little solute further even when the saturation limit is high.

References

[1] J. Stam, "Stable fluids," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 121–128.

[2] C. J. Curtis, S. E. Anderson, J. E. Seims, K. W. Fleischer, and D. H. Salesin, "Computer-generated watercolor," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 421–430.

[3] T. L. Kunii, G. V. Nosovskij, and T. Hayashi, "A diffusion model for computer animation of diffuse ink painting," in *Computer Animation'95., Proceedings*. IEEE, 1995, pp. 98–102.

[4] T. L. Kunii, G. V. Nosovskij, and V. L. Vecherinin, "Two-dimensional diffusion model for diffuse ink painting," *International Journal of Shape Modeling*, vol. 7, no. 01, pp. 45–58, 2001.

[5] N. S.-H. Chu and C.-L. Tai, "Moxi: real-time ink dispersion in absorbent paper," in *ACM Transactions on Graphics (SIGGRAPH)*, vol. 24, no. 3. ACM, 2005, pp. 504–511.

[6] T. Van Laerhoven, J. Liesenborgs, and F. Van Reeth, "Real-time watercolor painting on a distributed paper model," in *Computer Graphics International, 2004. Proceedings*. IEEE, 2004, pp. 640–643.

[7] T. Van Laerhoven and F. Van Reeth, "Real-time simulation of watery paint," *Computer Animation and Virtual Worlds*, vol. 16, no. 3-4, pp. 429–439, 2005.

[8] H. T. Wong and H. H. Ip, "Virtual brush: a model-based synthesis of chinese calligraphy," *Computers & Graphics*, vol. 24, no. 1, pp. 99–113, 2000.

[9] J. Lee, "Diffusion rendering of black ink paintings using new paper and ink models," *Computers & Graphics*, vol. 25, no. 2, pp. 295–308, 2001.

[10] D.-L. Way and Z.-C. Shih, "The synthesis of rock textures in chinese landscape painting," in *Computer Graphics Forum*, vol. 20, no. 3. Wiley Online Library, 2001, pp. 123–131.

[11] Y. J. Yu, Y. B. Lee, H. G. Cho, and D. H. Lee, "A model based technique for realistic oriental painting," in *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on*. IEEE, 2002, pp. 452–453.

[12] S.-W. Huang, D.-L. Way, and Z.-C. Shih, "Physical-based model of ink diffusion in chinese paintings," 2003.

[13] Y. Morimoto, M. Tanaka, R. Tsuruno, and K. Tomimatsu, "Visualization of dyeing based on diffusion and adsorption theories," in *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*. IEEE, 2007, pp. 57–64.

[14] S. Liu, G. Chen, P. Yang, J. Zhang, and J. Sun, "Realistic simulation of stains on cloth," *Journal of CAD&CG*, vol. 20, no. 9, pp. 1110–1116, 2008.

[15] S. Liu and D. Chen, "A computational approach to digital hand-painted printing patterns on cloth," *Multimedia Tools and Applications*, vol. 75, no. 8, pp. 4577–4591, 2016.

[16] J. Stam, "Flows on surfaces of arbitrary topology," *ACM Transactions On Graphics (TOG)*, vol. 22, no. 3, pp. 724–731, 2003.

[17] L. Shi and Y. Yu, "Inviscid and incompressible fluid simulation on triangle meshes," *Computer Animation and Virtual Worlds*, vol. 15, no. 3-4, pp. 173–181, 2004.

[18] S. Elcott, Y. Tong, E. Kanso, P. Schröder, and M. Desbrun, "Stable, circulation-preserving, simplicial fluids," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 1, p. 4, 2007.

[19] O. Azencot, S. Weißmann, M. Ovsjanikov, M. Wardetzky, and M. Ben-Chen, "Functional fluids on surfaces," in *Computer Graphics Forum*, vol. 33, no. 5. Wiley Online Library, 2014, pp. 237–246.

[20] O. Azencot, O. Vantzos, M. Wardetzky, M. Rumpf, and M. Ben-Chen, "Functional thin films on surfaces," in *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2015, pp. 137–146.

[21] B. Liu, G. Mason, J. Hodgson, Y. Tong, and M. Desbrun, "Model-reduced variational fluid simulation," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 244, 2015.

[22] S. Auer, C. B. Macdonald, M. Treib, J. Schneider, and R. Westermann, "Real-time fluid effects on surfaces using the closest point method," in *Computer Graphics Forum*, vol. 31, no. 6. Wiley Online Library, 2012, pp. 1909–1923.

[23] H. Wang, G. Miller, and G. Turk, "Solving general shallow wave equations on surfaces," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2007, pp. 229–238.

[24] H. Wang, P. J. Mucha, and G. Turk, "Water drops on surfaces," in *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 921–929.

[25] Y. Zhang, H. Wang, S. Wang, Y. Tong, and K. Zhou, "A deformable surface model for real-time water drop animation," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 8, pp. 1281–1289, 2012.

[26] Y. Jung and J. Behr, "Gpu-based real-time on-surface droplet flow in x3d," in *Proceedings of the 14th international conference on 3D web technology*. ACM, 2009, pp. 51–54.

[27] K. Djado, R. Egli, and F. Granger, "Particle-based drop animation on meshes in real time," *Computer Animation and Virtual Worlds*, vol. 23, no. 3-4, pp. 301–309, 2012.

[28] F. de Goes, B. Liu, M. Budninskiy, Y. Tong, and M. Desbrun, "Discrete 2-tensor fields on triangulations," *Computer Graphics Forum*, vol. 33, no. 5, pp. 13–24, 2014.

[29] O. Azencot, M. Ovsjanikov, F. Chazal, and M. Ben-Chen, "Discrete derivatives of vector fields on surfaces – an operator approach," *ACM Transactions on Graphics*, vol. 34, no. 3, pp. 29:1–29:13, 2015.

[30] B. Liu, Y. Tong, F. D. Goes, and M. Desbrun, "Discrete connection and covariant derivative for vector field analysis and design," *ACM Transactions on Graphics*, vol. 35, no. 3, pp. 23:1–23:17, Mar. 2016. [Online]. Available: http://doi.acm.org/10.1145/2870629

[31] U. Clarenz, U. Diewald, and M. Rumpf, "Anisotropic geometric diffusion in surface processing," in *Proceedings of the Conference on Visualization '00*, 2000, pp. 397–405.

[32] A. Singer and H.-T. Wu, "Vector diffusion maps and the connection laplacian," *Communications on Pure and Applied Mathematics*, vol. 65, no. 8, pp. 1067–1144, 2012.

[33] O. Diamanti, A. Vaxman, D. Panozzo, and O. Sorkine-Hornung, "Designing n-polyvector fields with complex polynomials," *Computer Graphics Forum*, vol. 33, no. 5, pp. 1–11, 2014.

[34] B. Ren, Y. T., L. C., X. K., and H. S. M., "Real-time high-fidelity surface flow simulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 99, pp. 1–1, 2017.

[35] R. Angst, N. Thuerey, M. Botsch, and M. Gross, "Robust and efficient wave simulations on deforming meshes," in *Computer Graphics Forum*, vol. 27, no. 7. Wiley Online Library, 2008, pp. 1895–1900.

[36] P. Neill, "Fluid flow on interacting, deformable surfaces," Ph.D. dissertation, 2008.

[37] K. Hegeman, M. Ashikhmin, H. Wang, H. Qin, X. Gu *et al.*, "Gpu-based conformal flow on surfaces," *Communications in Information & Systems*, vol. 9, no. 2, pp. 197–212, 2009.

[38] S. Jeong and C.-H. Kim, "Combustion waves on the point set surface," in *Computer Graphics Forum*, vol. 32, no. 7. Wiley Online Library, 2013, pp. 225–234.

[39] A. Bensoussan, J.-L. Lions, and G. Papanicolau, *Asymptotic analysis for periodic structures*. Elsevier, 1978.

[40] V. V. Jikov, O. Oleinik, and S. M. Kozlov, *Homogenization of differential operators and integral functionals*. Springer, 1994.

[41] H. Owhadi and L. Zhang, "Metric-based upscaling," *Communications on Pure and Applied Mathematics*, vol. 60, no. 5, pp. 675–723, 2007.

[42] L. Kharevych, P. Mullen, H. Owhadi, and M. Desbrun, "Numerical coarsening of inhomogeneous elastic materials," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 51.

[43] M. Nesme, P. G. Kry, L. Jeřábková, and F. Faure, "Preserving topology and elasticity for embedded deformable models," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 52.

[44] S.-H. Bae, H. Motomura, and Z. Morita, "Diffusion/adsorption behaviour of reactive dyes in cellulose," *Dyes and pigments*, vol. 34, no. 4, pp. 321–340, 1997.

[45] N. Umetani, D. M. Kaufman, T. Igarashi, and E. Grinspun, "Sensitive couture for interactive garment modeling and editing." *ACM Trans. Graph.*, vol. 30, no. 4, pp. 90–1, 2011.

[46] H. Fu, O. Kin-Chung Au, and C.-L. Tai, "Effective derivation of similarity transformations for implicit laplacian mesh editing," in *Computer Graphics Forum*, vol. 26, no. 1. Wiley Online Library, 2007, pp. 34–45.

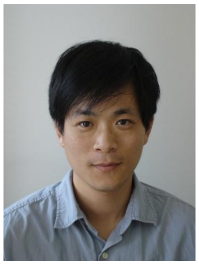
[47] W. Kabsch, "A discussion of the solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 34, no. 5, pp. 827–828, 1978.

[48] R. Lucas, "Rate of capillary ascension of liquids," *Kolloid Z*, vol. 23, no. 15, pp. 15–22, 1918.

[49] I. Langmuir, "the constitution and fundamental properties of solids and liquids. part i. solids." *Journal of the American Chemical Society*, vol. 38, no. 11, pp. 2221–2295, 1916.

Xiaojun Wang received his B.S.(2008) in Automation from Beihang University, Beijing, China. He is currently a Ph.D. candidate at Michigan State University. His research activities are focused on fluid simulation and 3d surface representations.

Shiguang Liu received his Ph.D. degree from the State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou, China. He is a currently a Professor with the School of Computer Science and Technology, Tianjin University, Tianjin, China. His research interests include computer graphics, image/video editing, visualization, and virtual reality.



Yiying Tong is an associate professor at Michigan State University. He received his Ph.D. degree from University of Southern California in 2004. His research interests include discrete geometric modeling, physically-based simulation/animation, and discrete differential geometry. He received a National Science Foundation (NSF) Career Award in 2010.