AN EXPLICIT NUMERICAL ALGORITHM FOR SURFACE RECONSTRUCTION FROM UNORGANIZED POINTS USING GAUSSIAN FILTER

HYUNDONG KIM[1], CHAEYOUNG LEE[1], JAEHYUN LEE[2], JAEYEON KIM[2], TAEYOUNG YU[2], GENE CHUNG[2], AND JUNSEOK KIM[1][†]

[1]DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 02841, REPUBLIC OF KOREA
E-mail address: cfdkim@korea.ac.kr

[2]SEOUL SCIENCE HIGH SCHOOL, SEOUL 03066, REPUBLIC OF KOREA

ABSTRACT. We present an explicit numerical algorithm for surface reconstruction from unorganized points using the Gaussian filter. We construct a surface from unorganized points and solve the modified heat equation coupled with a fidelity term which keeps the given points. We apply the operator splitting method. First, instead of solving the diffusion term, we use the Gaussian filter which has the effect of diffusion. Next, we solve the fidelity term by using the fully implicit scheme. To investigate the proposed algorithm, we perform computational experiments and observe good results.

1. INTRODUCTION

An important work in the surface reconstruction process is to preserve the original image and achieve the suitable smoothing effect. There are a number of papers that have performed the surface smoothing, the image processing, the surface or volume reconstruction by applying various techniques such as Gaussian smoothing [1], spherical diffusion [2], anisotropic diffusion [3], non-iterative feature preserving filtering [4], minimizing total variation [5, 6], level-set method [7], embedding narrow band [8, 9], and phase-field method [10].

The following model is used for removing noise from given images, which based on minimizing total variation of image [11]:

$$\frac{\partial \phi}{\partial t} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \lambda(f(\mathbf{x}) - \phi), \quad \mathbf{x} \in \Omega, \ t > 0, \tag{1.1}$$

where $\phi(\mathbf{x}, t)$ is an image to be restored in a domain Ω, $f(\mathbf{x})$ is the original image, and λ is a nonzero Lagrange multiplier.

In Eq. (1.1), since the term $\nabla \cdot (\nabla \phi / |\nabla \phi|)$ describes the effect of diffusion, we equivalently consider the following evolutionary governing equation:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \Delta \phi(\mathbf{x}, t) + \lambda(f(\mathbf{x}) - \phi(\mathbf{x}, t)), \quad \mathbf{x} \in \Omega, \tag{1.2}$$

where $\phi(\mathbf{x}, t)$ is the phase-field, $f(\mathbf{x})$ is the original data set, and λ is the fidelity parameter. We define $\phi = 1$ on the points in the domain Ω and $\phi = -1$ otherwise. Assuming that the reconstructed surface is sufficiently away from the boundary, the Dirichlet boundary condition, $\phi = -1$, can be used [12].

The main purpose of this article is to present an explicit numerical algorithm for surface reconstruction from unorganized points by using the Gaussian filter.

This article is organized the following manner. In Section 2, we describe a mathematical configuration and numerical solution algorithm using the Gaussian filter. To investigate the proposed algorithm, we perform computational experiments in Section 3. Conclusions are derived in Section 4.

2. MATHEMATICAL CONFIGURATION AND NUMERICAL SOLUTION ALGORITHM

In this section, we present an explicit numerical algorithm for surface reconstruction from unorganized points using the Gaussian filter. Let the computational domain Ω be $(a, b) \times (c, d) \times (e, f)$, the number of grid points N_x, N_y, and N_z be positive integers, and the uniform grid size h be $(b - a)/(N_x - 1) = (d - c)/(N_y - 1) = (f - d)/(N_z - 1)$. We define a discrete domain $\Omega_h = \{\mathbf{x} = (x_i, y_j, z_k) : x_i = a + (i - 0.5)h, y_j = c + (j - 0.5)h, z_k = e + (k - 0.5)h, 1 \leq i \leq N_x, 1 \leq j \leq N_y, 1 \leq k \leq N_z\}$. The notation $\phi(x_i, y_j, z_k, n\Delta t)$ is simply denoted by ϕ_{ijk}^n, where Δt is the temporal step size. By using the operator splitting method, we split the diffusion term and fidelity term of Eq. (1.2) into the following equations:

$$\frac{\partial \phi}{\partial t} = \Delta \phi, \tag{2.1}$$

$$\frac{\partial \phi}{\partial t} = \lambda(f(\mathbf{x}) - \phi). \tag{2.2}$$

When solving Eq. (1.2), we can replace the diffusion Eq. (2.1) using a Gaussian function [1]. For the sake of clarity, we first explain the function in two-dimensional space:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}},$$

where σ is the standard deviation of distribution. Figure 1 shows a graphical representation of two-dimensional Gaussian function with $\sigma = 1.0$, which resembles a bell shape. As the
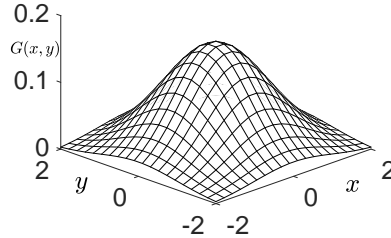

FIGURE 1. Two-dimensional Gaussian function with $\sigma = 1.0$.

first step to find the numerical solution, instead of solving the diffusion Eq. (2.1), we use the numerical convolution [13] $G * \phi_{ij}^0$ with an initial condition ϕ_{ij}^0.

Now, we consider $(m \times m)$ smoothing kernel. Here, m is an odd integer greater than 2. We define $W_{pq} = G((p - (m + 1)/2)h, (q - (m + 1)/2)h)$ for $1 \leq p, q \leq m$. In general, the value of sum of weighting coefficients holds $\sum_{p=1}^{m} \sum_{q=1}^{m} W_{pq} \neq 1$. Therefore, we normalize the weighting coefficients to make the value of sum equal to be 1, i.e., $w_{pq} = W_{pq} / \sum_{i=1}^{m} \sum_{j=1}^{m} W_{ij}$. Figure 2 shows that the shape of w_{pq} is smoother as the value of σ increases when $m = 3$.
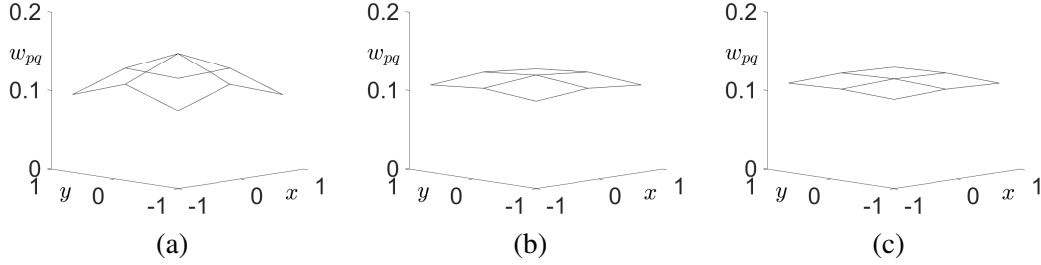

FIGURE 2. When $m = 3$, (a), (b), and (c) are the graphical representations of Gaussian filter with $\sigma = 0.5$, 1.0, and 1.5, respectively.

The solution of Eq. (2.1) obtained by using the Gaussian filter is as follows:

$$\phi_{ij}^* = \sum_{p=1}^{m} \sum_{q=1}^{m} w_{pq} \phi_{i-\frac{m+1}{2}+p, j-\frac{m+1}{2}+q}^n, \quad 1 \leq i \leq N_x,\ 1 \leq j \leq N_y, \tag{2.3}$$

which has the effect of diffusion.

When $m = 3$, Fig. 3 illustrates the stencil of $\phi_{i-(m+1)/2+p, j-(m+1)/2+q}$ and normalized weighting coefficient w_{pq} for $1 \leq p, q \leq m$. Similarly to the two-dimensional Gaussian function, the three-dimensional Gaussian function is expressed as

$$G(x, y, z) = \frac{1}{(\sqrt{2\pi}\sigma)^3} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}}.$$

For $1 \leq i \leq N_x,\ 1 \leq j \leq N_y,\ 1 \leq k \leq N_z$, we can extend the two-dimensional numerical solution ϕ_{ij}^* in Eq. (2.3) to the following three-dimensional numerical solution ϕ_{ijk}^*:

$$\phi_{ijk}^* = \sum_{p=1}^{m} \sum_{q=1}^{m} \sum_{r=1}^{m} w_{pqr} \phi_{i-\frac{m+1}{2}+p, j-\frac{m+1}{2}+q, k-\frac{m+1}{2}+r}^n. \tag{2.4}$$

Figure 4 illustrates stencil of $\phi_{i-2+p, j-2+q, k-2+r}$ and normalized weighting coefficient w_{pqr}.
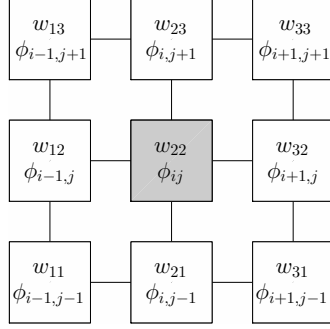
FIGURE 3. Schematic illustration of $\phi_{i-2+p,j-2+q}$ and normalized weighting coefficient w_{pq} for $1 \leq p, q \leq 3$.
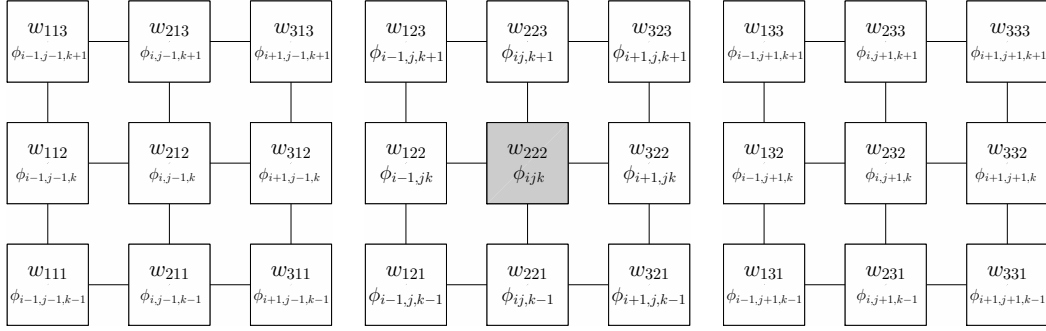

FIGURE 4. Schematic illustration of $\phi_{i-2+p,j-2+q,k-2+r}$ and normalized weighting coefficient w_{pqr} for $1 \leq p, q, r \leq 3$.

As the second step, we solve Eq. (2.2) by using the fully implicit scheme:

$$\frac{\phi_{ijk}^{n+1} - \phi_{ijk}^{*}}{\Delta t} = \lambda(f - \phi_{ijk}^{n+1}). \tag{2.5}$$

To simplify the representation, we take $\theta = 1/(1 + \Delta t \lambda)$. Then Eq. (2.5) is given as

$$\phi_{ijk}^{n+1} = \theta \phi_{ijk}^{*} + (1 - \theta)f, \quad 0 < \theta < 1$$

Since f is the initial data ϕ_{ijk}^{0}, we can straightforwardly find the numerical solution as follows:

$$\phi_{ijk}^{n+1} = \theta \phi_{ijk}^{*} + (1 - \theta)\phi_{ijk}^{0}. \tag{2.6}$$

As a consequence, an explicit numerical algorithm consists of Eqs. (2.4) and (2.6). Therefore, the implicit numerical solver such as multigrid solver is not needed to find the numerical solution. Also, the execution of the algorithm is fast.

3. COMPUTATIONAL EXPERIMENTS

In this section, we perform numerical tests using the proposed numerical method. Figure 5 shows surface reconstruction of the Stanford bunny [14] on the domain $\Omega = (0,1) \times (0,1) \times (0,1)$. We take the following computational parameters: $N_x = N_y = N_z = 60$, $h = 1/60$, $\Delta t = h^2$, $\sigma = h$, and $\lambda = 700$. From left to right, the given unorganized points set, initial reconstruction, and the numerical results after 5 iterations.

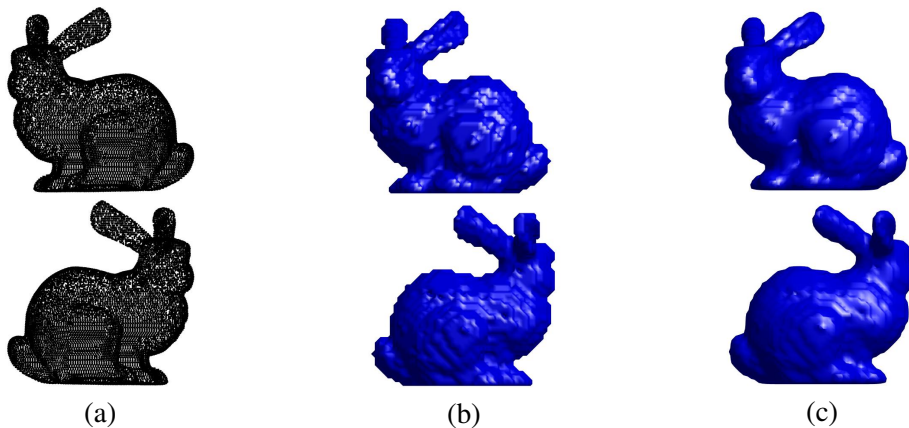(a)   (b)   (c)

FIGURE 5. Surface reconstruction of bunny: (a) the given points set, (b) initial reconstruction, and (c) the numerical results after 5 iterations, respectively.

3.1. **Effect of smoothing kernel.** On the three-dimensional space $\Omega = (0,1) \times (0,1.5) \times (0,1)$, we simulate the effect of smoothing kernel. The initial data is set to the Stanford dragon model [14]. We take the following computational parameters: $N_x = 120$, $N_y = 180$, $N_z = 120$, $h = 1/120$, $\Delta t = 0.05h$, $\sigma = 2h/\sqrt{3}$, and $\lambda = 3000$. Figure 6(a) shows the initial condition. At the final time $T = 5\Delta t$, the results of $3 \times 3 \times 3$ and $5 \times 5 \times 5$ smoothing kernel are shown in Fig. 6(b) and (c), respectively. We can see that $5 \times 5 \times 5$ smoothing kernel are smoother than $3 \times 3 \times 3$ smoothing kernel. It means that the initial data is lost. Therefore, all computational experiments shall use the Gaussian filter computed by $3 \times 3 \times 3$ smoothing kernel, from now on.

3.2. **Effect of standard deviation σ.** In this section, we confirm the effects of the standard deviation σ on the three-variable Gaussian filter dynamics. The initial data is taken from the Stanford bunny model. We can observe the shape of the model at time $T = 5\Delta t$ with respect to different standard deviations. In Fig. 7, the parameters are given as $N_x = 120$, $N_y = 180$, $N_z = 120$, $h = 0.0147$, $\Delta t = 0.05h$, $\lambda = 0$, and $\Omega = (0,1) \times (0,1.5) \times (0,1)$. As shown in Fig. 7, when the standard deviation of the given Gaussian filter is small, it is seen that the resulting surface closely resembles the initial model. On the other hand, when the standard deviation of the filter becomes larger, the resulting surface is smoother. Thus, the standard deviation of the filter must be set to an appropriate value for each set of initial data.
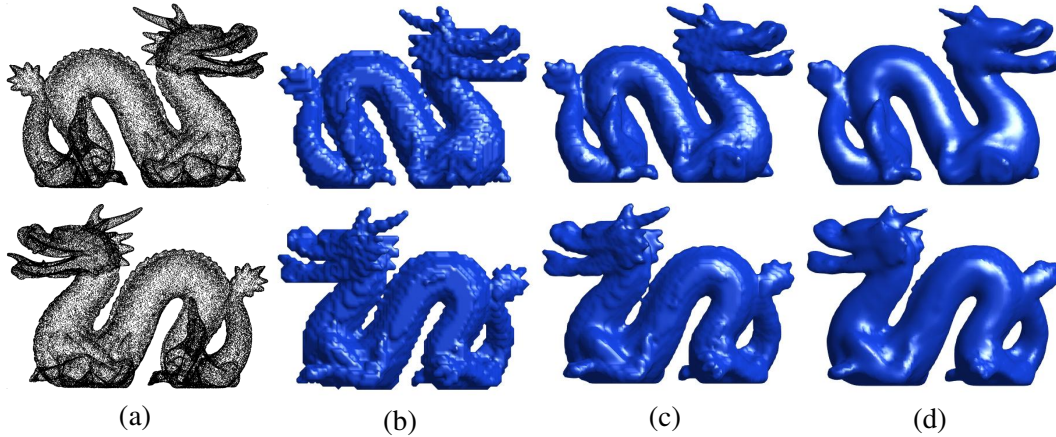
FIGURE 6. Right and left views of Stanford dragon: (a) initial point set, (b) initial condition (c) $3 \times 3 \times 3$, and (d) $5 \times 5 \times 5$ smoothing kernel, respectively.
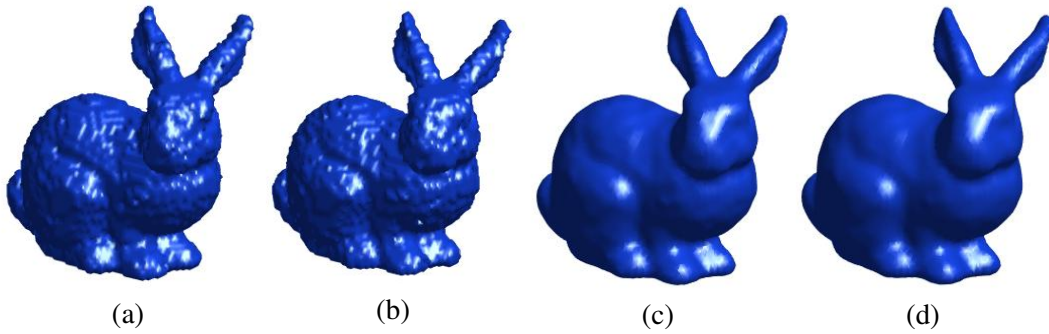

FIGURE 7. The graphical views of Stanford bunny: (a) shows the initial condition and (b), (c), and (d) are the numerical results with $\sigma = 0.01h, h$, and $100h$, respectively.

3.3. **Effect of parameter** λ. From now on, we investigate the effect of parameter λ on the three-variable Gaussian filter dynamics. The initial condition is used as Stanford bunny and dragon model. We can observe the shape of model at the total time $T = 5\Delta t$ with respect to different values of λ. The common performance parameters are taken as following: $N_x = 120$, $N_y = 180$, $N_z = 120$, $\Delta t = 0.05h$, $\sigma = 2h/\sqrt{3}$, and $\Omega = (0, 1) \times (0, 1.5) \times (0, 1)$, where $h = 0.01$ in Fig. 8 and $h = 0.0147$ in Fig. 9. As shown in Fig. 8, 9, when the value of parameter λ is small, we can see that the result is oversmoothing by the Gaussian filter with a sufficiently large standard deviation σ. Meanwhile, if the value of parameter λ is large, then the results are rougher, which seems like an initial condition. Therefore, λ must be set to a suitable value for each given initial data.
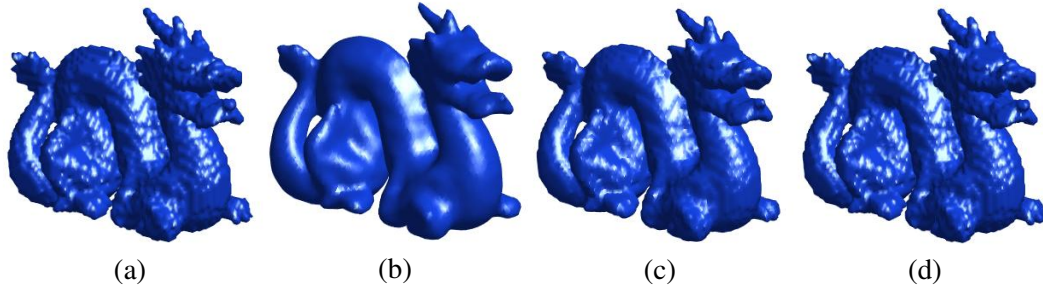
FIGURE 8. The graphical views of Stanford dragon: (a) shows the initial condition and (b), (c), and (d) are the numerical results with $\lambda = 1,\ 1000$, and 10000, respectively.
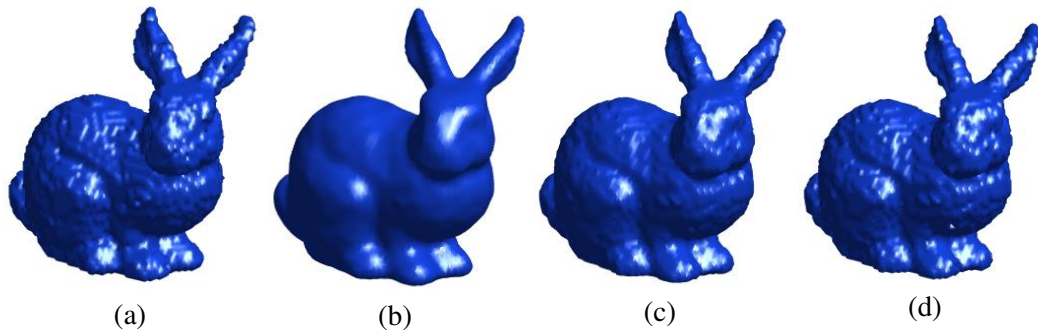

FIGURE 9. The graphical views of Stanford bunny: (a) shows the initial condition and (b), (c), and (d) are the numerical results with $\lambda = 1,\ 1000$, and 10000, respectively.

4. CONCLUSION

In this article, we presented an explicit numerical algorithm for surface reconstruction from unorganized points by using the Gaussian filter. We have investigated the diffusion effect of the Gaussian filter through the computational experiments and observed the numerical results. The proposed algorithm can be utilized in many aspects of industry, such as the three-dimensional structure printing from scattered scanned data.

ACKNOWLEDGMENT

References

[1] G. Taubin, *Curve and surface smoothing without shrinkage*, In Proceedings of IEEE international conference on Computer Vision, (1995), 852–857.

[2] T. Bulow, *Spherical diffusion for 3D surface smoothing*, IEEE Trans. Pattern Anal. Mach. Intell. **26**, (2004), 1650–1654.

[3] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, *Geometric surface smoothing via anisotropic diffusion of normals*, In Proceedings of the conference on Visualization, (2002), 125–132.

[4] T. R. Jones, F. Durand, and M. Desbrun, *Non-iterative, feature-preserving mesh smoothing*, ACM Trans. Graphics **22** (2003), 943–949.

[5] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D. **60** (1992), 259–268.

[6] L. Vese, and S. Osher, *Modeling textures with total variation minimization and oscillating patterns in image processing*, J. Sci. Comput. **19** (2003), 553–572.

[7] H. K. Zhao, S. Osher, and R. Fedkiw, *Fast surface reconstruction using the level set method*, In Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision, (2001), 194–201.

[8] Y. Li, D. Lee, C. Lee, J. Lee, S. Lee, J. Kim, S. Ahn, and J. Kim, *Surface embedding narrow volume reconstruction from unorganized points*, Comput. Vis. Image Underst. **121** (2014), 100–107.

[9] Y. Li and J. Kim, *Fast and efficient narrow volume reconstruction from scattered data*, Pattern Recognit. **48** (2015), 4057–4069.

[10] D. Jeong, Y. Li, H. Lee, S. Lee, J. Yang, S. Park, H. Kim, Y. Choi, and J. Kim, *Efficient 3D volume reconstruction from a point cloud using a phase-field method*, Math. Probl. Eng. (2018).

[11] L. Rudin, and S. Osher, *Total variation based image restoration with free local constraints*, In Proceedings of 1st International Conference on Image Processing, **1** (1994), 31–35.

[12] H. Li, Y. Li, R. Yu, J. Sun, and J. Kim, *Surface reconstruction from unorganized points with l_0 gradient minimization*, Comput. Vis. Image Underst. **169** (2018) 108–118.

[13] Y. Li and J. Kim, *A fast and accurate numerical method for medical image segmentation*, J. KSIAM. **14** (2010), 201–210.

[14] Stanford university computer graphics laboratory, http://lightfield. stanford.edu/acq.html.