Meshfree Particle Method

Huafeng Liu and Pengcheng Shi
Medical Image Computing Group, Department of Electrical and Electronic Engineering
Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

Abstract

Many of the computer vision algorithms have been posed in various forms of differential equations, derived from minimization of specific energy functionals, and the finite element representation and computation have become the de facto numerical strategies for solving these problems. However, for cases where domain mappings between numerical iterations or image frames involve large geometrical shape changes, such as deformable models for object segmentation and nonrigid motion tracking, these strategies may exhibit considerable loss of accuracy when the mesh elements become extremely skewed or compressed. We present a new computational paradigm, the meshfree particle method, where the object representation and the numerical calculation are purely based on the nodal points and do not require the meshing of the analysis domain. This meshfree strategy can naturally handle large deformation and domain discontinuity issues and achieve desired numerical accuracy through adaptive node and polynomial shape function refinement. We discuss in detail the element-free Galerkin method, including the shape function construction using the moving least square approximation and the Galerkin weak form formulation, and we demonstrate its applications to deformable model based segmentation and mechanically motivated left ventricular motion analysis.

1 Introduction

1.1 Finite Element Methods

Many of the computer vision algorithms are posed as various energy minimization problems, and become partial differential equations (PDEs) subject to image data constraints through natural and essential boundary conditions. Because the analysis domains in these problems are often spatially irregular and sampled at discrete points, the finite element methods (FEMs) have become the de facto computational strategy to provide numerical solutions through the discretization of the analysis domains into meshes with predefined connectivity between nodal points. The main computational power of these approaches results from the fundamental idea of replacing a continuous function $f(\mathbf{x})$ defined over the entire analysis domain by piecewise polynomial approximations over a set of finite number of geometrically simple sub-domains such as triangles. With suitable regularization constraints (i.e. smoothness, continuum mechanical models, etc.) and proper formulation principles (i.e. the virtual work), governing differential equations can be approximated by a set of algebraic equations for all the elements, and image-derived boundary conditions are enforced to provide the solutions. An incomplete survey of more recent works shows a wide range of FEM-based strategies in computer vision, including object segmentation [5, 23], shape representation and characterization [11, 20], correspondence and motion estimation [18, 21], image registration [9], and image guided surgery [4, 7].

Although the idea of domain division has been proven really ingenious and well-suited for many vision problems, proper mesh generation from the sampling nodes can sometimes be difficult and time consuming, especially for constrained meshing of material and/or kinematics discontinuities and for three-dimensional (3D) cases. Furthermore, in dynamic formulation which is common for computer vision problems such as segmentation and motion analysis, the numerical accuracy and efficiency of FEM decreases drastically whenever the mesh becomes extremely skewed or compressed, i.e. there is large geometrical shape changes of the objects between numerical iterations or image frames. In these situations, adaptive remeshing and/or node refinement must be performed throughout the evolution in order to prevent the severe distortion of elements [11], to allow mesh lines to remain coincident with any discontinuities [1], and to maintain reasonable numerical accuracy. However, remeshing requires the projection of field variables between meshes in successive stages of the problem, which leads to huge logistical problems even for medium size problems. Further, for large three-dimensional problems which are becoming more common, the computational cost of remeshing at each step often becomes prohibitively expensive.

1.2 Meshfree Particle Methods

The emerging meshfree particle methods (MPMs) offer computationally efficacious alternatives to circumvent the aforementioned problems encountered by the FEMs. Orig-

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

inated as special numerical methods for computational mechanics analysis, with variations such as the smooth particle hydrodynamics methods, the diffuse element methods, the element free Galerkin methods, the reproducing kernel particle methods, and the partitions of unity methods, MPMs have recently shown their prowess in solving general PDEs over complex domain [16, 19].

The main motive to adopt MPMs is to eliminate at least part of the mesh structure by constructing the approximation of the field function $f(\mathbf{x})$ entirely in terms of the nodal points, whereas no specific pairwise characterization of the nodal interrelationship is defined or needed [1]. MPMs represent the analysis domain with only a set of nodal points without mesh constraints, and establish a system of algebraic equations for the whole problem domain based on the particle-derived interpolating shape functions. Procedure-wise, MPMs are actually very similar to FEMs, with the fundamental differences in the elimination of mesh and the construction of the shape functions from nodes only. In FEMs, the shape functions are constructed using the mesh structure, and they are the same for all the elements of the same type in the natural coordinates systems. These shape functions are usually predetermined for different types of elements *before* the analysis starts. In MPMs, however, the shape functions are usually constructed for a particular point of interest (POI), and they will change as the location of the POI changes. Hence, the construction of the meshfree shape function is performed *during* the analysis.

MPMs are better suited to cope with geometric changes of the domain of interest, e.g. free surfaces and large deformations, than FEMs [1, 15, 16]. From numerical efficiency and accuracy point of view, the principal attraction of MPMs is the possibility of simplifying spatial adaptivity (node addition or elimination) and shape function polynomial order adaptivity (approximation/interpolation types), and handling moving boundaries and discontinuities. Adaptive meshing procedure, which may be needed for a large variety of problems including deformable models, image registration, and motion detection, can be effectively treated in a much simpler manner as a node refinement problem. In areas where more refinement is needed, nodes can be added easily to achieve desired numerical accuracy. Since there is no need to generate the mesh representation and the connectivity between nodes is generated as part of the computation and can be changed over time, MPMs facilitate the handling of very large deformations and material/kinematics discontinuity. An illustration of the accuracy advantage of MPM over FEM (without adaptive remeshing) in a deformation simulation is shown in Figure 1. Further, several recent meshfree efforts incorporate the multi-scale concept for problems involving widely varying scales through wavelet based basis function enhancement [24], which meshes well with the scale-space tradition in computer vision research.

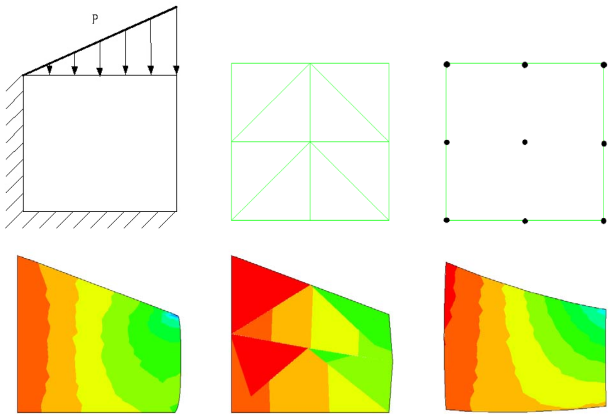

Figure 1: Comparison of FEM and MPM computed vertical strains for a simple elastic deformation simulation: experiment setup and theoretical strain distribution (left), FEM mesh and strain map (middle), and MPM representation and strain map (right). All the strain maps here use the same color scale.

1.3 Contributions

In this paper, we present a meshfree particle *computational* paradigm for certain computer vision problems involving object deformation. We want to emphasize that we are *not* presenting a new deformable model per se, but rather novel ways for object representation and numerical calculation. The framework has its roots in the popular finite element methods, but is purely based on the nodal points and does not require the meshing of the analysis domain. This meshfree strategy can naturally handle large deformation and domain discontinuity issues and achieve desired numerical accuracy through adaptive node and polynomial shape function refinement. We discuss in detail the element-free Galerkin method, one of the more robust and better developed meshfree methods, through the construction of the shape functions using the moving least square approximation, the Galerkin weak form formulation, and the imposition of boundary conditions using the penalty method. We demonstrate the computational power of this framework with applications in object segmentation with one- and two-dimensional deformable models (contours and annuli), and multi-frame non-rigid motion analysis of the left ventricle using mechanical model constrained optimal filtering.

2 EFGM Framework for Vision Computation

Various meshfree particle methods have emerged in the last a few years, as summarized in [1, 15], and the most basic common feature is that a predefined mesh is not necessary, at least not for field function interpolation. We focus on the element-free Galerkin method (EFGM) in this paper because it is relatively well developed, robust, and has straightforward and obvious links to the finite element methods [2]. In many applications, EFGM has shown superior rate of convergence and high efficiency in modelling moving interfaces [16].

As a meshfree particle method, EFGM requires only the definition of a set of nodes distributed over the entire analysis region, and the definition of the boundary conditions (Figure 1). The all important shape functions are constructed using the moving least square (MLS) approximations, generated at each nodal point and its associated local influence domain. However, EFGM does require the *background cells* for the integration of system matrices derived from the Galerkin weak form over the problem domain, which in practice can be very easily created. Further, EFGM does not allow direct imposition of essential boundary conditions, which are instead enforced via Lagrange multipliers [2] or penalty methods [10]. In the derivation of the EFGM procedures for vision computation, we assume that we are dealing with domain evolution problems (i.e. segmentation) or domain mapping problems (i.e. non-rigid motion) using elastic continuum mechanical models. However, this assumption is only used as an example to develop the weak form and the differential governing equations. In reality, EFGM can be used as the computational tools for any of the energy or variational formulations in computer vision.

2.1 Galerkin Weak Form Formulation

A Galerkin statement incorporates differential equations in their weak form so that they are satisfied over a domain in an integral or average sense rather than at every point [6]. Under the Hamilton's principle of energy and work [16], assuming commonly used linear elastic materials with the material constitutive equation $\sigma = \mathbf{c}\varepsilon$ where $\mathbf{c}$ is the material matrix, and the strain-displacement equation $\varepsilon = \mathbf{Lu}$ where $\mathbf{L}$ is a differential operator matrix dependent on the strain types (infinitesimal or finite), we arrive at:

$$\int_\Omega \delta(\mathbf{Lu})^T \mathbf{c}(\mathbf{LU})\, d\Omega - \int_\Omega \delta\mathbf{u}^T \mathbf{b}\, d\Omega - \int_{\Gamma_t} \delta\mathbf{u}^T \mathbf{t}\, d\Gamma + \int_\Omega \rho\delta\mathbf{u}^T \ddot{\mathbf{u}}\, d\Omega = 0 \qquad (1)$$

where Ω is the volume of the domain, $\mathbf{u}$ the displacement field variable, ρ the material density, σ the stress tensor, ε the strain tensor, $\mathbf{b}$ the body force, Γ_t the surface where external forces are prescribed, and $\mathbf{t}$ the traction force. For static problems, i.e. only essential boundary conditions are available as in most vision problems where the *process* of the evolution or mapping is not of essential concern, the last term on the left hand of Equation (1) is dropped.

2.2 Meshfree Particle Representation

The analysis domain is modelled and represented using sets of sampling nodes scattered in the analysis domain and its boundaries. The density of the nodes are controlled by the needs on numerical accuracy and by the available computational resources, and they can be non-uniformly sampled with denser distributions in areas with larger shape variation and larger field variable gradient. Further, because of the intrinsic spatial adaptivity of the EFGM, the node density can be automatically and adaptively controlled, and the initial nodal distribution quality needs not to be of major concerns [16]. See Figures 1, 4, and 5 for examples.

2.3 Shape Function Construction

The purpose of the shape functions is to approximate the field functions using their values at sampling nodes in the analysis domain. In FEMs, the shape functions are constructed using the mesh of elements. For meshfree particle methods, the challenges are to construct shape functions using only the sampling nodes in the domain without any predefined node connectivity, and to satisfy the basic requirements of compatibility throughout analysis domain, arbitrary node distribution for easy node adaptivity, stability and consistency to ensure numerical convergence, compact support for computational accuracy and efficiency, and Kronecker delta function property for imposing of essential boundary condition [16]. The moving least squares (MLS) approximation [14] has been used for constructing the shape functions in EFGM, and it has provided continuous and smooth field variable approximation throughout the problem domain with desired order of consistency.

2.3.1 Moving Least Squares Approximation

MLS approximation is a method of finite series representation of function through local regression. Let $u(\mathbf{x})$ be the field variable function defined in domain Ω, and $u^h(\mathbf{x})$ the approximation of $u(\mathbf{x})$ at point $\mathbf{x}$. In MLS approximation,

$$u^h(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x}) a_j(\mathbf{x}) \equiv \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \qquad (2)$$

where $\mathbf{p}(\mathbf{x})$ is the polynomial basis functions, m the number of terms in $\mathbf{p}(\mathbf{x})$, and $\mathbf{a}(\mathbf{x})$ the unknown location-dependent coefficients which can be obtained at any point $\mathbf{x}$ by minimizing a weighted, discrete L_2 norm:

$$J = \sum_{I}^{n} w(\mathbf{x} - \mathbf{x}_I)[\mathbf{p}^T(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) - u_I]^2 \qquad (3)$$

with n the number of points $\mathbf{x}_I$ in the neighborhood of $\mathbf{x}$ (the *influence domain*) (Figure 2) for which the weight function $w(\mathbf{x} - \mathbf{x}_I) \neq 0$, and the nodal value $u_I = u(\mathbf{x}_I)$.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

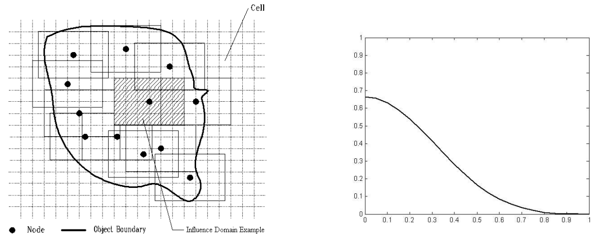

Figure 2: Influence domains and cubic spline weighting function

The stationarity of J in Equation (3) with respect to $\mathbf{a}(\mathbf{x})$, $\frac{\partial J}{\partial \mathbf{a}} = 0$, leads to:

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{U}_s \tag{4}$$

where

$$\begin{aligned} \mathbf{A}(\mathbf{x}) &= \sum_I^n w(\mathbf{x} - \mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)\mathbf{p}^T(\mathbf{x}_I) && (5) \\ \mathbf{B}(\mathbf{x}) &= [\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_n] && (6) \end{aligned}$$

with $\mathbf{B}_I = w(\mathbf{x} - \mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)$ and $\mathbf{U}_s^T = \{u_1, u_2, ..., u_n\}$. Note that we require $n >> m$ which prevents the singularity of the matrix $\mathbf{A}$ and ensures the existence of $\mathbf{A}^{-1}$, and this condition can be guaranteed by the adaptively controlled influence domain size. Substituting the results into Equation (2), and we have

$$\begin{aligned} u^h(\mathbf{x}) &= \sum_I^n \sum_j^m p_j(\mathbf{x})(\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}))_{jI} u_I && (7) \\ &\equiv \sum_I^n \phi_I(\mathbf{x}) u_I \equiv \mathbf{\Phi}(\mathbf{x})\mathbf{U}_s && (8) \end{aligned}$$

where the MLS-derived shape function $\phi_I(\mathbf{x})$ is

$$\phi_I(\mathbf{x}) = \sum_j^m p_j(\mathbf{x})(\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}))_{jI} = \mathbf{p}^T\mathbf{A}^{-1}\mathbf{B}_I \tag{9}$$

and $\mathbf{\Phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ..., \phi_n(\mathbf{x})] = \mathbf{p}^T\mathbf{A}^{-1}\mathbf{B}$. Further, the derivatives of the shape functions that are necessary to compute the gradients of the approximations are given by:

$$\mathbf{\Phi}'(\mathbf{x}) = (\mathbf{p}^T)'\mathbf{A}^{-1}\mathbf{B} + \mathbf{p}^T((\mathbf{A}^{-1})'\mathbf{B} + \mathbf{A}^{-1}\mathbf{B}') \tag{10}$$

It can be shown that if the weight functions $w(\mathbf{x} - \mathbf{x}_I)$ is continuous together with its first k derivatives, then the shape function is also continuous together with its first k derivatives [14]. More efficient schemes for derivatives computation can be found in [3].

2.3.2 Basis and Weighting Functions

The use of the basis functions $\mathbf{p}(\mathbf{x})$, including special terms such as singularity functions, ensures the consistency of the approximation and the accuracy of the results. Typically, the basis functions consist of monomials of the lowest orders to ensure minimum completeness:

$$\begin{aligned} Linear: \quad \mathbf{p}^T_{(m=3)} &= \{1, x, y\} && (11) \\ Quadratic: \quad \mathbf{p}^T_{(m=6)} &= \{1, x, y, x^2, xy, y^2\} && (12) \end{aligned}$$

Singular enhancement functions can be included in the basis to achieve better efficiency or to produce solutions of special characteristics such as singularity because function included in the basis can always be reproduced exactly by an MLS approximation [14].

The weight functions $w(\mathbf{x} - \mathbf{x}_I)$ play important roles in constructing the MLS shape functions. They should be positive to guarantee a unique solution for $\mathbf{a}(\mathbf{x})$; they should decrease in magnitude as the distance $d_I = |\mathbf{x} - \mathbf{x}_I|$ increases to enforce proper local neighbor influence; and they should ensure that nodes leave or enter the local influence domains in a gradual (smooth) manner when $\mathbf{x}$ moves, such that the MLS shape functions satisfying the compatibility condition of the Hamilton's principle.

Theoretically, the weighting functions can be any functions as long they are positive and continuous together with its derivatives up to the desired degree, and they satisfies the positivity ($w(\mathbf{x} - \mathbf{x}_I) > 0$ inside influence domain $d\Omega$), compactness ($w(\mathbf{x} - \mathbf{x}_I) = 0$ outside $d\Omega$), and unity ($\sum_I w(\mathbf{x} - \mathbf{x}_I) = 1$) conditions. In our implementation, we choose to use the cubic spline function (Figure 2):

$$w(r) = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3 & \text{for } r \leq \frac{1}{2} \\ \frac{4}{3} - 4r + 4r^2 - \frac{4}{3}r^3 & \text{for } \frac{1}{2} < r \leq 1 \\ 0 & \text{for } r > 1 \end{cases} \tag{13}$$

in which $r = d_I/d_{mI}$ is a normalized radius from the I^{th} node to the POI, and the support size d_{mI} is set by

$$d_{mI} = d_{max}c_I \tag{14}$$

where d_{max} is a scaling parameter, and the distance c_I is determined by searching for enough neighbor nodes for the matrix $\mathbf{A}$ in Equation (5) to be regular, i.e. invertible at every point in the domain. This notion actually relates to the scale-space concept in vision, and it facilitates the spatial adaptivity of the node distribution as well. For two-dimensional case, tensor product concepts are employed to determine the influence domains and the construction of the weighting functions (Figure 2) [8].

It should be noted that the MLS shape functions do not satisfy the Kronecker delta criterion and result in $u^h(\mathbf{x}_I) \neq u_I$. Therefore, they are not interpolants but rather approximants of the field function. The approximation of the function at the I^{th} node $u^h(\mathbf{x}_I)$ depends not only on the nodal parameter u_I but also on the nodal parameters of all the nodes within the influence domain of node I. This property makes the imposition of essential boundary conditions more complicated than that in the FEM. Modified *interpolating* MLS approximations have been proposed to eliminate this issue by employing singular weight functions [12, 14].

2.3.3 Remarks

As mentioned earlier, an attractive property of MLS approximations is that their continuity is related to the continuity of the weighting functions. Therefore, a low order polynomial basis $\mathbf{p}(\mathbf{x})$ such as the linear one can be used to generate highly continuous approximations by choosing appropriate weight functions. Thus, unlike FEM, there is no need of post-processing to generate smooth stress and strain fields.

The standard least-squares interpolant/approximant is obtained if the weight functions are constant over the entire domain. The FEM equivalents can also be reached if the weight functions are defined as piecewise-constant over each influence domain.

2.4 Imposing Essential Boundary Conditions

In general, the MLS shape functions lack the Kronecker delta function property of the FEM shape functions, which implies difficulties when imposing essential (displacement or kinematic) boundary conditions on the boundaries, discrete curves, and isolated points of the analysis domain. Thus, the Galerkin weak form formulation of Equation (1) has to be modified or constrained to overcome the problem.

We have adopted a constrained variational principle with a penalty function, which has roots in FEMs [10]. Consider the example problem of making a functional Π stationary subject to the unknown $\mathbf{u}$ obeying some set of additional constraints $\mathbf{C}(\mathbf{u}) = \mathbf{0}$ on the boundary of domain Ω. We would add to the original functional Π a penalty term

$$\Theta = \Pi + \frac{1}{2}\int_\Omega \mathbf{C}^T(\mathbf{u})\alpha\mathbf{C}(\mathbf{u})\,d\Omega \qquad (15)$$

where α is a penalty number vector. Using the Hamilton's principle, we come to the modified Galerkin weak form:

$$\int_\Omega \delta(\mathbf{L}\mathbf{u})^T\mathbf{c}(\mathbf{L}\mathbf{U})\,d\Omega - \int_\Omega \delta\mathbf{u}^T\mathbf{b}\,d\Omega - \int_{\Gamma_t}\delta\mathbf{u}^T\mathbf{t}\,d\Gamma - \int_\Omega \delta\frac{1}{2}\mathbf{C}(\mathbf{u})^T\alpha\mathbf{C}(\mathbf{u})\,d\Omega + \int_\Omega \rho\delta\mathbf{u}^T\ddot{\mathbf{u}}\,d\Omega = 0 \qquad (16)$$

For the most common Dirichlet boundary condition where displacement $\mathbf{u} = \bar{\mathbf{u}}$ is prescribed on the essential boundary Γ_u, $\mathbf{C}(\mathbf{u}) = \mathbf{u} - \bar{\mathbf{u}}$ is inserted into Equation 16.

In practice, the boundary conditions will not be properly enforced if the penalty factor α is too small. On the other hand, numerical problems will arise if it is too large. We have used the general rule of $\alpha = 10^{4-13} \times \max$ (diagonal elements in the stiffness matrix) as suggested in [16]. Further, α can be a varying function related to the trustworthiness of the prescribed boundary conditions, i.e. the reliability of the image data constraints.

2.5 Construction of System Equations

2.5.1 System Dynamics

From Equation 16, we arrive at the following governing system equations following the minimum work principles [6]:

$$M\ddot{\mathbf{U}} + C\dot{\mathbf{U}} + [K + K^b]\mathbf{U} = R + R^b \qquad (17)$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{n_t}]^T$ is the displacement vector with n_t the total number of nodes in the domain, M the mass matrix, C the damping matrix, K the stiffness matrix, R the external force, K^b the boundary condition penalty matrix, and R^b the boundary condition force:

$$\begin{aligned}
M_{I,J} &= \int_\Omega \rho\mathbf{\Phi}_I^T\mathbf{\Phi}_J\,d\Omega && (18)\\
K_{I,J} &= \int_\Omega \mathbf{S}_I^T\mathbf{c}\mathbf{S}_J\,d\Omega && (19)\\
C_{I,J} &= \lambda_1 M_{I,J} + \lambda_2 K_{I,J} \text{ (Rayleigh Damping)} && (20)\\
K^b_{I,J} &= \int_{\Gamma_u} \mathbf{\Phi}_I^T\alpha\mathbf{\Phi}_J\,d\Gamma && (21)\\
R^b_I &= \int_{\Gamma_u} \mathbf{\Phi}_I^T\alpha\bar{\mathbf{u}}\,d\Gamma && (22)
\end{aligned}$$

with

$$\mathbf{\Phi}_I = \begin{bmatrix} \phi_I & 0 \\ 0 & \phi_I \end{bmatrix}, \quad \mathbf{S}_I = \mathbf{L}\mathbf{\Phi_I} = \begin{bmatrix} \phi_{I,x} & 0 \\ 0 & \phi_{I,y} \\ \phi_{I,y} & \phi_{I,x} \end{bmatrix} \qquad (23)$$

where $\phi_{I,x}$ and $\phi_{I,y}$ represent the derivatives of the MLS shape functions with respect to x and y.

2.5.2 Evaluation of Integrals and Background Cells

In practice, in order to evaluate the entries of the system matrices, one needs to integrate over the problem domain or the curves for boundaries. This can be carried out through numerical techniques such as the Gauss quadrature [6], using a mesh of non-overlapping cells, called the *background mesh* in EFGM. The background cells are usually totally independent of the arrangement of nodes, such as the example in Figure 2 which shows a regular-grid background mesh used for the integration process. These cells also facilitate the identification of nodes which contributes to the discrete L_2 norm at a quadrature point as in Equation (3).

In FEM, the integration mesh is the same as the element mesh, which must be sufficiently fine with a sufficient number of integration points. The background mesh in EFGM,

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

however, is used merely for the integration of the system matrices but not for field variable interpolation, and needs to be properly designed to obtain an appropriate solution of desired accuracy [2]. In the regular-grid cell structure, there may exist cells that do not entirely belong to the analysis domain. It means that only a portion of a such cell belongs to the domain. A visibility scheme that automatically separates the portion of the cell which lies outside of the physical domain is employed. Another important question is the number of cells to be used. Following the spirit of [2], we have used $m_c \times m_c$ cells, where $m_c = \sqrt{(n_t)}$ and n_t is the total number of nodes in the domain. In each cell, Gauss quadrature is used, and the number of quadrature points depends on the number of nodes in a cell, and we have used $n_Q \times n_Q$ Gauss quadrature where $n_Q = \sqrt{m} + 2$ and m is the number of nodes in a cell. Alternatively, we have used 4×4 Gauss quadrature with reasonable results.

3 Applications and Experiments

We have applied the meshfree EFGM framework to several image analysis problems involving domain evolution and domain mapping. Here, we present the applications to 1D and 2D deformable models, i.e. contours and annuls, for object segmentation, and to multiframe nonrigid motion analysis of the left ventricle using biomechanical constraints.

3.1 Object Segmentation

We formulate the segmentation problems using active deformable models with inherent smoothness constraints (mathematical or mechanical) where the EFGM-derived PDEs define the dynamical behavior under the image forces that push the models interfaces towards object boundaries. This becomes the classical Snakes model [13] for 1D contour case and we use an elastic solid model for the detection of annulus objects in 2D. While the MPMs do not have obvious numerical advantages for 1D contour situations because node addition/elimination is trivial for contours, we use this classic model to illustrate the applicability of MPMs to non-mechanical PDE formulations.

3.1.1 Deformable Contour

Assuming parametric active contour $v(s) = (x(s), y(s))$, with $s \in [1, 0]$ the parameter, equilibrium of the model is obtained through the minimization of energy functional:

$$E = E_{data} + E_{model} \tag{24}$$
$$= \int_0^1 (w_1(s)|\frac{\partial v}{\partial s}|^2 + w_2(s)|\frac{\partial^2 v}{\partial s^2}|^2)\, ds + \int_0^1 P(v(s))\, ds$$

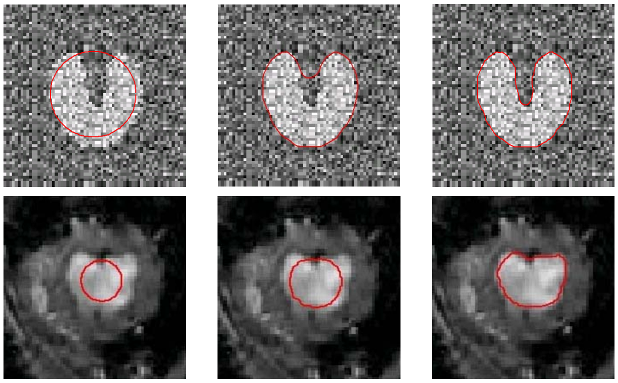

Figure 3: Top: active contour segmentation for synthetic image: initialization (left), FEM result (middle), and EFGM result (right). Bottom: EFGM solutions of the active contour segmentation for cardiac MRI image: initial, middle stage, and final results.

where $P(v(s))$ denotes a potential function defined from the image data. In our implementation, a force potential field is generated from the gradient vector flow (GVF) [22].

The contour $v(s)$ which minimizes the energy E must satisfy the Euler-Larange equation:

$$-\frac{\partial}{\partial s}(w_1 \frac{\partial v}{\partial s}) + \frac{\partial^2}{\partial s^2}(w_2 \frac{\partial^2 v}{\partial s^2}) + \nabla P(v(s)) = 0 \tag{25}$$

Applying the Galerkin principle to Equation (25), we have

$$\int_0^1 [-\frac{\partial}{\partial s}(w_1 \frac{\partial v}{\partial s}) + \frac{\partial^2}{\partial s^2}(w_2 \frac{\partial^2 v}{\partial s^2}) + \nabla P(v(s))]\chi(s)\, ds = 0 \tag{26}$$

where $\chi(s)$ is an arbitrary test function. Let $v = \phi_i u_i$ where ϕ_i is the MLS shape function, u_i a vector field variable for node i, and $U^T = [u_1, u_2, ..., u_n]$, undergoing the EFGM procedures developed earlier, Equation (26) yields

$$KU = R \tag{27}$$

where $R_i = \int_0^1 \phi_i^T t ds$, $t = -\nabla P_i(v(s))$, $K = \sum K_i$, and

$$K_i = K_\alpha + K_\beta \tag{28}$$
$$= \int_0^1 (\frac{\partial \phi_i}{\partial s})^T w_1 (\frac{\partial \phi_i}{\partial s}) ds + \int_0^1 (\frac{\partial^2 \phi_i}{\partial s^2})^T w_2 (\frac{\partial^2 \phi_i}{\partial s^2}) ds$$

Solving the equilibrium Equation (27), we arrive at the detected object boundary. Figure 3 shows the segmentation result comparison on noisy synthetic image ($SNR = 5dB$) using FEM (without node addition/elimination) and EFGM strategies, as well as the EFGM segmentation results of the left ventricle from MRI image.

3.1.2 Elastic Deformable Solid

For annulus-shape object (Figure 4), we use an elastic solid mechanical model for the simultaneous segmentation of

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

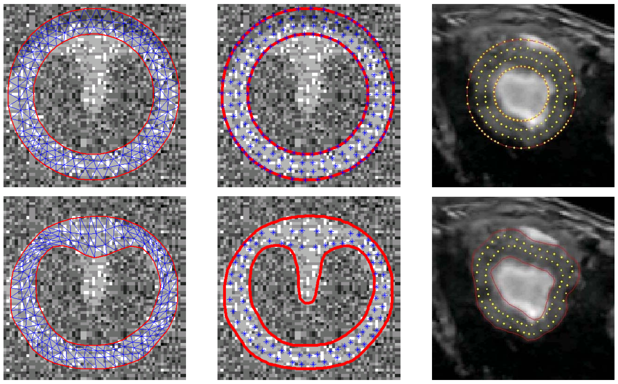

Figure 4: FEM (left) and EFGM (middle) solutions of the annulus-shaped elastic solid model segmentation for synthetic image, and EFGM results for cardiac MRI image (right): initialization (top) and final results (bottom).

both boundaries. The object is define by the two boundaries and scattered particles in-between. Once again, we relate the image GVF to our external data force t, and for equilibrium state, Equation (17) becomes

$$KU = \eta F \tag{29}$$

where external image force component $F_i = \int_\Gamma \phi_i^T t d\Gamma$ and η is a weighting constant. Figure 4 shows the FEM (without adaptive remeshing) and EFGM results on noisy synthetic image, and the EFGM results on a cardiac MRI image.

3.2 Multi-frame Cardiac Motion Analysis

Recovery of the non-rigid kinematics of the heart is essential in medical image analysis. Here, we apply the EFGM framework to the analysis of biomechanically constrained left ventricular motion throughout the cardiac cycle, relying on a state space strategy which performs optimal multi-frame estimation using H_2 filter (see details in [17]).

Assuming Markov process, the system dynamics equation (Equation 17) is transformed into a discrete-time state-space representation of a linear time-invariant stochastic system. Along with the associated measurement equation which describes the observed imaging data $y(t)$, we have

$$x(t+1) = Ax(t) + Bw(t) + v(t) \tag{30}$$

$$y(t) = Dx(t) + e(t) \tag{31}$$

where $A = e^{A_c \Delta T}$ and $B = A_c^{-1}(e^{A_c \Delta T} - I)B_c$, with

$$x(t) = \begin{bmatrix} \mathbf{U}(t) \\ \dot{\mathbf{U}}(t) \end{bmatrix}, \; w(t) = \begin{bmatrix} 0 \\ R + R^b \end{bmatrix} \tag{32}$$

$$A_c = \begin{bmatrix} 0 & I \\ -M^{-1}(K+K^b) & -M^{-1}C \end{bmatrix} \tag{33}$$

$$B_c = \begin{bmatrix} 0 & 0 \\ 0 & M^{-1} \end{bmatrix} \tag{34}$$

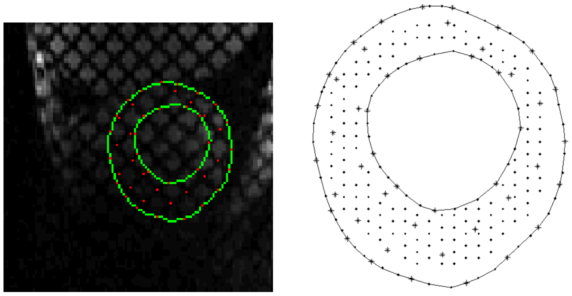

Figure 5: Segmented MRI tagging image and its meshfree particle representation.

D is a known measurement matrix, $v(t)$ is the zero-mean, white process noise ($E[v(t)] = 0, E[v(t)v(s)'] = Q_v(t)\delta_{ts}$, $e(t)$ is the zero mean, and white measurement noise ($E[e(t)] = 0, E[e(t)e(s)'] = R_e(t)\delta_{ts}$), and ΔT is the time interval. A recursive procedure is then used for the optimal state estimation based on H_2 filtering [17].

The MR tagging images are segmented and the tag-tag and tag-boundary crossing are detected for all sixteen images of the cardiac cycle (Figure 5). These crossings provide a set of 2D displacements between images frames, and are used as the prescribed boundary conditions in the EFGM framework. The state space representation is established, and the optimal estimates of the displacement field is achieved through the recursive process. Further, cardiac specific strain maps are derived (Figure 6).

4 Conclusions

we present a meshfree particle computational paradigm for computer vision problems. It is purely based on the nodal points and does not require the meshing of the analysis domain. This meshfree strategy can better handle large deformation and domain discontinuity issues and achieve desired numerical accuracy. We demonstrate the computational power of this framework with applications in object segmentation with deformable models, and multi-frame non-rigid motion analysis of the left ventricle using mechanical model. This work is supported by RGC-HKUST 6031/01E.

References

[1] T. Belystchko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47, 1996.

[2] T. Belystchko, Y.Y. Lu, and L. Gu. Element-free Galerkin methods. *Internaltional Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

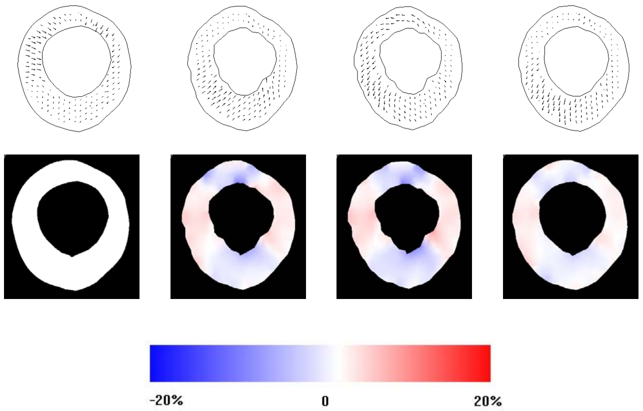

Figure 6: Frame-to-frame displacement field: #1-2, #5-6, #9-10, and #13-14; and cardiac-specific radial strain maps with respect to end diastole: frame #1, #5, #9, and #13.

[3] P. Breitkopf, A. Rassineux, G. Touzot, and P. Villon. Explicit form and efficient computation of mls shape functions and their derivatives. *Int J for Numerical Methods in Engineering*, 48:451–466, 2000.

[4] M. Bro-Nielsen. Finite element modeling in surgery simulation. *Proc of IEEE*, 86(3):490–503, 1998.

[5] L.D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.

[6] R.D. Cook, D.S. Malkus, M.E. Plesha, and R.J. Witt. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, New York, 2002.

[7] H. Delingette. Realistic soft-tissue modeling in medical simulation. *Proc of IEEE*, 86(3):512–523, 1998.

[8] J. Dolbow and T. Belytschko. An introduction to programming the meshless element free galerkin method. *Arch Computational Mechanics*, 5(3):207–241, 1998.

[9] M. Ferrant, A. Nabavi, B. Macq, F.A. Jolesz, R. Kikinis, and S.K. Warfield. Registration of 3-D intraoperative MR images of the brain using a finite-element biomechanical model. *IEEE Transactions on Medical Imaging*, 20(12):1384–1397, 2001.

[10] L. Gavete, J.J. Benito, S. Falcon, and A. Ruiz. Implementation of essential boundary conditions in a meshless method. *Communications in Numerical Methods in Engineering*, 16:409–421, 2000.

[11] W.C. Huang and D.B. Goldgof. Adaptive-size meshes for rigid and nonrigid shape analysis and synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):611–616, 1993.

[12] I. Kaljevic and S. Saigal. An improved element free galerkin formulation. *Int J for Numerical Methods in Engineering*, 40:2953–2974, 1997.

[13] M. Kass, A. Witkin, and D. Terzopoulos. SNAKES: Active contour models. *International Journal of Computer Vision*, 1:321–332, January 1988.

[14] P. Lancaster and K. Salkauskas. Surface generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981.

[15] S.F. Li and W.K. Liu. Meshfree and particle methods and their applications. *Applied Mechanics Review*, 55:1–34, 2002.

[16] G.R. Liu. *Mesh Free Methods*. CRC Press, Boca Raton, 2003.

[17] H. Liu and P. Shi. Meshfree representation and computation: applications to cardiac motion analysis. In *Information Processing in Medical Imaging*, in press.

[18] S. Sclaroff and A.P. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):545–561, 1995.

[19] N. Sukumar and B. Moran. c^1 natural neighbor interpolant for partial differential equations. *Numerical Methods for Partial Differential Equations*, 15(4):417–447, 1999.

[20] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformation: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(17), 1991.

[21] L.V. Tsap, D.B. Goldgof, and S. Sarkar. Nonrigid motion analysis based on dynamic refinement of finite element models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):526–543, 2000.

[22] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[23] A.A. Young, D.L. Kraitchman, L. Dougherty, and L. Axel. Tracking and finite element analysis of stripe deformation in magnetic resonance tagging. *IEEE Trans Medical Imaging*, 14(3):413–421, 1995.

[24] L.T. Zhang, W.K. Liu, S.F. Li, D. Qian, and S. Hao. Survey of multi-scale meshfree particle methods. *Lecture Notes in Computational Science and Engineering*, 26:441–458, 2002.