

Simulation of Tagged MR Images With Linear Tetrahedral Solid Elements

Kent J. Truscott, BS, MS¹ and Michael H. Buonocore, MD, PhD^{2*}

Computer simulation of tagged magnetic resonance (MR) images is presented using signals derived from connected solid triangular or tetrahedral elements. Simulation provides precisely-known deformations for validating tag quantification algorithms. MR imaging signal is derived by Fourier-transforming over a standard element, then mapping to actual elements via transformations. Linear intensity variations over the element are interpolated from vertex intensities determined by tag calculations. J. Magn. Reson. Imaging 2001;14:336–340. © 2001 Wiley-Liss, Inc.

Index terms: MRI; magnetic resonance imaging; simulation; tagging; deformation; grid analysis

MAGNETIC RESONANCE IMAGING (MRI) tagging (1) has made possible the in vivo measurement of tissue deformation. Measurement of myocardial deformation through the cardiac cycle has spurred rapid development of the technique, including methods to accurately quantify the deformation seen in the tagged images (2–7). These tag quantification methods have been verified using mathematical predictions compared with digitized tag analysis in a phantom (2), qualitative evaluation of image features (3), and computer-simulated images of known deformation (5).

Computer simulation is particularly important for verifying tag quantification methods. With simulation, the deformation is exactly known, non-ideal equipment behavior is not present to confound the interpretation of the data, and deformations can be modeled that are not easily generated physically. A large number of tests can be done without incurring the cost of MRI system usage. Previous simulation methods have been based mainly on signal derived from a discrete set of sampled points or pixels (8–10). Signal simulation based on continuous two-dimensional ellipses (11), which provides the motivation for the current work, stands out as an exception to the use of a discrete set of points, although it cannot model tagging or deformation, is limited to two

dimensions, and cannot depict non-elliptical structures well.

This paper presents the so-called tetrahedral simulation, a new method for solid simulation that represents structures of arbitrary size and shape as adjoining tetrahedrons for three dimensions and triangles for two dimensions. Each tetrahedron or triangle is associated with a continuous two- or three-dimensional analytical signal during MRI. The signal from each element is sampled and summed with the signal from the other elements. The prescribed image resolution of the MRI acquisition determines the number and spacing of signal sample points, which are independent of the definition of the structure being imaged. Deformed tags are easily modeled, and inversion of the deformation coordinates is not needed.

MATERIALS AND METHODS

The presented method simulates the radiofrequency signals of individual finite tissue elements such that the signal intensity emitted from each material point in the element varies linearly across the element. Each element has a signal that is accumulated with the other element signals in the k-space data array.

Tetrahedral elements are used for three-dimensional simulation and triangular elements are used for two-dimensional simulation because (a) any physical shape can be easily approximated with such elements, (b) tetrahedral and triangular element signals can be easily derived from the signal for a “standard element” (derived later), and (c) signal intensity can be unambiguously modeled as a linear function across such elements. The derivation below is for the three-dimensional/tetrahedron case; the two-dimensional/triangle case uses an analogous set of equations.

The first step is to represent the specimen to be imaged as a mesh of tetrahedral elements. The size of the mesh elements depends on the intended use of the simulation: for simple contrast modeling, the elements need only represent shapes to the user's satisfaction; for modeling tags and deformation, the mesh is iteratively refined until pixel intensities converge to a user-selected tolerance, such as less than 5% magnitude variation in any pixel in consecutive simulations with mesh sizes decreasing by a factor of two.

If tagging or other spatial intensity variation is desired, the relative signal intensity at each mesh vertex is

¹Biomedical Engineering Graduate Group, University of California, Davis, California.

²Department of Radiology, UCDCMC, UC Davis Imaging Center, Sacramento, California.

*Address reprint requests to: M.H.B., Department of Radiology, UCDCMC, UC Davis Imaging Center, Room 1215, 4701 X Street, Sacramento, CA 95817. E-mail: mhbuonocore@ucdavis.edu

Received July 2, 1999; Accepted April 20, 2001.

calculated with methods shown elsewhere (9) and tabulated. Then, any desired deformation is modeled by applying a displacement function, f , to the spatial coordinates R of each mesh vertex; $f(R) = r$.

The next step requires that each mesh vertex spatial location, r , be interpreted as having an alternative set of coordinates, ω , in a set of axes defined by the gradients used in the imaging process. The frequency- and phase-encoding gradients used for spatial encoding can be thought of as defining three “frequency axes”, with one axis corresponding to the frequency emitted by a material point during a particular data acquisition and the other two axes corresponding to the phase change from the phase-encode gradients.

The transformation from r to ω , $A_{r\omega}:r \rightarrow \omega$, is found by finding $A_{r\omega}^{-1}$ and then inverting; the emitted frequency at x cm along the frequency-encode axis is $\gamma \cdot |G_f| \cdot x$ radian/second, where G_f is the frequency-encoding gradient; therefore, $\omega = (1 \text{ rad/s}, 0, 0)$ is mapped to $1/\gamma|G_f|$ cm along that axis whose unit vector is $G_f/|G_f|$, so the first column of $A_{r\omega}^{-1}$ is $G_f/\gamma|G_f|^2$ with units of second \cdot cm/radians for proper unit conversion. At a distance of y cm along the first phase-encode direction, we define a “gradient frequency” $\gamma \cdot \Delta t_{\phi_1} \cdot y$ in units of rad/(gauss/cm), where Δt_{ϕ_1} is the phase-encode pulse width for the first-phase-encode gradient vector that has average increment of ΔG_{ϕ_1} . Therefore, $\omega = (0, 1 \text{ rad} \cdot \text{cm/g}, 0)$ is mapped to $1/\gamma\Delta t_{\phi_1}$ cm along that axis whose unit vector is $\Delta G_{\phi_1}/|\Delta G_{\phi_1}|$. The second column of $A_{r\omega}^{-1}$ can thus be $\Delta G_{\phi_1}/\gamma|\Delta G_{\phi_1}|^2\Delta t_{\phi_1}$ with units of gauss/radian. However, this expression’s dependence on the particular phase-encode pulse width, Δt_{ϕ_1} , can be removed in favor of user-visible parameters by noting that ΔG_{ϕ_1} is chosen so that it causes one cycle of phase offset across the field of view (FOV_{ϕ_1}) during Δt_{ϕ_1} ; $\gamma \cdot |\Delta G_{\phi_1}| \cdot \Delta t_{\phi_1} \cdot \text{FOV}_{\phi_1} = 1 \text{ cycle} = 2\pi$ radians. Solving for Δt_{ϕ_1} and substituting, the second column of $A_{r\omega}^{-1}$ becomes $\Delta G_{\phi_1} \cdot \text{FOV}_{\phi_1}/2\pi$ with units of g/rad. Similar reasoning is used for the other phase-encode axis for the third column of $A_{r\omega}^{-1}$. Inverting,

$$A_{r\omega} = \begin{bmatrix} G_f/\gamma|G_f|^2 \text{ s}\cdot\text{cm/rad} & \Delta G_{\phi_1}\cdot\text{FOV}_{\phi_1}/2\pi \text{ g/rad} \\ \Delta G_{\phi_2}\cdot\text{FOV}_{\phi_2}/2\pi \text{ g/rad} & \end{bmatrix}^{-1} \quad (1)$$

From this point on, all mesh point locations are in frequency coordinates, ω , not spatial coordinates, r .

Let $t = (t, g_1, g_2)$ denote the time and gradient “sample coordinates” of a signal sample during imaging, $t = n_1\Delta t$, $g_1 = n_2\Delta G_{\phi_1}$, $g_2 = n_3\Delta G_{\phi_2}$, where Δt , ΔG_{ϕ_1} , and ΔG_{ϕ_2} are the time and gradient increment amplitudes used between samples in the frequency and phase-encode directions, respectively, and n_1 , n_2 , and n_3 are integers. The specimen’s total signal value at sample grid point t can be expressed as the sum of all the element signals for sample t :

$$\text{signal}_{\text{total}}(t) = \sum_{\text{element}, e} \text{signal}_e(t) \quad (2)$$

The k-space data from the specimen is considered to trace out a *continuous* two- or three-dimensional surface with *continuous* independent variables t , g_1 , and g_2 .

This simulation treats this continuous surface as the specimen “signal”. A continuous signal function for each element is derived by integrating the signal for a point over the entire element volume. These expressions are evaluated for each point t on the k-space sample grid and accumulated for all elements in the k-space data array.

Equation 2 is now expanded in three ways: the signals from all the elements of a given material are combined, intensity factors (proton density [PD] and relaxation factor [T] due to T_1 and T_2 are kept in a separate table) are applied to the sum of element signals for the same material, and each element signal is rewritten as a frequency-shift exponential term, $e^{-j(\omega_{\text{offset}}t)}$, times the signal of an identical element with one vertex at the frequency coordinate origin (such an element is called DC-referenced):

$$\begin{aligned} \text{signal}_{\text{total}}(t) &= \sum_{\text{materials}, m} \left[PD_m \cdot T_m \cdot \sum_{\text{elements}, e} e^{-j(\omega_{\text{offset}}t)} \cdot \text{signal}_{e, \text{DC}}(t) \right] \quad (3) \end{aligned}$$

where ω_{offset} is the before-shift frequency coordinates of the vertex that is moved to the origin. This DC-referenced element is next mapped to a “standard element”.

The signal for an element with a vertex at the origin is derived from the signal for a “standard element”, a tetrahedron with vertices at the origin $D = (0,0,0)$ and at $A = (1,0,0)$, $B = (0,1,0)$, and $C = (0,0,1)$ on a set of axes for new “normalized frequency” variables v_1 , v_2 , and v_3 . If the edges of the tetrahedron connecting to the vertex at the origin are taken to be the basis vectors of a new coordinate system, let each edge be mapped to an edge of the standard element by the linear transformation S such that $\omega = Sv$, $v = (v_1, v_2, v_3)$. S is chosen to map from v axes to ω axes because S is easily found; its columns are the vectors describing the edges of the element to be mapped, arranged in a sequence such that they form a right-handed coordinate system.

The signal at a material point with frequency coordinates $\omega = (\omega_1, \omega_2, \omega_3)$ is represented as $e^{-j(\omega_1 t + \omega_2 g_1 + \omega_3 g_2)}$, with t , g_1 , and g_2 representing the coordinates of the sample point, t , as given earlier. The sum of products in the exponent is hereafter represented as a dot-product, $\omega \cdot t$. This exponential point function, scaled with an intensity function, is integrated over the element volume to get the total element signal:

$$\begin{aligned} \text{signal}_{e, \text{DC}}(t) &= \iiint_{\text{element}} \bar{I}_e(\omega) e^{-j(\omega \cdot t)} d\omega_1 d\omega_2 d\omega_3 \\ &= \int_0^1 \int_0^{1-v_1} \int_0^{1-v_1-v_2} I_e(v) e^{-j(t \cdot Sv)} |S| dv_3 dv_2 dv_1, \quad (4) \end{aligned}$$

where $\bar{I}_e(\omega)$ is the intensity function for the original element and $I_e(v)$ is the intensity function for the transformed element.

To accumulate samples from different element signals, each of which is mapped to the standard element

with a different linear transformation S , each k -space sample point t in Eq. [3] must be modified for each element. The dot-product expression in Eq. [4] is changed to matrix form to find the transformed sample input, $\mathbf{u} = (u_1, u_2, u_3): t \cdot S\mathbf{v} = t^T S\mathbf{v} = (t^T S)\mathbf{v} = (S^T t)^T \mathbf{v} = S^T t \cdot \mathbf{v}$. Therefore, let the time/gradients sample \mathbf{u} for the standard element be set according to the formula $\mathbf{u} = S^T t$:

$$\begin{aligned} \text{signal}_{e,DC}(\mathbf{t}) &= \int_0^1 \int_0^{1-v_1} \int_0^{1-v_1-v_2} I_e(\mathbf{v}) e^{-j(t \cdot S\mathbf{v})} |S| dv_3 dv_2 dv_1 \\ &= \int_0^1 \int_0^{1-v_1} \int_0^{1-v_1-v_2} I_e(\mathbf{v}) e^{-j(\mathbf{u} \cdot \mathbf{v})} |S| dv_3 dv_2 dv_1 \\ &= |S| \text{signal}_{std}(\mathbf{u}) = |S| \text{signal}_{std}(u_1, u_2, u_3) \end{aligned} \quad (5)$$

The linear intensity variation across an element, $I_e(\mathbf{v})$, is modeled as the sum of four functions: a constant function and three linear functions. If the element intensity is a , b , c , and d at vertices A , B , C , and D , respectively, then model the element's linear intensity as:

$$I_e(\mathbf{v}) = (a - d)v_1 + (b - d)v_2 + (c - d)v_3 + d \quad (6)$$

The signal for the constant function and the linear function for the first axis are shown in Table 1. The table only shows two functions each for the two- and three-dimensional cases because symmetry of the standard element allows the integral used for intensity variation in the v_1 direction to be used for v_2 and v_3 by changing the order of integration and then exchanging dummy variables. For example, derive the signal for linear intensity variation with v_3 by integrating with respect to v_1 first:

$$\begin{aligned} i_{v_3}(u_1, u_2, u_3) &= \int_0^1 \int_0^{1-v_3} \int_0^{1-v_3-v_2} v_3 e^{-j(u_1 v_1 + u_2 v_2 + u_3 v_3)} dv_1 dv_2 dv_3 \end{aligned}$$

Since v_1 and v_3 are integrated out and so can be treated as dummy variables, exchanging them produces:

$$\begin{aligned} i_{v_3}(u_1, u_2, u_3) &= \int_0^1 \int_0^{1-v_1} \int_0^{1-v_1-v_2} v_1 e^{-j(u_3 v_1 + u_2 v_2 + u_1 v_3)} dv_3 dv_2 dv_1 \\ &= i_{v_1}(u_3, u_2, u_1) \end{aligned}$$

demonstrating that the same functions can be used along both v_1 and v_3 axes by exchanging input variables. A similar procedure yields the integral for v_2 .

Table 1
Simulation Functions for Two-Dimensional and Three-Dimensional Elements

Two-dimensional intensity functions	
1. $I(\mathbf{v}) = \text{constant}$	$i_0(\mathbf{u}) = \int_0^1 \int_0^{1-v_1} (1) e^{-j(\mathbf{u} \cdot \mathbf{v})} dv_2 dv_1 = \frac{(e^{-ju_1} - 1)}{u_1 u_2} + \frac{e^{-ju_2}(e^{ju_2 - u_1}) - 1}{u_2(u_2 - u_1)}$
2. $I(\mathbf{v}) = \text{linear function}$	$i_{v_1}(\mathbf{u}) = \int_0^1 \int_0^{1-v_1} v_1 e^{-j(\mathbf{u} \cdot \mathbf{v})} dv_2 dv_1 = \frac{(e^{-ju_1}(ju_1 + 1) - 1)}{ju_2 u_1^2} + \frac{e^{-ju_2}(e^{ju_2 - u_1})(j(u_2 - u_1) - 1) + 1}{ju_2(u_2 - u_1)^2}$
	$i_{v_2}(u_1, u_2) = i_{v_1}(u_2, u_1)$
Three-dimensional intensity functions	
1. $I(\mathbf{v}) = \text{constant}$	$i_0(\mathbf{u}) = \int_0^1 \int_0^{1-v_1} \int_0^{1-v_1-v_2} (1) e^{-j(\mathbf{u} \cdot \mathbf{v})} dv_3 dv_2 dv_1 = \frac{e^{-ju_1} - 1}{ju_1 u_2 u_3} + \frac{e^{-ju_3} - e^{-ju_1}}{ju_3(u_2 - u_3)(u_1 - u_3)} + \frac{e^{-ju_2} - e^{-ju_1}}{ju_3(u_1 - u_2)} \left(\frac{1}{u_3 - u_2} + \frac{1}{u_2} \right)$
2. $I(\mathbf{v}) = \text{linear function}$	$i_{v_1}(\mathbf{u}) = \int_0^1 \int_0^{1-v_1} \int_0^{1-v_1-v_2} v_1 e^{-j(\mathbf{u} \cdot \mathbf{v})} dv_3 dv_2 dv_1 = \frac{c_2(\mathbf{u})}{(u_1 - u_2)^2} [e^{j(u_2 - u_1)}(j(u_1 - u_2) + 1) - 1] + \frac{c_3(\mathbf{u})}{(u_1 - u_3)^2} [e^{j(u_3 - u_1)} \times (j(u_1 - u_3) + 1) - 1] + \frac{c_{23}(\mathbf{u})}{u_1^2} [e^{-ju_1}(ju_1 + 1) - 1]$
	where $c_2(\mathbf{u}) = \frac{e^{-ju_2}}{u_3} \left(\frac{1}{(u_3 - u_2)} + \frac{1}{u_2} \right)$, $c_3(\mathbf{u}) = \frac{-e^{-ju_3}}{u_3(u_3 - u_2)}$, and $c_{23}(\mathbf{u}) = \frac{-1}{u_2 u_3}$.

$\mathbf{v} = (v_1, v_2)$ and $\mathbf{u} = (u_1, u_2)$ for the two-dimensional case, similar for the three-dimensional case. \mathbf{v} corresponds to the standard frequency domain and \mathbf{u} corresponds to the standard sampled domain.

Using all the terms in Table 1, the total signal for a single element is shown in the *linear intensity equation*:

$$\begin{aligned} \text{signal}_{e,DC}(\mathbf{t}) &= |S| \text{signal}_{std}(\mathbf{u}) = |S| ((a - d)i_{v_1}(u_1, u_2, u_3) \\ &\quad + (b - d)i_{v_1}(u_2, u_1, u_3) + (c - d)i_{v_3}(u_3, u_2, u_1) \\ &\quad + di_0(u_1, u_2, u_3)) \end{aligned} \quad (7)$$

Using Table 1, this equation is evaluated at the point specified by \mathbf{u} , accumulated in the k -space data array, and the total reconstructed into an image.

In summary, the complete simulation process is:

- I. Generate a mesh of elements to represent the structures to be imaged.

- II. Determine the relative signal intensity for each mesh vertex due to tagging or other spatial intensity variation, such as slice- or volume-selection.
- III. Displace the mesh vertices to model deformation; $f(\mathbf{R}) = r$.
- IV. From the image prescription, determine the k-space sample grid t and the vectors \mathbf{G}_r , $\Delta\mathbf{G}_{\phi 1}$, and $\Delta\mathbf{G}_{\phi 2}$.
- V. Transform the mesh vertex coordinates using Eq. [1] and $\omega = A_{r\omega}r$.
- VI. For each material, sample, accumulate, and scale signals for all elements for that material.
 - A. For each element of this material, determine its signal samples and accumulate them.
 1. Using the ω values, determine the linear mapping, $\mathbf{S}: \mathbf{v} \rightarrow \omega$, of the standard element to the element edges.
 2. Find the transformed samples, $\mathbf{u} = (u_1, u_2, u_3) = \mathbf{S}^T t$.
 3. At the sample points \mathbf{u} , evaluate the linear intensity equation, Eq. [7], using the expressions in Table 1.
 4. As per Eq. [3], multiply the current element's samples times the ω_{offset} factor and accumulate into an array for the current material.
 - B. Scale the accumulated data for the current material according to the PD and relaxation parameters (T_1, T_2) of the material, and add that array to the total k-space data array according to Eq. [3].
- VII. Reconstruct the image from $\text{signal}_{\text{total}}(t)$.

The standard two-dimensional element is a triangle with vertices at (0,0), (1,0), and (0,1). Its intensity function is modeled with three functions instead of four (one "constant level" function and two linear functions).

Two two-dimensional simulations were run as demonstrations. The first demonstration was a variety of objects with three different materials, chosen to show the versatility of the shapes that can be simulated easily. Only 19 basic shapes were needed to define all the objects: two triangles, five rectangles, three circles, and nine partial circles. A mesh routine took each basic shape and broke it into elements. Holes were created by specifying a shape inside another shape with PD equal to the negative of the larger shape. One material was embedded in another by specifying a hole and then specifying the new material in the shape of the hole.

The second demonstration simulated a silicone gel phantom to represent a realistic in-plane (two-dimensional) torsion of a structure that had already been imaged and analytically described (2); a similar object was simulated in (9). The deformation used here was derived by Young, et al from the Mooney-Rivlin strain energy function (12). Letting θ be the angular coordinate of a material point, R the radial coordinate, R_1 the inner radius, R_2 the outer radius, and Z the coordinate perpendicular to the plane of the deformation, then the displacement functions for each of the cylindrical coordinates are: $Z_{\text{def}}(R, \theta, Z) = Z$,

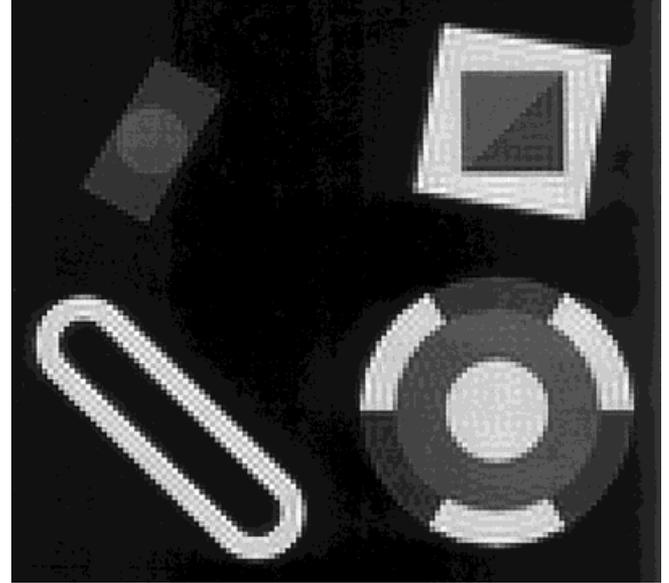


Figure 1. A variety of objects to demonstrate the solid simulation. Realistic edge transitions and Gibbs artifact are present. Upper left: 1 rectangle, 2 circles, including one circular hole; upper right: 2 rectangles, including one rectangular hole, and 2 triangles; lower left: 2 partial circles, and 2 rectangles, including one rectangular hole; lower right: 7 partial circles and 1 circle.

$R_{\text{def}}(R, \theta, Z) = R$, and $\theta_{\text{def}}(R, \theta, Z) = \theta + \Delta\theta(R)$, where $\Delta\theta(R) = \theta_1 \left(\frac{R_2^{-2} - R^{-2}}{R_2^{-2} - R_1^{-2}} \right)$ and θ_1 is the angle by which the inner surface is rotated. Due to the forcing constraints placed on the object, the deformation is in-plane (two-dimensional) despite the phantom being three-dimensional.

The algorithm was implemented and run with MATLAB, version 5.0 (The MathWorks, Natick, MA).

RESULTS

Figure 1 shows various assembled shapes. The contrast between different materials was easily visible, with between-materials borders meshing cleanly. Edge pixels showed smooth intensity transitions. Half-circles and rectangles in the lower-left object combined smoothly to create a closed, hollow object. Gibbs artifact was visible at the edges.

Figure 2 shows the simulation of the silicone gel phantom previously imaged and described by Young, et al (2). Figure 2a shows a simulation of the undeformed phantom, and Figure 2b shows a simulation of the same phantom after the inner surface was rotated 45°. As in Figure 1, edges and tags had smooth transitions with no evidence of individual elements.

DISCUSSION

A method of simulating MR images with signals from solid elements instead of point-samples has been derived and qualitatively demonstrated. The motivation is to have a set of images wherein tag deformations are

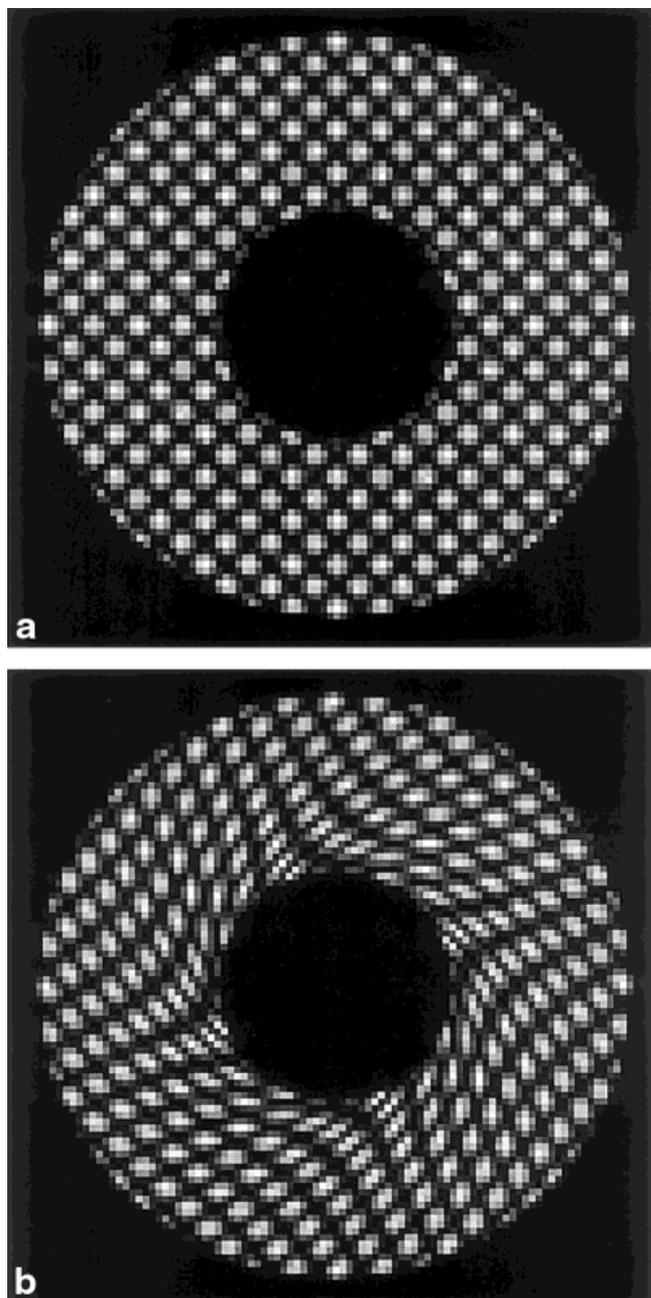


Figure 2. Simulation of axial shear for a silicone gel phantom. **a:** Undeformed structure; outer radius = 47.6 mm, inner radius = 19.0 mm, two sets of 2nd-order tags 90° apart, 5 mm tag spacing. **b:** Simulation of the same phantom where the outer border has not moved, while the inner border has been rotated 45° counter-clockwise.

precisely known so tag analysis software can be validated. Verifying the simulation quantitatively, a subject of later work, will include comparing predicted versus imaged contrast, measuring Gibbs artifact peak and valley amplitudes, and measuring the shapes of edge transitions.

The demonstrated method generates realistic presentations of tagging, deformation, and contrast with either two- or three-dimensional models; simulations that do not include tagging or deformation can be done with fewer, larger elements. This method has both advantages and disadvantages compared to other simulation methods. Advantages include: the structure definition is independent of the image resolution whereas other methods (8–10) require defining sample points or materials for each pixel; structures of any shape can be assembled from basic shapes to an approximation as accurate as the user requires, versus being restricted to only one shape (11); and no deformation inversion is needed. A disadvantage is that adequate simulation of complex, deformed structures with tagging may require many elements to get the necessary convergence of imaging results, with the necessary cost of more computational time.

Tetrahedral simulation can be extended in a number of directions, such as adapting mesh generation to interact with commercial mesh-generation software, modeling field inhomogeneity with a deformation function that maps the mesh vertices to their effective locations, adding noise to the k-space data, and simulating specific MR k-space trajectories such as echo-planar or spiral to see the influence of their spatial resolution limitations.

REFERENCES

1. Axel L, Dougherty L. MR imaging of motion with spatial modulation of magnetization. *Radiology* 1989;171:841–845.
2. Young AA, Axel L, Dougherty L, Bogen DK, Parenteau CS. Validation of tagging with MR imaging to estimate material deformation. *Radiology* 1993;188:101–108.
3. Guttman MA, Prince JL. Image analysis methods for tagged MRI cardiac studies. In: *Medical Imaging IV: Image Processing*, Loew, MH (Editor); Proceedings of SPIE—the International Society for Optical Engineering, (vol 1233) Bellingham, 1990. p 168–175.
4. Bazille A, Guttman MA, McVeigh ER, Zerhouni EA. Impact of semi-automated versus manual image segmentation errors on myocardial strain calculation by magnetic resonance tagging. *Invest Radiol* 1994;29:427–433.
5. Amini AA, Chen Y, Curwen RW, Mani V, Sun J. Coupled B-snake grids and constrained thin-plate splines for analysis of 2-D tissue deformations from tagged MRI. *IEEE Trans Med Imaging* 1998;17:344–356.
6. Bundy JM, Lorenz CH. TAGASIST: a post-processing and analysis tools package for tagged magnetic resonance imaging. *Comput Med Imaging Graph* 1997;21:225–232.
7. McVeigh ER. MRI of myocardial function: motion tracking techniques. *Magn Reson Imaging* 1996;14:137–150.
8. Petersson JS, Christoffersson JO, Golman K. MRI simulation using the k-space formalism. *Magn Reson Imaging* 1993;11:557–568.
9. Crum WR, Berry E, Ridgway JP, Sivananthan UM, Tan L-B, Smith MA. Simulation of two-dimensional tagged MRI. *J Magn Reson Imaging* 1997;7:416–424.
10. Crum WR, Berry E, Ridgway JP, Sivananthan UM, Tan L-B, Smith MA. Frequency-domain simulation of MR tagging. *J Magn Reson Imaging* 1998;8:1040–1050.
11. Buonocore MH, Perlmutter RJ, Kirk GL, Shepp LA. An analytic simulation method for magnetic resonance imaging. In: *Book of Abstracts, Society of Magnetic Resonance in Medicine, Fourth Annual Meeting, London, 1985*; pp. 34–35.
12. Green AE, Zerna W. *Theoretical elasticity*. 2nd edition. London: Oxford University; 1968. 457 p.