

Ultrasound Elastography: A Dynamic Programming Approach

Sepehr Ghamari - 40273405

Shabnam Ghareh Mohammadi - 40239861

Ala Keramati - 40255113

Concordia University

Abstract: We endeavored to develop a 2-D strain imaging technique utilizing dynamic programming (DP) by computing a cost function. Initially, we adopted a 1-D DP approach, utilizing sets of pre-compression and post-compression images. The deformation of tissue was considered solely in the axial direction for a single dimension. Following this, we computed the cost function according to the specifications in the original paper Rivaz et al., 2008. Subsequently, we generated a displacement map of the post-compression image concerning the pre-compression image. Following the displacement map calculation, we formulated a strain function to determine the gradient of the displacement map, resulting in the generation of the strain image, which allowed for clear tumor visualization. To enhance the outcomes, we then implemented 2-D Dynamic Programming

1. INTRODUCCIÓN

Elastography, an innovative medical imaging technique for evaluating tissue elasticity variations, holds promise for applications such as tumor detection. Garra et al., 1997 This study focuses on static elastography, an established method combining quasi-static tissue compression with ultrasound imaging. Ultrasound analysis yields a tissue displacement map, processed through the least squares technique to generate a low-noise strain estimate. Greenleaf et al., 2003

Despite its clinical potential, static elastography faces challenges hindering widespread adoption. Signal decorrelation between pre- and post-compression images introduces significant noise, particularly during high axial compression, nonaxial probe movements, and physiological motion. Many elastography methods rely on correlation analysis of radiofrequency (RF) echoes, necessitating large windows that, while reducing variance in displacement estimation, also limit tolerable compression levels.

Mechanical actuators are often used to minimize signal decorrelation in compression Varghese and Ophir, 1997, but freehand palpation elastography, requiring

Draft no extra hardware, is preferred despite introducing unwanted operator hand motion. Additionally, the computational expense of elastography poses challenges for real-time elastogram display, crucial for image-guided surgical operations. Existing attempts at real-time elastography face efficiency issues.

In a closely related study by Pellet-Barakat et al., 2004, they proposed a method that minimizes an energy function by combining constraints related to echo amplitude conservation and displacement continuity. Recognizing that data alone may be insufficient to resolve ambiguities arising from signal decorrelation, the incorporation of physical priors regarding tissue motion continuity enhances the robustness of their technique. Their approach involves upsampling RF data by a factor of four in the axial direction, followed by subdividing the image into four parts. They iteratively calculate a coarse displacement map for each part. This process is repeated, with each part further divided into four parts, and the displacement of each segment is calculated using the same iterative technique, initialized with the displacement of the parent grid. While this method is demonstrated to generate accurate and low-noise displacement fields, it is reported to have a computation time exceeding 1 minute for a strain image with fewer pixels than those in the strain images generated in the current paper. Conse-

quently, this method may not be immediately suitable for real-time elastography.

In line with these challenges, this paper introduces a dynamic programming (DP)-based elastography technique for image matching, demonstrating its feasibility. DP, an efficient noniterative global optimization method, offers potential in reducing computational costs.

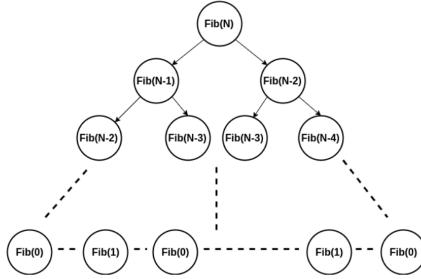


Figure 1: DP

2. Theory

2.1. Dynamic Programming (DP)

Dynamic programming is an optimization algorithm that breaks down complex problems into smaller overlapping subproblems, solving each subproblem only once and storing its solution for future reference, making decisions iteratively to efficiently optimize a cost function. Cormen et al., 2022

Dynamic programming is often applied to optimization problems, where the goal is to find the best solution from a set of feasible solutions. One of the key features of dynamic programming is the concept of "memoization" or storing intermediate results to avoid redundant calculations, which helps improve the efficiency of the algorithm.

So as an example of dynamic programming we can consider to the Fibonacci sequence. This is a series of numbers in which each number (Fibonacci number) is the sum of the two preceding ones, usually starting with 0 and 1.

The recursive definition of the Fibonacci sequence is given by Lew and Mauch, 2006:

$$F(n) = F(n-1) + F(n-2) \quad (1)$$

with base cases:

$$F(0) = 0 \quad F(1) = 1 \quad (2)$$

This recursive definition can lead to redundant calculations, as the same Fibonacci numbers are computed multiple times (Figure 1). Dynamic programming, specifically memoization, can be applied to optimize this computation. This example illustrates the essence of dynamic programming: breaking down a problem into overlapping subproblems, solving each subproblem only once, and storing the solutions for reuse. This approach significantly reduces the time complexity of the algorithm.

2.2. Displacement 1-D

In this section, our analysis revolves around a pair of images— one captured before compression and the other after compression. These images are of identical size, ensuring a one-to-one correspondence between each pixel before and after compression. We aim to estimate the displacement of individual pixels about their pre-compression positions.

Initially, we presume that the displacement falls within a range, for instance, between -4 and 4, and is exclusively in the axial direction. This implies that a pixel can be displaced up to 4 positions backward or forward. Our process begins by focusing on the estimation of the first line in the first column of the pre-compression image. We compare this line with the first column of the post-compression image only in axial.

Subsequently, we consider displacements within the range of -4 to 4. For each displacement value, we calculate the associated cost. Our goal is to identify the displacement that incurs the minimum cost, signifying optimal alignment. This selected displacement is then chosen for further analysis and processing.

$$\Delta(i, d) = g(i) - g'(i + d) \quad (3)$$

In order to mitigate the impact of these alterations on Δ (Delta), both precompression and postcompression ultrasound images undergo normalization by dividing them by the maximum value from either image. The smoothness of the displacements is denoted as S .

$$S(d_i, d_{i-1}) = (d_i - d_{i-1})^k \quad (4)$$

As previously mentioned, our goal is to estimate each

pixel by determining the displacement with the minimum cost at a particular point. This process is independently applied to calculate the displacement map for all A-lines. To achieve this, we utilize the cost function formula and minimize it as part of our analysis.

$$C(i, d_i) = \min_{d_{i-1}} \{C(i-1, d_{i-1}) + \omega S(d_i, d_{i-1})\} + \Delta(i, d_i) \quad (5)$$

We then employ dynamic programming (DP) for further analysis. Following the computation of C at each step, we store the results in a matrix, eliminating the need to recalculate it entirely for subsequent steps.

By storing the minimum costs in the matrix, we store the corresponding displacements that result in the minimum costs (desired displacements) in the M matrix. This matrix provides information about the number of displacements for each line.

$$M(i, d_i) = \operatorname{argmin}_{d_{i-1}} \{C(i-1, d_{i-1}) + \omega S(d_i, d_{i-1})\}_{\text{aft}} \quad (6)$$

2.3. Displacement 2-D

Up to this point, our consideration for 1-D displacement has been limited to the axial direction. However, in the context of 2-D displacement, we expand our scope to include both axial and lateral dimensions. We calculate the minimum cost associated with the displacement.

$$C_j(d_a, d_l, i) = \min_{\delta_a, \delta_l} \left\{ \frac{1}{2} (C_j(\delta_a, \delta_l, i-1) + C_{j-1}(\delta_a, \delta_l, i)) + \omega S(d_a, d_l, \delta_a, \delta_l) + \Delta(d_a, d_l, i) \right\} \quad (7)$$

This acknowledges the significance of the displacements not only axial but also in lateral dimension, so all the displacements, smoothness and cost function change accordingly.

$$\Delta(i, j, d_a, d_l) = |g_j(i) - g_{j+d_l}'(i + d_a)| \quad (8)$$

$$S(d_{a_i}, d_{l_i}, d_{a_{i-1}}, d_{l_{i-1}}) = (d_{a_i} - d_{a_{i-1}})^2 + (d_{l_i} - d_{l_{i-1}})^2 \quad (9)$$

3. Methodology

We implemented the methodology outlined in this paper using MATLAB and for the data we used RF data. After regularizing hyperparameters, we employed dynamic programming (DP) to compute the 1-D displacement within the range of [-100, 0], determined through a trial-and-error approach. Subsequently, we calculated the strain image, utilizing the LSQ [Rivaz, n.d.] method referenced in the GitHub repository. The strain images were computed using windows 43 and 73.

Expanding our analysis, we computed the 2-D displacement, maintaining the same range for axial displacement but adjusting the range for lateral displacement to [-4, 0]. The strain calculation followed the same procedure as in the previous step. To compare the results, we computed mathematical unitless metrics known as Signal-to-Noise Ratio (SNR) and Contrast-to-Noise Ratio (CNR).

SNR is a measure used to assess the quality of a signal by comparing the strength of the desired signal to the level of background noise present. In the context of medical imaging, a higher SNR indicates a clearer and more reliable signal, making it easier to distinguish relevant information from noise. Gonzalez, 2009

$$\text{SNR} = \frac{\bar{s}}{\sigma} \quad (10)$$

In this project based on the paper, we select 10 random windows from the entire image and compute the SNR for each window. Subsequently, we determine the average SNR across all these windows.

It is a measure that evaluates the contrast between two regions in an image relative to the background noise level. CNR is particularly relevant in medical imaging when assessing the visibility and distinction of structures, such as tumors or abnormalities, from the surrounding tissue. A higher CNR indicates better visibility and contrast between different features in the image. Gonzalez, 2009

$$\text{CNR} = \frac{\sqrt{2(\bar{s}_b - \bar{s}_t)^2}}{\sqrt{\sigma_b^2 + \sigma_t^2}} \quad (11)$$

So in this project, we partition the entire image into

36 background windows. Additionally, we consider a target window that contains the tumor . CNR is computed for both the target windows and the background windows, resulting in 36 values. Subsequently, a histogram is derived from these 36 values, and the average is calculated for comparison purposes.

4. Results and Discussion

For experimental evaluation, RF data was acquired from an Antares Siemens system (Issaquah, WA) with a 7.27-MHz linear array at a sampling rate of 40 MHz. The original dataset, its state before compression, and the resultant data post-compression are illustrated in Figure 2.

4.1. One Dimension (1D)

Concerning the outcome of the 1-D displacement, the hyperparameters include a ω value of 0.15, a range of axial displacement from -100 to 0, and a minimum window size for minimization set at 3.

The displacement map created by the code and a median filtered version is shown in Figure 3.

The algorithm mentioned earlier was employed to compute the strain image, as depicted in Figure 4. Strain calculations utilized window sizes of 43 and 73.

To begin, the displacement map reveals a discernible image featuring a centrally located, denser tumor. However, clarity is compromised, particularly on the right side, exhibiting noisy stripes—a likely outcome of lateral displacement during image acquisition. As a result, axial displacement fails to accurately depict the shift between the two images.

Next, the application of a median filter effectively diminishes noise, yet introduces undesired blurriness to the image.

Lastly, the calculated strain images exhibit the expected tumor presence in the image center. However, precise information is lacking at the image's extremities due to the absence of reliable displacement in these regions.

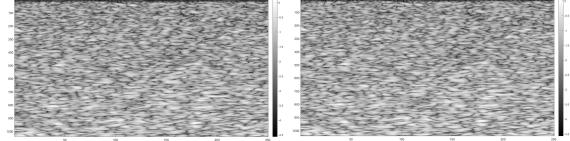


Figure 2: *Original Data Before and After Compression*

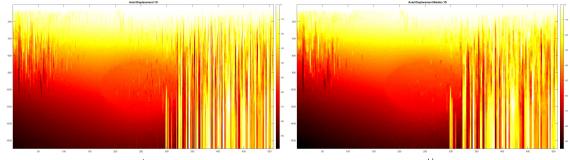
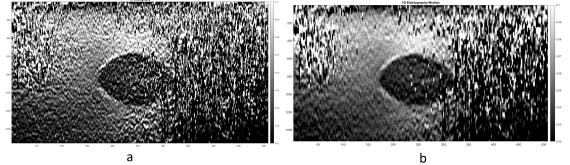


Figure 3: a) 1-D Displacement Map b) 1-D Displacement Map + Median Filter



Draft
Figure 4: 1-D Strain a) window 43 b) window 73

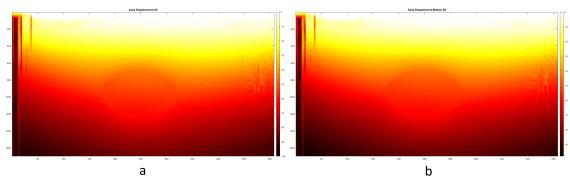


Figure 5: a) 2-D Displacement Map b) 1-D Displacement Map + Median Filter

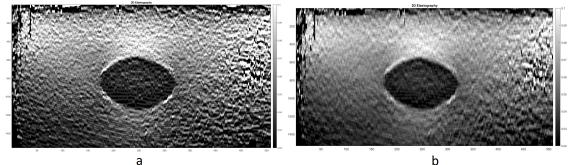


Figure 6: 2-D Strain a) window 43 b) window 73

4.2. Two Dimension (2D)

The algorithm in two dimensions was run for the same sample ultrasound images shown in Figure 2. The hyperparameters include a W value of 0.15, an axial displacement range from -100 to 0, lateral displacement

ranging from -4 to 4, and a minimum window size for minimization set at 3. The displacement map created by the code and a median filtered version are shown in Figure 5. Strain image was calculated based on the algorithm previously described and it is shown in Figure 6.

The improved displacement map, as compared to Figure 3, exhibits enhanced clarity, allowing for easy identification of the tumor owing to an increased contrast ratio. Notably, the previously observed noisy stripes are absent, affirming lateral displacements in the right and left sections of the images. The image clarity is such that the application of a median filter proves unnecessary in this instance.

The strain images present a significantly improved depiction, accurately outlining the tumor's location and boundaries. While there is still a slight background noise, it doesn't obscure the target itself. Additionally, the absolute values for the target are within a narrower range compared to the noisier version in Figure 6.

For the performance metrics, improvements were made on all of them. The Signal-to-Noise Ratio (SNR) is computed by randomly selecting 10 different windows across the entire image and subsequently averaging the results. The data presented in Table 8 indicates that employing a two-dimensional discrete wavelet transform yields a higher SNR compared to the one-dimensional method. Furthermore, the application of a median filter demonstrates an improvement in the obtained results.

In the subsequent phase, Contrast-to-Noise Ratio (CNR) is computed between the target window and 36 background windows. Figure 9 displays the normalized histogram of CNR, revealing that the 2D method produces superior results with a broader distribution, featuring higher values across numerous bins compared to the 1D method. Additionally, the average CNR for each method is calculated and presented in Table 8, highlighting that the CNR value for 2D DP is approximately three times greater than 1D DP. Moreover, the application of a median filter enhances the obtained results.

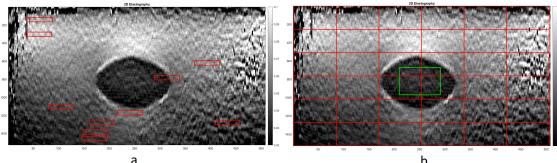
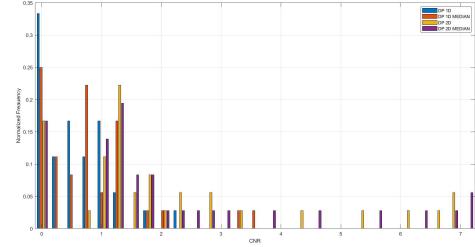


Figure 7: a) SNR b) CNR Windows

Method	SNR	Method	CNR
1D DP	40.5892	1D DP	0.6750
1D DP + Median	47.5327	1D DP + Median	0.9299
2D DP	63.9497	2D DP	2.2169
2D DP + Median	66.4052	2D DP + Median	2.3127

a b

Figure 8: a) Averaged SNR b) Averaged CNR



Draft
Figure 9: Normalized CNR values of the lesion, obtained by dividing each bin by the total of 36 CNR measurements.

To enhance visualization, the 2D strain image is graphically represented in a three-dimensional plot (10). It is evident that pixels housing the lesion exhibit lower intensity compared to surrounding pixels. This visualization underscores the efficacy of our method in accurately detecting tumors in the presence of displacement.

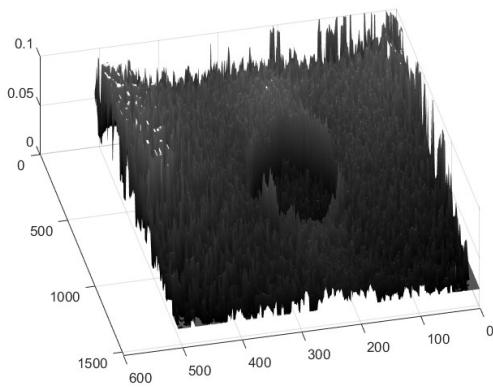


Figure 10: 2D Strain

5. Conclusion

The outcomes indicate the successful implementation of the Dynamic Programming methodology outlined in Rivaz et al., 2008. The 1D application produced noisy results as it solely considered the axial direction. Conversely, the 2D approach yielded superior results due to its sophistication, accounting for both axial and lateral displacement.

In terms of performance metrics, including CNR and SNR, the 2D method demonstrated better performance compared to the 1D approach. As a suggestion for future work, the team proposes coding in C or an equivalent lower-level programming language to enhance runtime efficiency.

Furthermore, readers are encouraged to explore additional research papers, specifically Rivaz et al., 2011 and Hashemi and Rivaz, 2017, as these introduce newer approaches to the same problem after grasping the dynamic programming approach.

REFERENCES

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. MIT press.
- Garra, B. S., Cespedes, E., Ophir, J., Spratt, S., Zuurbier, R. A., Magnant, C. M., & Pennanen, M. (1997). Elastography of breast lesions: Initial clinical results. *Radiology*, 2021, 79–86. <https://api.semanticscholar.org/CorpusID:27401372>
- Gonzalez, R. C. (2009). *Digital image processing*. Pearson education india.
- Greenleaf, J. F., Fatemi, M., & Insana, M. (2003). Selected methods for imaging elastic properties of biological tissues [PMID: 12704084]. *Annual Review of Biomedical Engineering*, 5(1), 57–78. <https://doi.org/10.1146/annurev.bioeng.5.040202.121623>
- Hashemi, H. S., & Rivaz, H. (2017). Global time-delay estimation in ultrasound elastography. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 64(10), 1625–1636.
- Lew, A., & Mauch, H. (2006). *Dynamic programming: A computational tool* (Vol. 38). Springer.
- Pellet-Barakat, C., Frouin, F., Insana, M., & Hermant, A. (2004). Ultrasound elastography based on multiscale estimations of regularized displacement fields. *IEEE Transactions on Medical Imaging*, 23(2), 153–163. <https://doi.org/10.1109/TMI.2003.822825>
- Rivaz, H. (n.d.). *Ultrasound elastography*. https://users.encs.concordia.ca/~hrivaz/Ultrasound_Elastography/
- Rivaz, H., Boctor, E., Foroughi, P., Zellars, R., Fichtinger, G., & Hager, G. (2008). Ultrasound elastography: A dynamic programming approach. *IEEE transactions on medical imaging*, 27(10), 1373–1377.
- Rivaz, H., Boctor, E. M., Choti, M. A., & Hager, G. D. (2011). Ultrasound elastography using three images. *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2011: 14th International Conference, Toronto, Canada, September 18-22, 2011, Proceedings, Part I* 14, 371–378.
- Varghese, T., & Ophir, J. (1997). Enhancement of echo-signal correlation in elastography using temporal stretching. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 44(1), 173–180. <https://doi.org/10.1109/58.585213>