Assessing the Speed and Accuracy of Real-time Motion Tracking Algorithms for Abdominal Organ Motion Management in a 1.5T MR-Linac System

Mohammed Mehdi Shahid[1,3], Travis Salzillo, PhD[2], and Jihong Wang, PhD[3]

[1] Department of Physics & Astronomy, Carleton College, Northfield, Minnesota,
[2] Department of Radiation Oncology, MD Anderson Cancer Center, Houston, Texas,[3] Department of Radiation Physics, MD Anderson Cancer Center, Houston, Texas

THE UNIVERSITY OF TEXAS MD Anderson Cancer Center

INTRODUCTION

Real-time magnetic resonance imaging (MRI) using the novel MR-Linac provides the opportunity for organ motion tracking during MRI-guided radiotherapy.
Managing organ motion is important in radiotherapy to mitigate normal tissue toxicities. Real-time accurate organ motion tracking will enable physicians to further personalize radiotherapy treatment plans.

Purpose

We investigated the feasibility of three algorithms for real-time organ motion tracking. With the results of the project, we aim to implement a real-time organ motion tracker on a 1.5T MR-Linac system.

MATERIALS & METHODS

We tested 3 algorithms based on template matching to determine the 2D displacement of an abdominal target region. The script for Normalized Cross Correlation (NCC) was already set up[1]. We developed the script to use Mutual Information as a tracking algorithm. Finally, we implemented OpenCV's Kernelized Correlation Filter (KCF) for organ motion tracking.

Normalized Cross Correlation (NCC)

$$NCC = \cos\theta = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{|a||b|} = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2}\sqrt{\sum_i b_i^2}}, -1 \le NCC \le 1$$

- In its simplest form, NCC represents the strength of correlation between two vectors **a** and **b**.
- Using a fast Fourier transform (FFT) approach described in **Ref. 1**, we can increase the speed efficiency of the NCC algorithm[1].

Mutual Information (MI)

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p_{(X,Y)}(x,y)\log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\,p_Y(y)}\right)$$

- MI is calculated between two variables and measures the reduction in uncertainty (entropy) for one variable given a known value of the other variable[2].
- MI is common correlation coefficient and can be used as an image matching metric[2].

OpenCV's Kernelized Correlation Filter (KCF)

- A novel tracking framework that utilizes properties of circulant matrices and fast Fourier transforms (FFT)[3].
- KCF can be implemented easily with the OpenCV python package and can run hundreds of frames-per-second[3].

Assessing Spatial Accuracy and Computational Speed

Spatial accuracy was determined using a ground truth script that measured the displacement of pixels for a given region frame-by frame (**Fig. 1**). Computational speed was measured using the in-built Python time counter.

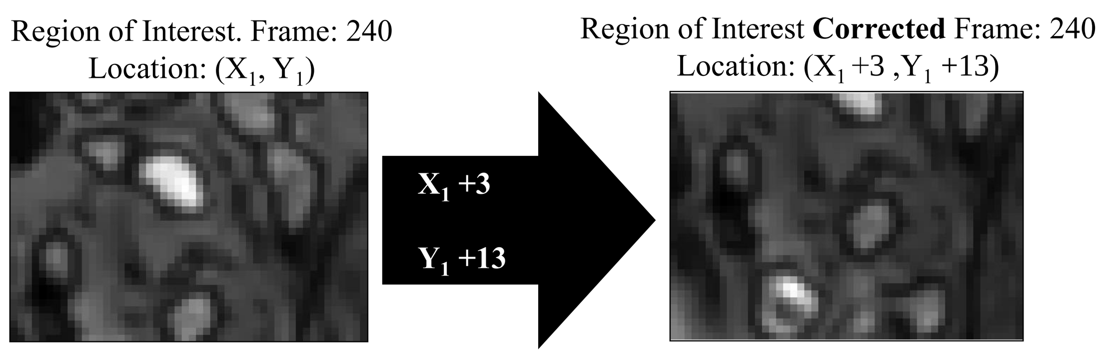

Fig 1: Procedure for measuring the ground truth displacement of a target region used to measure spatial accuracy of the algorithms. This figure is an example where the total displacement was 13.34.

SPATIAL ACCURACY RESULTS

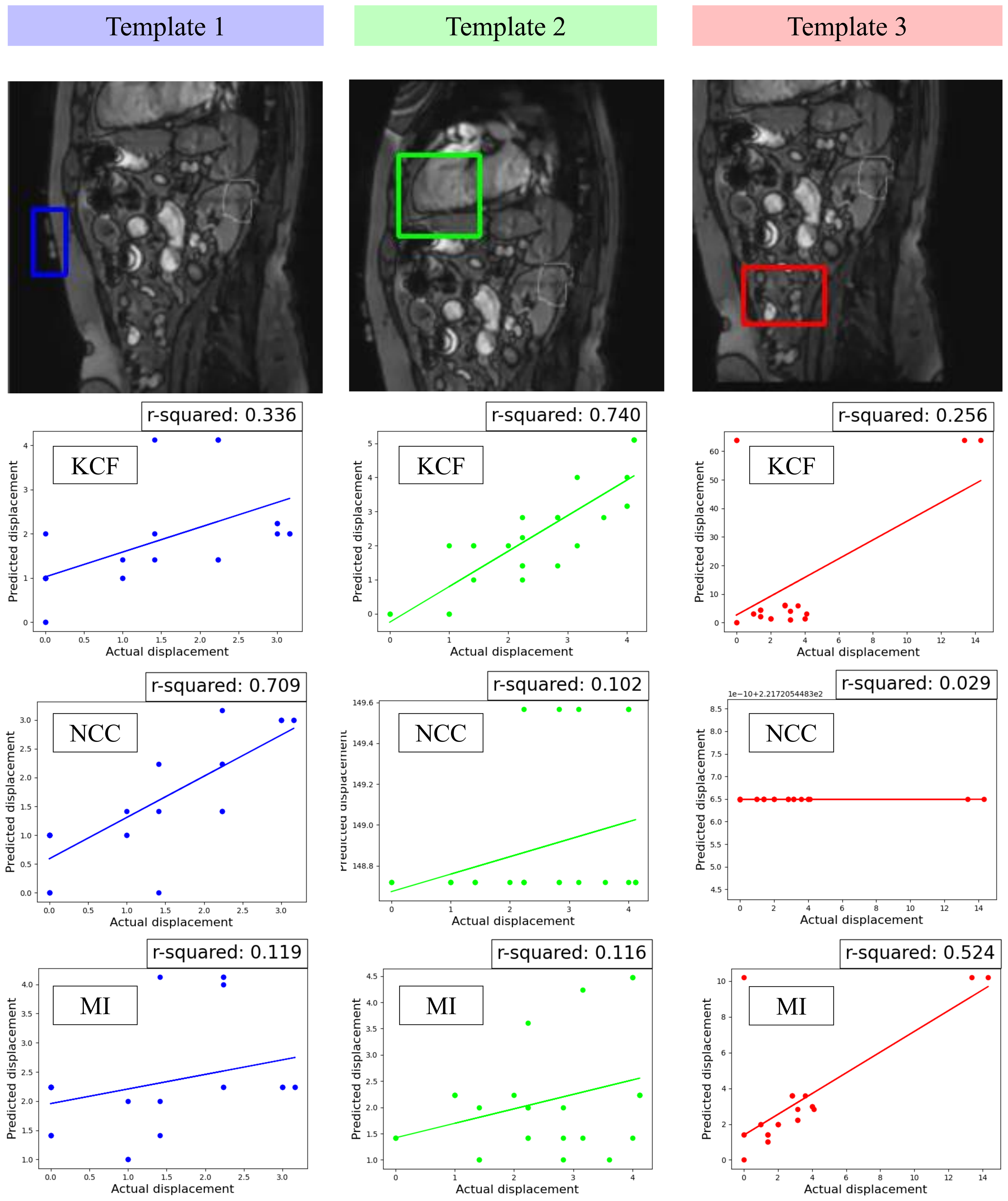

Fig 2: Correlation plots showing the predicted displacement vs. the actual displacement for each template for each algorithm. A higher r-squared value denotes that the predicted displacement by the algorithm matched to a higher degree with the actual displacement of a template region.

Comparing the predicted displacement with the actual displacement for each algorithm in three different target locations we find that the KCF and MI algorithms perform better than the NCC (**Fig. 2**). NCC is reasonably accurate for Template 1 (vitamin E beads) but is unable to detect the target location in templates 2 and 3. KCF appears to correlate the most, on average, with the actual displacement and is therefore the most feasible algorithm.

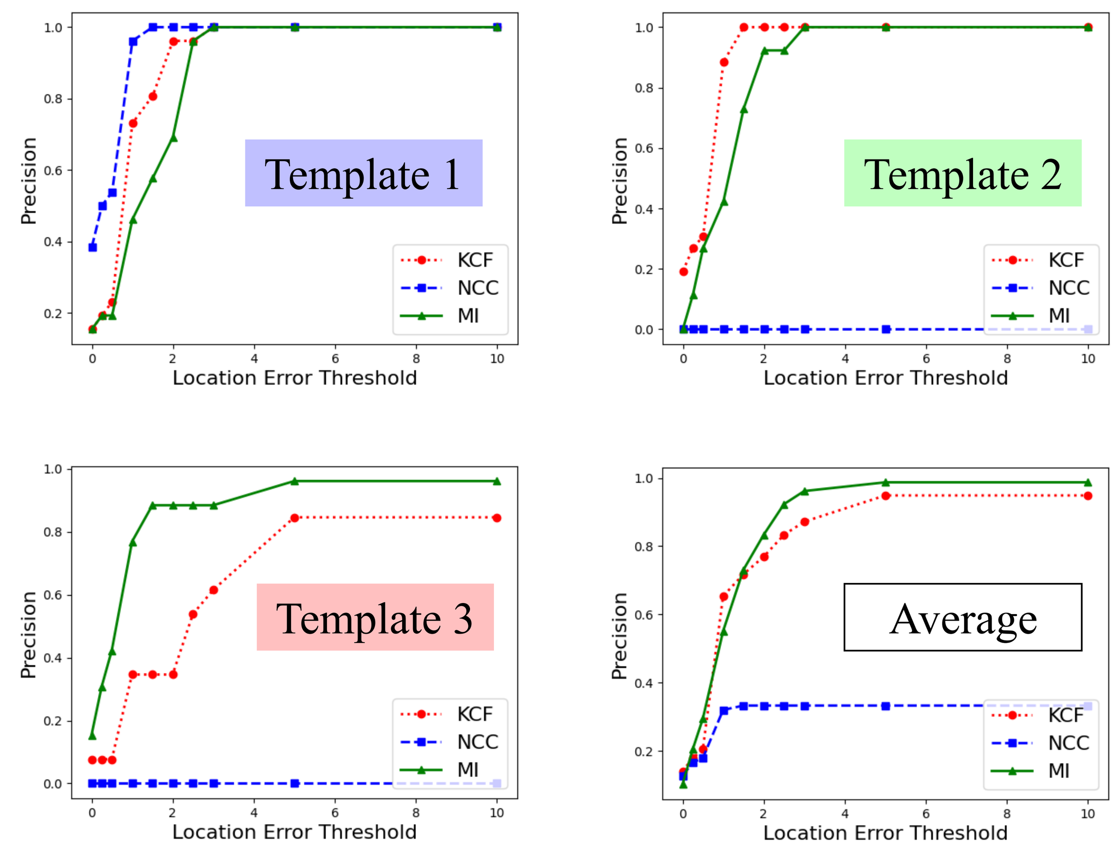

Fig 3: Precision plots showing the performance measure of the three algorithms. From top-left to bottom-right: Template 1, Template 2, Template 3, Average. A higher precision score at lower thresholds denotes a batter performance.

- Evaluating a precision score shows the percentage of correctly tracked frames for a range of distance thresholds.
- A higher precision at low thresholds corresponds to higher spatial accuracy.
- KCF and MI show similar precision curves, on average.
- Variability between templates shows that the nature of the target region being tracked can affect the spatial accuracy of the given algorithm.

COMPUTATIONAL SPEED RESULTS

Template	Time per frame (ms)		
	KCF	NCC	MI
Template 1	9.27	101.38	15810
Template 2	15.08	113.19	18560
Template 3	12.72	100.34	20490
Average	**12.36**	**104.97**	**18290**

KCF had the fastest computational speed followed by NCC. MI was computationally inefficient and so is not applicable to the real-time motion tracking of abdominal organs.

Both KCF and NCC utilize fast Fourier transforms (FFTs), and so this data indicates that FFTs are a prospect for fast real-time motion tracking. The MI algorithm used does not utilize any FFTs.

CONCLUSION & FUTURE WORK

- KCF and NCC are computationally efficient in motion tracking. MI is too computationally expensive. While MI is parallelizable, it would still be computationally inefficient compared to KCF and NCC.
- KCF and MI are more spatially accurate than NCC. KCF performs on overall better than the NCC and MI in terms of template variability, precision and accuracy.

Future Work

We aim to implement an organ motion tracker in the clinic using OpenCV's KCF tracker via a Raspberry Pi and LED Strip system. The workflow, as seen in **Fig. 4**, will utilize the video signal provided from the MR-Linac to the console to power a LED strip. The LED strip will signal patient whether they are breathing too hard.

Future work involves:

- Implementing the Raspberry Pi system with the MR-Linac.
- Testing the system for end-to-end time, reproducibility and accuracy using volunteer studies.

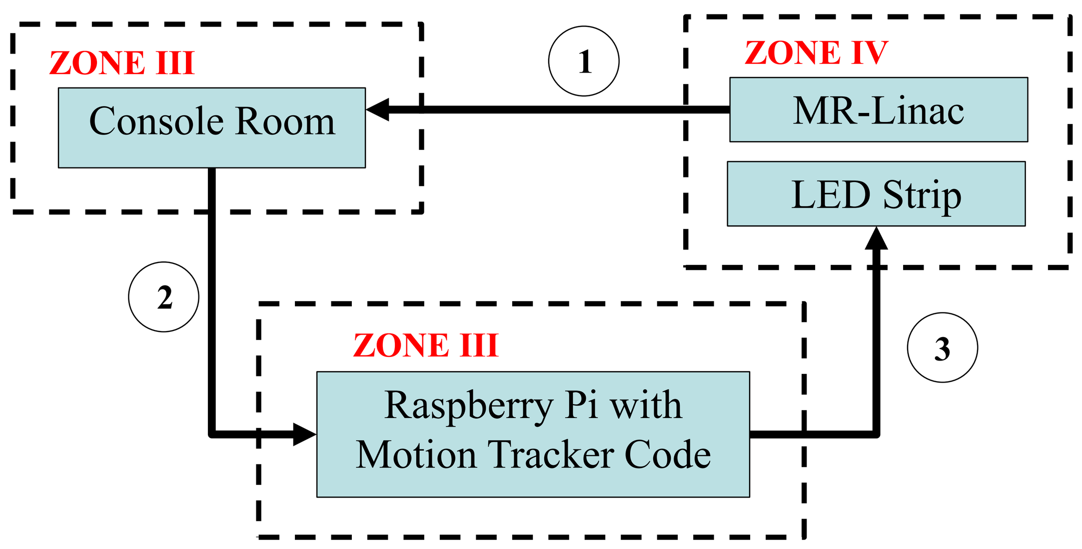

Fig 4: Workflow for organ motion tracking with appropriate safety zone marking. **1**: During treatment, MR-Linac sends video input to the console. **2**: The Raspberry Pi uses the console's screen output to initiate tracking with the motion tracking script. **3**: The Pi sends results of motion tracking to a LED strip that signals the patient.

ACKNOWLEDGEMENTS

We would like to thank Dr. Yao Ding, PhD for their suggestions on the further feasibility testing of the motion tracker in the clinic. This presentation is supported by the CPRIT-CURE program and the Carleton Career Center. For further information, please contact Mohammed Mehdi Shahid at shahidm@carleton.edu

References

1. A. Kaso, PLOS ONE 13 (9), e0203434 (2018).. https://doi.org/10.1371/journal.pone.0203434.
2. T.M. Cover and J.A. Thomas, Elements of Information Theory. (Wiley, 2012).
3. J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, IEEE Transactions on Pattern Analysis and Machine Intelligence **37** (3), 583 (2015)