

# MULTIPLE REGION OF INTEREST TRACKING OF NON-RIGID OBJECTS USING DEMON'S ALGORITHM

Rohan Pillai<sup>1</sup>, Abhishikta Yalavali<sup>2</sup> Saima Mohan<sup>1</sup>, and Amol Patil<sup>1</sup>

<sup>1</sup>iGate, Tower 3, Magarpatta City, Pune, India,  
{rohan.pillai,saima.mohan,amol.vpatil}@igate.com,  
<sup>2</sup>abhishikthaharsha@gmail.com

## **ABSTRACT**

*In this paper we propose an algorithm for tracking multiple ROI (region of interest) undergoing non-rigid transformations. Demon's algorithm based on the idea of Maxwell's demon, has been applied here to estimate the displacement field for tracking of multiple ROI. This algorithm works on pixel intensities of the sequence of images thus making it suitable for tracking objects/regions undergoing non-rigid transformations. We have incorporated a pyramid-based approach for demon's algorithm computations of displacement field, which leads to significant reduction in the convergence speed and improvement in the accuracy. This algorithm is applied for tracking non-rigid objects in laproscopy videos which would aid surgeons in Minimal Invasive Surgery (MIS).*

## **KEYWORDS**

*Computer vision, Maxwell's demon, multi-resolution pyramid, non-rigid deformation, optical flow, ROI tracking*

## **1. INTRODUCTION**

Tracking a region of interest is well-known in image processing community. In its long history, a number of algorithm have been designed that address this problem quite successfully<sup>1</sup>. In some real world situations, objects tend undergo non-rigid transformation and are thus difficult to track. These scenarios especially are prominent in medical applications where usually the region under consideration undergoes non rigid transformation mainly due to camera movement. In this paper we address the problem of tracking multiple region of interest (ROI) whose geometric appearance are changing gradually over time. We demonstrate the application of this algorithm for tracking multiple ROI's in a video recorded during endoscopy.

In our previous paper<sup>6</sup>, we have successfully demonstrated the use of demon based technique for tracking an ROI in video sequence. In our current paper, we refine our earlier work by considering the entire image and using a multi-resolution approach for calculating the displacement field. This introduction of multi-resolution approach has resulted in improved

computational efficiency and accuracy of the tracking algorithm which will be discussed in detail in subsequent sections of this paper. The overview of demons algorithm along with its use in tracking is presented in section 2. The multi-resolution approach is discussed in section 3. Discussions, results and conclusion appear in section 4, 5 and 6 respectively.

## 2. DEMON BASED TRACKING

Thirion<sup>7</sup> introduced the demons forces for registration of images acquired from different modalities. The concept behind his experiment was to deform one image to match another. The work involved calculation of displacement of a physical point present in two images with the support of gradient based optical flow computation. The expression for displacement field of two images is as given in equation (1)

$$\vec{u}(p) = \frac{(M(p) - S(p)) \vec{\nabla} S(p)}{(\vec{\nabla} S(p))^2 + (M(p) - S(p))^2} \quad (1)$$

Here  $\vec{u}(p) = [u_x(p) \ u_y(p)]^T$  is the displacement along x and y direction of a point, p, in the static image S with reference to a movable image M.  $\vec{\nabla} S(p)$  is the gradient at point p on the image S. If  $S'$  is the image to be deformed then movable image M is constructed using a deformation operator as given in equation (2) and (3).

$$M = S' \circ \vec{u} \quad (2)$$

$$M(p) = S'(p + \vec{u}(p)) \quad (3)$$

Thirion had re-normalized the optical flow equation by introducing the intensity difference between the images in the denominator. Due to this, the displacement of each point has to be calculated iteratively until the squared error is minimized. The objective of image registration technique is to solve the minimization problem, to find the displacement field  $\hat{U}$  equation (??),

$$\hat{U} \quad (4)$$

where, d is the dissimilarity metric to measure the similarity between two images.

Furthermore, to improve convergence speed and stability of the registration process, He Wang et al.<sup>8</sup> introduced the forces of the movable image.

$$\vec{u}(p) = \frac{(M(p) - S(p)) \vec{\nabla} S(p)}{(\vec{\nabla} S(p))^2 + \alpha (M(p) - S(p))^2} + \frac{(M(p) - S(p)) \vec{\nabla} M(p)}{(\vec{\nabla} M(p))^2 + \alpha (M(p) - S(p))^2} \quad (5)$$

Here,  $\alpha$  is a positive homogenization factor introduced by Cachier et al.<sup>4</sup> to bound the displacement field.

Tracking objects in a video can be accomplished by calculating the interframe displacement of the objects. Assuming that the scene being recorded is not undergoing any rapid movement or change, equation (5) gives us the displacement of points in one image with reference to the next image. The displacement field is Gaussian smoothed for regularization<sup>2</sup>. This displacement field is used to track a Region of Interest (ROI) in a video.

The metric function,  $d$ , used in this paper is Sum of Squared Difference (SSD) of the images.

### 3. PYRAMIDAL IMPLEMENTATION

Bouquet<sup>3</sup> demonstrated the use of pyramidal implementation in computing the optical flow algorithm based on Lucas Kanade method. The pyramid representation used in this paper is similar to that proposed by Bouquet. The method specified by Burt and Adelson<sup>9</sup> is used to construct recursively the pyramidal representation of the images  $I_{t-1}$  and  $I_t$  as:  $\{I_{t-1}\}^{m=0,1,\dots,L}$  where  $m$  is the pyramid Level. The value  $L$  is height of the pyramid. This value can be heuristically programmed by incorporating an estimation technique. Depending upon the expected inter frame displacement,  $L$  can take values 2, 3 or 4. In the image pyramid, image at  $m^{\text{th}}$  level is at  $\frac{1}{4}$  the resolution of the  $(m-1)^{\text{th}}$  level. The algorithm starts with the  $L^{\text{th}}$  level images and the initial displacement field initialized to a zero vector.

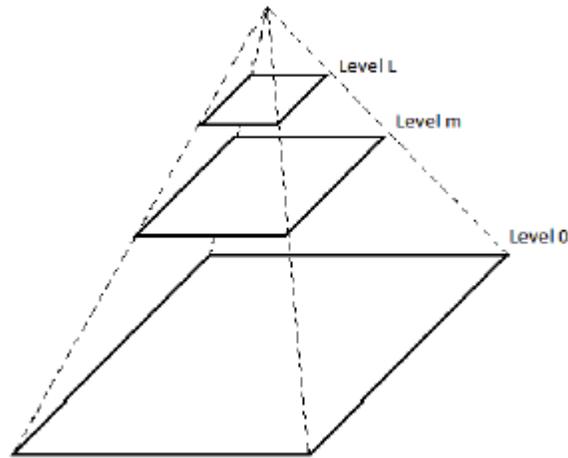


Fig. 1. Illustration showing hierarchy of pyramid structure of an image

The displacement field is iteratively computed for the images at this level until the termination criteria is satisfied. The termination criteria is defined by the iteration count or the rate of change of error. The estimated displacement field at the  $L^{\text{th}}$  level is then rescaled to the images at the  $(L - 1)^{\text{th}}$  level in the pyramid and used as the initial displacement field for estimation. The estimation requires that the displacement field be updated with values computed as per equation (5) so as to decrease the error. This process is repeated until the displacement field for the  $0^{\text{th}}$  level (original) images in the pyramid is obtained.

Figure 2 summarizes the Pyramidal Implementation of Demon's based algorithm as a flowchart.

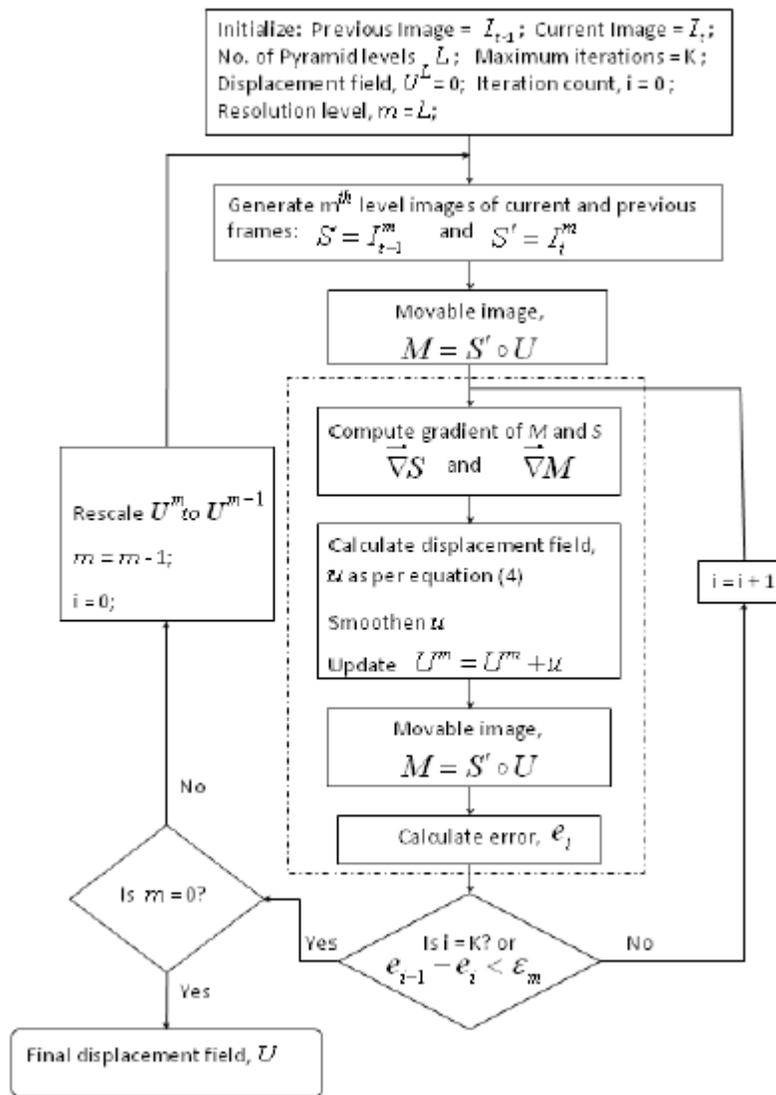


Fig. 2. Sequential execution of Pyramidal Implementation of Demon's based algorithm

#### 4. DISCUSSION

The ROI marked by a surgeon corresponds to an array of underlying pixels in an image. The displacement field calculated by the above algorithm represents the distance of a physical point in a frame being shifted with reference to the corresponding point in the previous frame. This displacement field can be applied to one or more ROI's which will propagate over the temporal sequence without any manual intervention. Using this principle multiple ROI's in the image can be tracked.

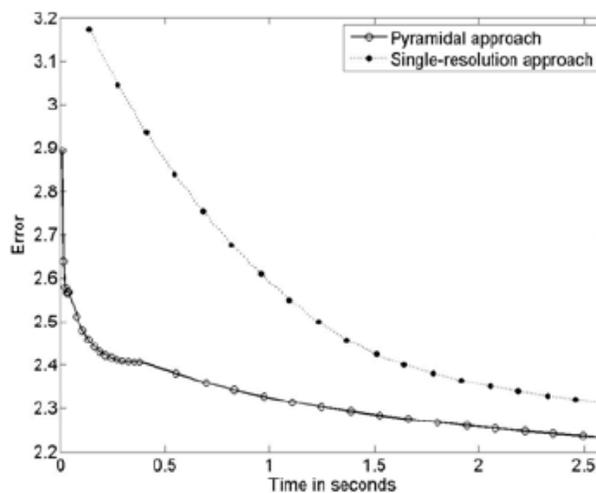
The total number of computations performed on an  $n_x \times n_y$  image would be  $Kmn_x n_y$ , where  $m$  computations are performed on each pixel of an image per iteration and the image converges after  $K$  iterations. Thus the computation time needed is directly proportional to the size of the image.

With the introduction of pyramidal approach it is observed that the total computation time has reduced significantly. The displacement field calculated at lower resolution images is refined as we work towards the higher resolution images. Hence, resulting in fewer iterations at subsequent levels of the pyramid to converge. The computational advantage gained is illustrated in the performance chart shown in figure 3(a).

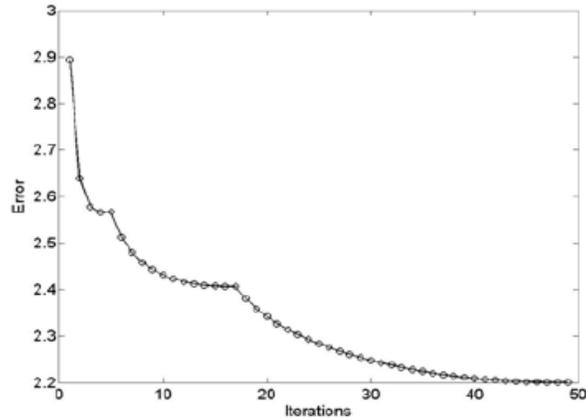
For successive iteration the model image has a higher resemblance to the static image. Thus the error metric, SSD, that measures the dissimilarity between the two images is expected to decrease with each iteration. However, the SSD metric is image resolution dependent hence can not be used while comparing convergence between different pyramid levels. To establish a benchmark when comparing the displacement field produced at each level in the pyramid, we rescale the displacement field at each iteration to full image size and then use it to compute error with respect to the original image. Figure 3(b) shows the graph of error convergence with respect to iterations for single resolution and pyramidal approach. This confirms that the displacement field calculated at every level in the pyramid facilitates the overall convergence. The two results can be compared as they are operating on the same base images. In Figure 3(b) a sharp drop in error is observed at iteration number 5 and 18 which is caused by transition from one level in the pyramid to another.

An Interesting observation about the type of convergence can be noted in figure 3 when plotted with respect to iteration count, the convergence graph looks like a series of decaying exponential. However, when the convergence graph is plotted against time, it looks more like a fast decaying exponential. Pyramidal implementation provides the freedom of making certain parameters like, smoothing kernel, termination criteria and the positive homogenization factor level dependent. The performance of the system can be further improved by selecting suitable values for these parameters.

Pyramidal implementation provides the freedom of making certain parameters like, smoothing kernel, termination criteria and the positive homogenization factor level dependent. The performance of the system can be further improved by selecting suitable values for these parameters.



(a)



(b)

Fig. 3. Performance graph. (a) Plot depicting the time taken by single resolution (dotted line) and the pyramidal implementation (dashed line) to converge. (b) Graph of error calculated after every iteration for pyramidal implementation

## 5. RESULTS

The performance of the algorithm was tested on a stream of synthesized images involving the motion of two blobs in arbitrary directions. For the numerical experiment, synthetic images of size 512x512 are generated with objects of interest imposed on a noisy background. To construct a noise model in our synthetic image, we used Gaussian noise (having zero mean and 0.005 variance) along with salt and pepper noise (0.005 probability) generated by MATLAB. For the first experiment figure 4, the objects considered are rigid bodies moving in different directions. Figure 4(a) indicates the initial state of objects with green marker enclosing them and figure 4(b) illustrates the process of tracking for an intermediate frame in the simulated video sequence.

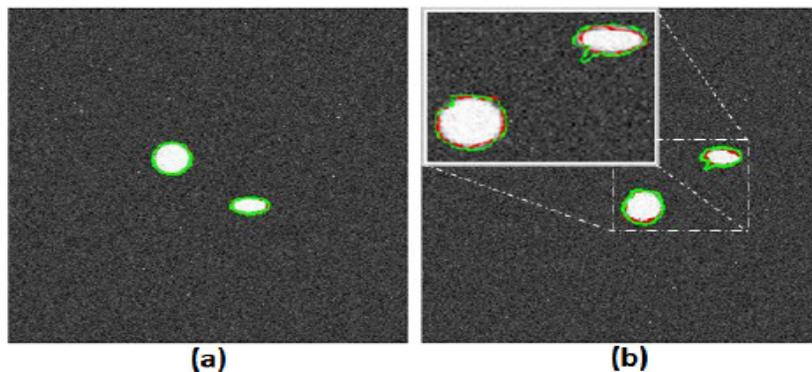


Fig. 4. Comparison of single resolution approach (green marker) and multi-resolution approach (red marker) for tracking multiple objects over time. (a) Second frame in the video (b) Result of tracking ROI's after 42<sup>th</sup> frame.

In medical imaging, regions of interest being filmed tend to undergo affine transformation due to continuous change in the camera position. The applicability of the discussed algorithm to track such non rigid objects is demonstrated in figure 5. It can be observed that the red line continues to

capture the complete object including the gradual change in appearance of the object. The array corresponding to the ROI to accommodate for non-rigid transformation. As the size of the deformable region changes over time the length of the marker array should be made adaptive to accommodate this.

Figure 6 shows the results for an endoscope video with a resolution of 640x480 as input. The algorithm works on the green channel of the video. The ROI's shown in blue are marked by the end user for tracking. The region undergoes non-rigid deformation over time which is tracked by the algorithm.

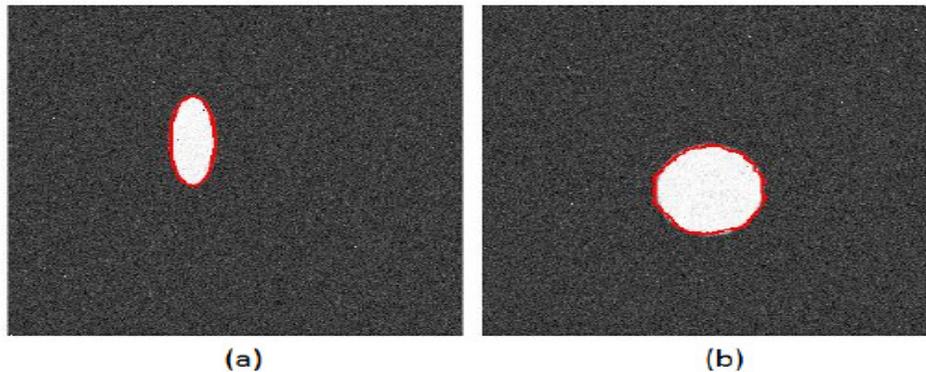


Fig. 5. Illustration of experimental setup for tracking an object deforming over time. (a) First frame in the video where red marker indicates the ROI. (b) After 39<sup>th</sup> frame, the red marker continues to enclose the region which has undergone deformation

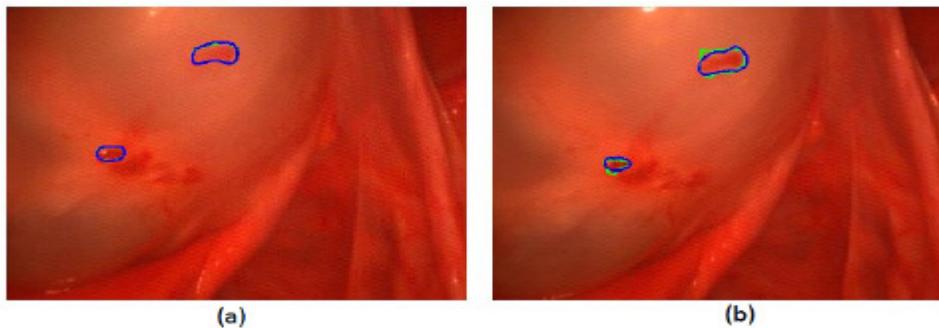


Fig. 6. Results of tracking multiple user-defined ROI on the endoscopy stream. The blue marker indicates the output of the multi-resolution approach and the green marker depicts the single-resolution approach. (a) Initial frame in the sequence (b) Result of tracking on the 54<sup>th</sup> frame in the sequence.

## 6. CONCLUSIONS & FUTURE WORK

In this paper, we demonstrated the use of pyramidal implementation for estimating the displacement field based on Thirion's demon. The algorithm successfully tracks multiple ROI's that undergo non-rigid transformation. Based on the experiments carried out it clearly indicates the advantage of pyramidal approach in terms of improvement in the convergence speed. Moreover, the result obtained using pyramidal approach generated robust and artifact free results as compared to the single resolution implementation.

Though the current approach is well suited for tracking multiple ROI's it is limited to cases of occlusion.

Future work will include parallelism of the current implementation using GPU to achieve even faster convergence. Apart from vision based assistance to MIS, the demon based tracking approach can also be applied for pathological cell detection and tracking. A classical clinical condition where cells deform rapidly over time and requires to be tracked for continuous analysis.

## REFERENCES

- [1] S.S. Beauchemin and J.L. Barron. The computation of optical flow. *ACM Computing Surveys*, 1995.
- [2] P. Anandan. A computational framework and an algorithm for the measurement of visual. Technical report, University of Massachusetts Amherst, 1987.
- [3] Jean-Yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. Technical report, Intel Corporation Microprocessor Research Labs, 2000.
- [4] Pascal Cachier, Xavier Pennec, and Nicholas Ayache. Fast non rigid matching by gradient descent: Study and improvements of the "demons" algo- rithm. Technical report, INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE, June 1999.
- [5] Visvanathan Ramesh, Dorin Comaniciu and Peter Meer. Kernel-based object tracking. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 25, NO. 5, 2003.
- [6] Abhinav Kumar, B Madhusudan Rao, Rajesh Ghole, Amol Patil, and Nilesh Ghatpande. Demons based tracking for non-rigid transformed region of interest. In *TENCON 2011 - 2011 IEEE Region 10 Conference, IEEEExplore*, November 2011.
- [7] J.-P. Thirion. Non-rigid matching using demons. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR '96*, 1996.
- [8] He Wang, Lei Dong, Jennifer O'Daniel, Radhe Mohan, Adam S Garden, K Kian Ang, Deborah A Kuban, Mark Bonnen, Joe Y Chang, and Rex Cheung. Validation of an accelerated 'demons' algorithm for deformable image registration in radiation therapy. *Physics in Medicine and Biology*, 50(12), 2005.
- [9] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, April 1983.