**The Open University of Israel**

**Department of Mathematics and Computer Science**

**Multi-scale Geometric Analysis (MGA) Algorithms for effective analysis of curves with application to Tracking Moving Dim Target**

Thesis submitted as partial fulfillment of the requirements towards an M.Sc. degree in Computer Science

The Open University of Israel

Computer Science Division

By

**Galit Volpert**

Prepared under the supervision of Dr. Ofer Levi

October 2019

**Acknowledgements**

I wish to thank my thesis supervisor, Dr. Ofer Levi, for his invaluable assistance, attention to details and helpful remarks through the thesis. His continuous support enriched me

knowledge and made this thesis possible.

I wish also to thank my family, for their encouragements, and patience during my work on this thesis.

**Contents**

[1 Introduction 6](#_Toc532407491)

[2 Overview of Related Topics 7](#_Toc532407492)

[2.1 The Radon and X-Ray Transforms 7](#_Toc532407493)

[2.2 Beamlet Analysis 7](#_Toc532407494)

[2.3 SHAS 7](#_Toc532407495)

[2.4 The Multiframe Dim-Target Detection Problem 7](#_Toc532407496)

[3 Dynamic SHAS 7](#_Toc532407497)

[4 Discussion: Fast Slant Stack vs. SHAS 7](#_Toc532407498)

[5 Experiments 8](#_Toc532407499)

[5.1 Experiments with 3D SHAS 8](#_Toc532407500)

[5.2 Experiments with 3D dynamic SHAS 21](#_Toc532407501)

[5.3 Implementation and run-time 23](#_Toc532407502)

[5.4 Qualitative results 23](#_Toc532407503)

[6 Conclusions 24](#_Toc532407504)

**Bibliography 28**

**List of Figures**

[– Example of a 3D cube with straight 3D line defined according to the Slope-Intercept System with a center aligned to Cube center. 8](#_Toc532407518)

[- Example of a 3D cube with curved 3D line composed of two different halves of 3D straights lines with a center aligned to Cube center. לא נמצאו ערכי רשימת איורים.9](#_Toc532407519)

[- Example of a 3D cube of side size 8 with maximum curved 3D line with a center aligned to Cube center. 9](#_Toc532407520)

[- Example of a 3D cube of side size 16 with maximum curved 3D line with a center aligned to Cube center 9](#_Toc532407521)

[- histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D straight lines aligned to cube center (a) cube with 4 side size (b) cube with 8 side size (c) cube with 16 side size 11](#_Toc532407522)

[- histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D curve lines aligned to cube center (a) cube with 4 side size (b) cube with 8 side size (c) cube with 16 side size 12](#_Toc532407523)

[- histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D maximum curve lines aligned to cube center. (a) cube with 8 side size (b) cube with 16 side size 12](#_Toc532407524)

[– histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing gaussian noise. (a) cube with 4 side size (b) cube with 8 side size(c) cube with 16 side size 12](#_Toc532407525)

[- histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D straight lines aligned to cube center and additive gaussian noise. (a) cube with 4 side size, SNR 1 (b) cube with 4 side size, SNR 4 (c) cube with 8 side size, SNR 1 (d) cube with 8 side size, SNR 4 (e) cube with 16 side size, SNR 1 (f) cube with 16 side size, SNR 4 13](#_Toc532407526)

[- histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D curved lines aligned to cube center and additive gaussian noise. (a) cube with 4 side size, SNR 1 (b) cube with 4 side size, SNR 4 (c) cube with 8 side size, SNR 1 (d) cube with 8 side size, SNR 4 (e) cube with 16 side size, SNR 1 (f) cube with 16 side size, SNR 4 14](#_Toc532407527)

[- histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D max. curved lines aligned to cube center and additive gaussian noise. (a) cube with 8 side size, SNR 1 (b) cube with 8 side size, SNR 4 (c) cube with 16 side size, SNR 1 (d) cube with 16 side size, SNR 4. 14](#_Toc532407528)

[– Graphs showing the outcome of the paradigm to find the multiply factor of std for each SNR (noise level) – for not normalized maximum coefficients 16](#_Toc532407529)

[- Graphs showing the outcome of the paradigm to find the multiply factor of std for each SNR (noise level) – for normalized by square root of the line length maximum coefficients 16](#_Toc532407530)

[- Number of lines Vs. percentage of predictions of max coefficient of a line aligned to cube center - for 3D Cube with 3D Straight lines no coefficients normalization 19](#_Toc532407531)

[- Number of lines Vs. percentage of predictions of max coefficient of a line aligned to cube center - for 3D Cube with 3D Straight lines coefficients normalization by square root of line length 19](#_Toc532407532)

[- Number of lines Vs. percentage of predictions of max coefficient of a line aligned to cube center alignment - for 3D Cube with 3D Straight lines coefficients normalization by line length 20](#_Toc532407533)

[- trajectory of no velocity no acceleration 21](#_Toc532407534)

[- trajectory of velocity of 0.33 pixel in y, velocity of -0.25 pixel in z no acceleration 21](#_Toc532407535)

[– trajectory of maximum velocity of 1 pixel in y, maximum velocity of -1 pixel in z no acceleration 21](#_Toc532407536)

[- trajectory of velocity from 1 pixel and acceleration decreasing logarithmic from -0.603 to -0.6145 pixel in y, velocity from -1 pixel and acceleration increasing logarithmic from 0.603 to 0.6145 pixel in z 22](#_Toc532407537)

[- trajectory of velocity from 1 pixel and acceleration decreasing in steps of -0.00006 pixel from 0 in y, velocity from -1 pixel and acceleration increasing from 0.00006 from 0 pixel in z 22](#_Toc532407538)

[- trajectory of velocity from 1 pixel and acceleration changing every 8 steps in a loop [-0.25 0 0.25 0] pixel in y, velocity from -1 pixel and acceleration changing every 8 steps in a loop [0.25 0 -0.25 0] pixel in z 22](#_Toc532407539)

[- trajectory of velocity from 1 pixel and acceleration repeating every 64 steps where in the first 48 steps no acceleration in the last 16 step 8 of -0.25 and 8 of 0.25 pixel in y, velocity from -1 pixel and acceleration repeating every 64 steps where in the first 48 steps no acceleration in the last 16 steps 8 of 0.25 and 8 of -0.25 pixel in z 22](#_Toc532407540)

**List of Tables**

### Abstract

The multiscale geometric analyses area[[1]](#footnote-1) and specifically its Beamlets analysis part lead by Professor D. Donoho shown to be useful in detection faint filaments and smoothed curves within 2D/3D images. In the work of Dr. Ofer Levi and Boris Aharon Efraty an efficient algorithm for the Beamlet transform named SHAS and its sliding window version named Dynamic SHAS were developed. In this work we present the Dynamic Shas algorithm with emphasize on its 3D version showing that it can be used for tracking maneuvering dim target as such target trajectory can be seen as faint filament in 3D images. The challenge in designing a system to track a maneuvering target trajectory is the submerging of the target within the strong noise scene. Other (Previous) methods often considered….. Recently, others have considered….. Our work is motivated by the observation that a curve can be seen as a piecewise of straight segments and the proven ability of Beamlet framework to describe and manipulate such curves. We make the following contributions. (i) We present compression between Beamlet transform algorithms Fast Slant Stack, Two Scale Recursion and the SHAS (ii) We present the Dynamic SHAS as reduced complexity algorithm compatible to running SHAS in a sliding window. Doing so we present efficient way to detect and follow faint filament anywhere in 3D image. Finally, (iii), we provide a tool based on Dynamic SHAS to track maneuvering dim target. We provide extensive qualitative and quantitative results, demonstrating that Dynamic SHAS is eligible Beamlet transform algorithm that can be used to track maneuvering dim target in a low Signal-to-Noise (SNR), with reduced complexity in compere to the one demanded by existing methods.

# Introduction

[[2]](#footnote-2)When the distance between optical imaging system and man-made flying vehicles satellites and IR sensors is usually more than 30,000 km that the target on a EO sensor is a small blob with only several pixels. Meanwhile, the energy of the object decays greatly for long distance propagation, and it is usually submerged in noise and clutter. This causes great difficulty for infrared dim small target detection and tracking [[1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118372/#b1-sensors-14-09451),[2](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118372/#b2-sensors-14-09451)]. The problem of how to effectively distinguish dim small targets from clutter has been widely studied over the past years, and a number of dim target detection algorithms have been developed and they can approximately be classified into two categories, namely, detection before track (DBT) and track before detection (TBD) [[3](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118372/#b3-sensors-14-09451)–[5](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118372/#b5-sensors-14-09451)]. Image filtering and content learning are the two basic methods of DBT-based target detection algorithms. The image filtering-based detection algorithms such as Top-Hat [[6](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118372/#b6-sensors-14-09451)], TDLMS [[7](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118372/#b7-sensors-14-09451)] and wavelet transform [[8](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118372/#b8-sensors-14-09451),[9](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118372/#b9-sensors-14-09451)] usually whiten the image signal, and then determine whether there is a target or not in every scan by the amplitude threshold using some criteria, such as constant false alarm ratio (CFAR). The content learning-based target detection algorithms such as principal component analysis (PCA) compare the similarity of the image and the template pre-learned by knowledge [[10](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118372/#b10-sensors-14-09451)]. The TBD-based algorithms jointly process more consecutive scans and declare the presence of a target and its track by searching the candidate trajectory using an exhaustive hypothesis. Temporal cross product (TCP) is presented to extract the characteristics of temporal pixels by using a temporal profile in infrared image sequences, and it could effectively enhance the **signal-to-clutter ratio** (SCR) [[11](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118372/#b11-sensors-14-09451)] Defining the **signal**‐to‐**clutter ratio** as the **ratio** of the peak **signal** to the rms value of the **clutter** . Higher detection probability and lower false alarm probability in every scan could not only facilitate analysis of characteristics, including movement analysis, but also simplify the computational complexity for TBD.

# Overview of Related Topics

## The Radon and X-Ray Transforms

## Beamlet Analysis

## SHAS

## The Multiframe Dim-Target Detection Problem

# Dynamic SHAS

# Discussion: Fast Slant Stack vs. SHAS

# Experiments

## Experiments with 3D SHAS

The experiments we conducted with 3D SHAS transform divided to two major subjects:

1. **Detection criteria** - Assessing the ability to differentiate between 3D cube with a X driven line and additive gaussian noise to 3D cube with noise.
2. **Tracking criteria** - Assessing the ability to estimate the center of X-driven line in a 3D cube with a X-driven line and additive gaussian noise.

The experiments are based on:

1. 3D cubes of side size - 4,8,16 (pixels)
2. 3D cube with 3D Lines –
   1. Line center aligned to the cube center by offset [0, quarter of cube side size] – as the cube side size and line length are even the center of the line is defined as its half plus one and the cube center is defined as half the side size plus one.
   2. Types:
      1. Straight – Defined in section (Beamlets)

תמונה שמכילה טקסט, מפה

התיאור נוצר באופן אוטומטי

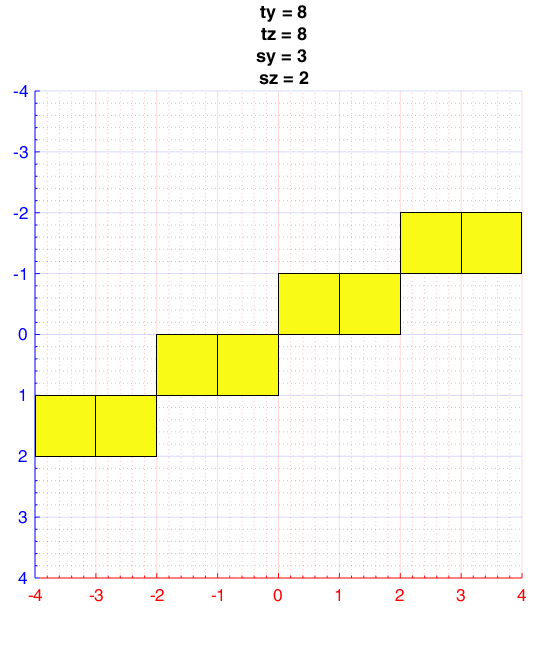
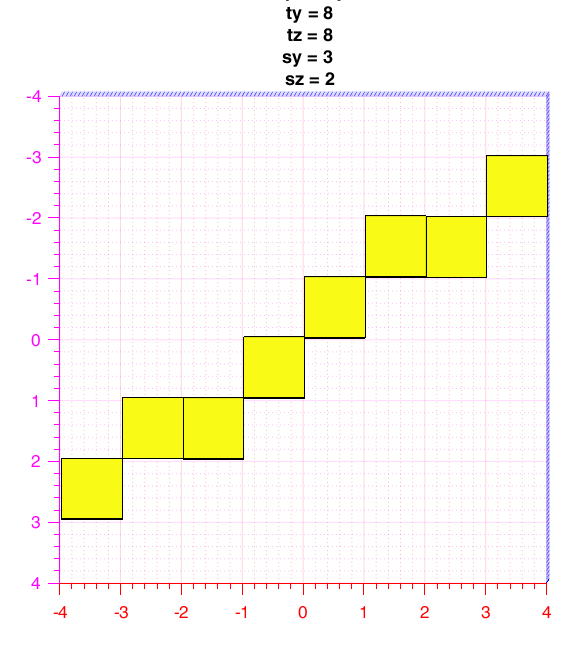
 

Figure 1 – Example of a 3D cube with straight 3D line defined according to the Slope-Intercept System with a center aligned to Cube center.

* + 1. Curve – Composed of two half's each belong to different half of straight line(a)

תמונה שמכילה טקסט, מפה

התיאור נוצר באופן אוטומטי תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Figure 2 - Example of a 3D cube with curved 3D line composed of two different halves of 3D straights lines with a center aligned to Cube center.

* + 1. Maximum Curve – Created by taking the first cube slice with a patch of pixels and calculating its position in the next slice using the kinematic formulas based on maximum velocity of pixel and acceleration of pixel.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי תמונה שמכילה טקסט

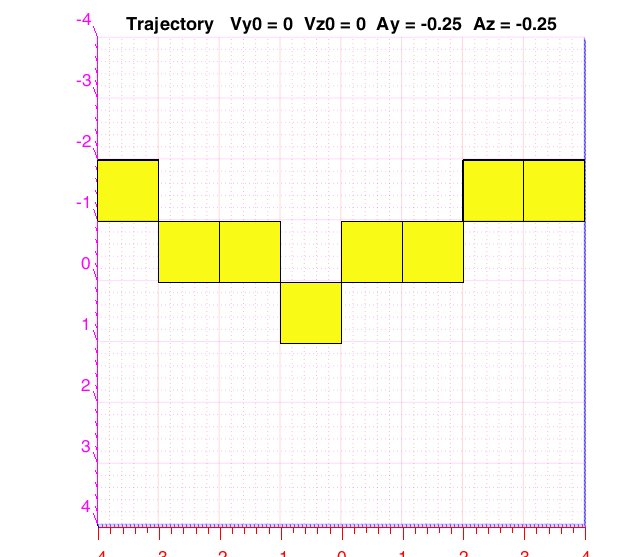
התיאור נוצר באופן אוטומטי 

Figure 3 - Example of a 3D cube of side size 8 with maximum curved 3D line with a center aligned to Cube center.

תמונה שמכילה טקסט, מפה

התיאור נוצר באופן אוטומטי תמונה שמכילה טקסט

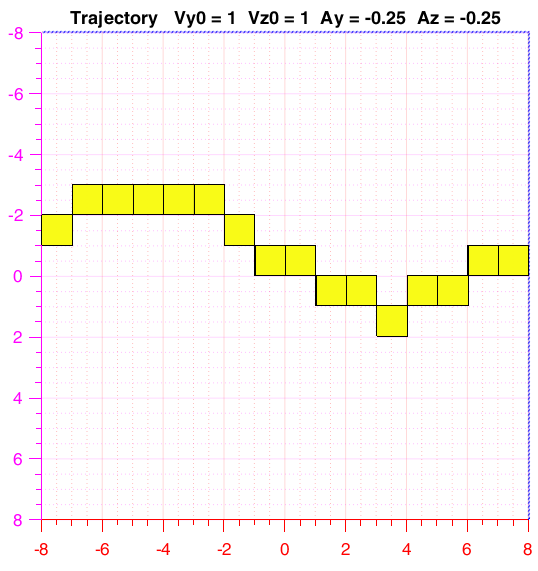
התיאור נוצר באופן אוטומטי 

Figure 4 - Example of a 3D cube of side size 16 with maximum curved 3D line with a center aligned to Cube center

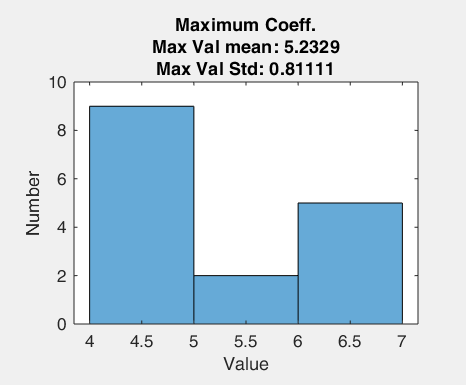
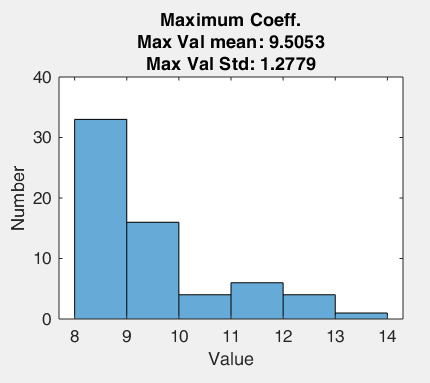
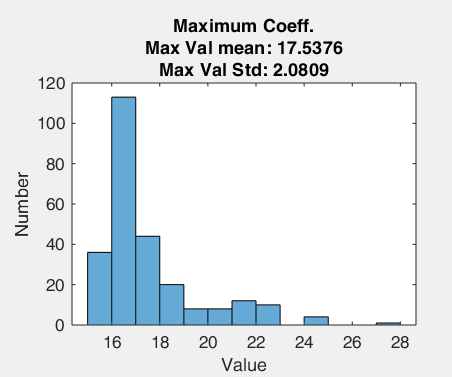
1. 3D cube with 3D line and additive gaussian noise in level -

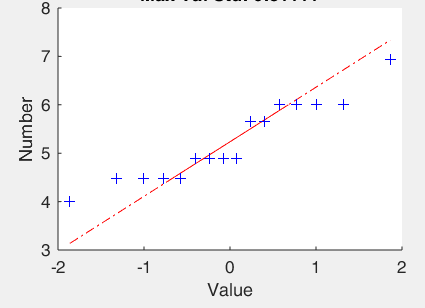
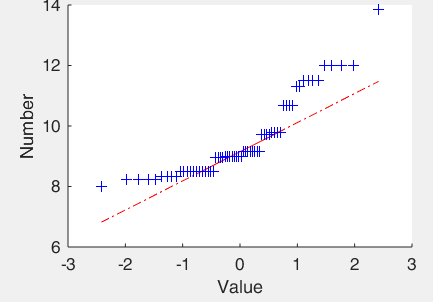
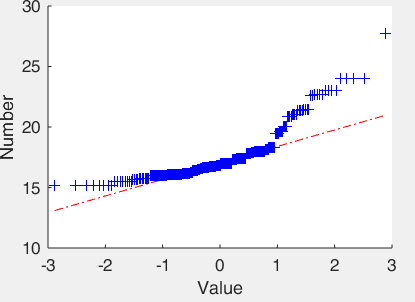
**Detection criteria –**

The evaluation of the data in those experiments is done using two groups:

1. Maximum Coefficients of transform on 3D cube with gaussian noise.
2. Maximum Coefficients of transform on 3D cube with gaussian noise and a line aligned to center of the cube by offsets [0, quarter of cube side size].
3. 3D cube with lines aligned to the cube center (offset 0).

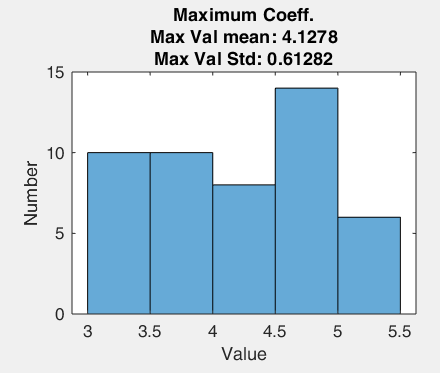
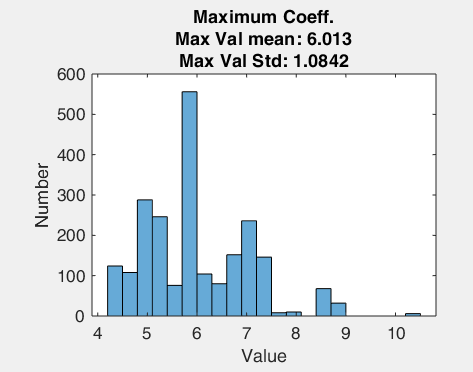
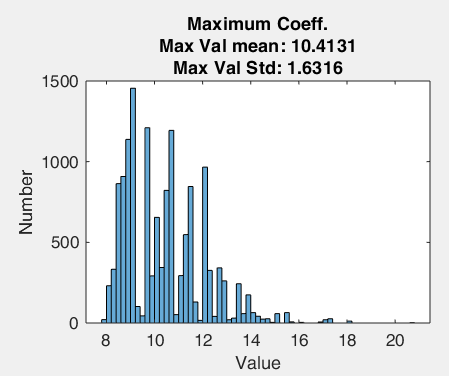
As can be seen in figure … the distribution of the maximum coefficient for cube with a line aligned to center is not normal.

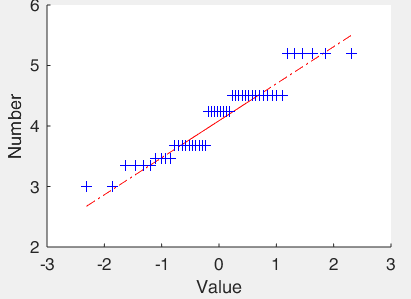
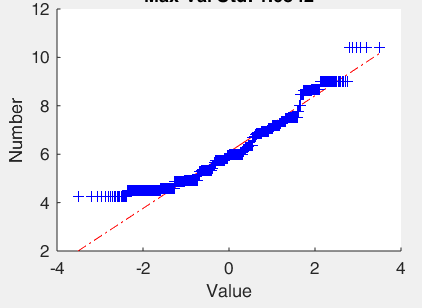
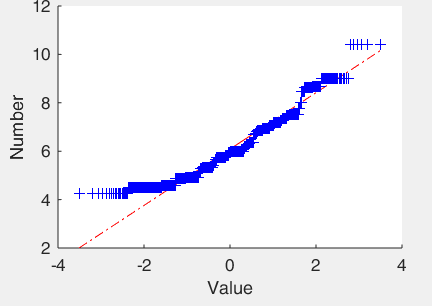
  

1. (b) (c)

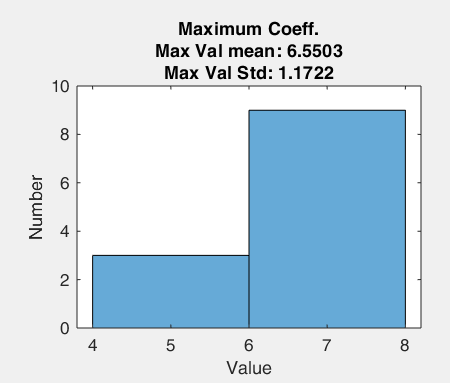
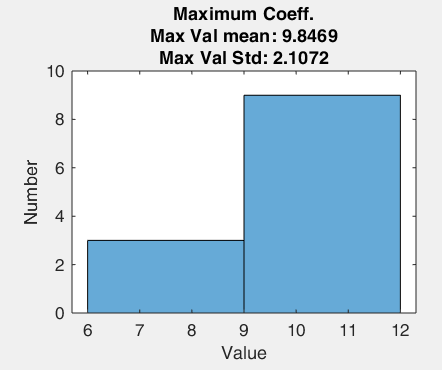
Figure 5- histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D straight lines aligned to cube center (a) cube with 4 side size (b) cube with 8 side size (c) cube with 16 side size

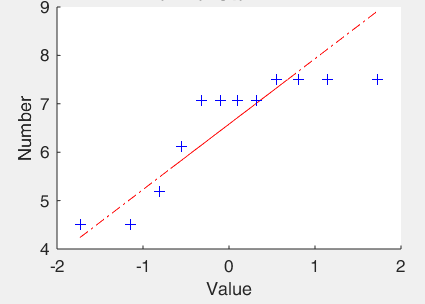
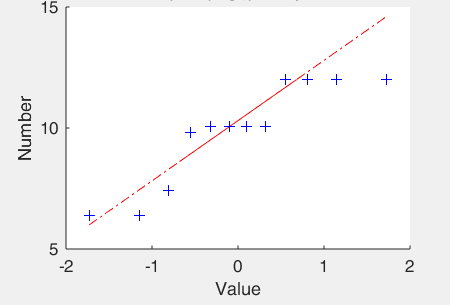
  

1. (b) (c)

Figure 6 - histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D curve lines aligned to cube center (a) cube with 4 side size (b) cube with 8 side size (c) cube with 16 side size

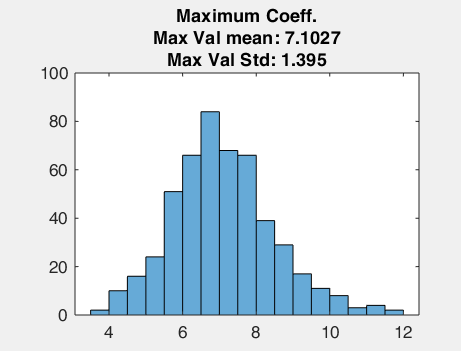
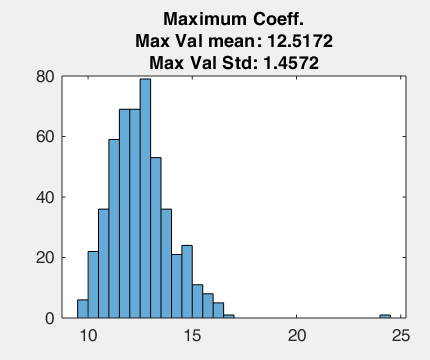
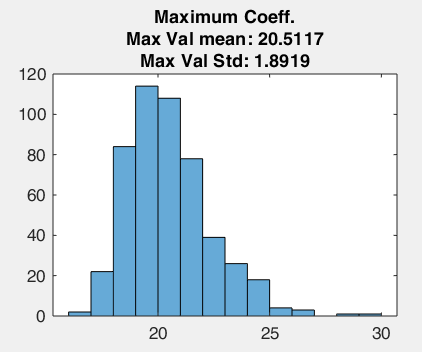
 

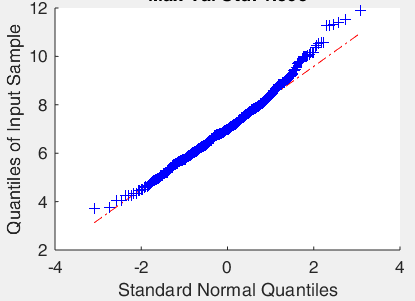
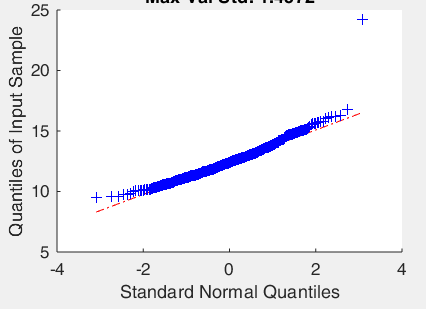
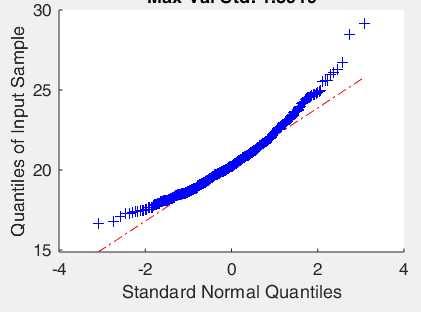
 

1. (b)

Figure 7 - histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D maximum curve lines aligned to cube center. (a) cube with 8 side size (b) cube with 16 side size

As seen in figures … the distribution of the maximum coefficients for cubes with gaussian noise is not normal.

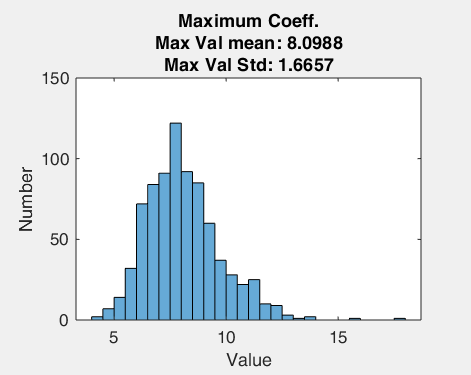
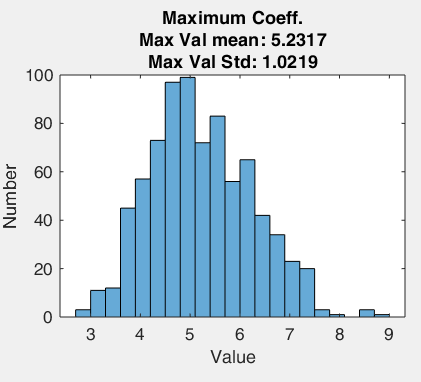
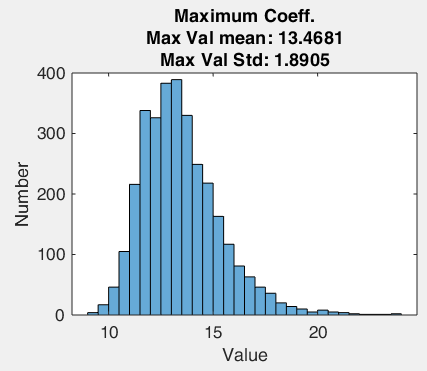
  

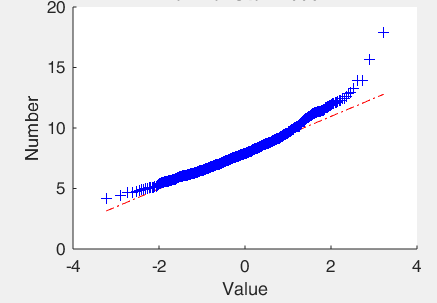
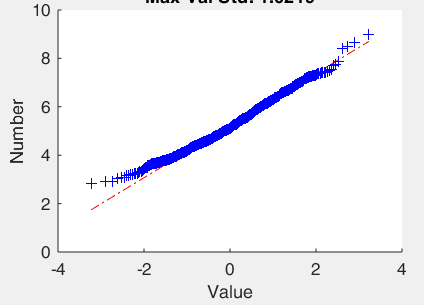
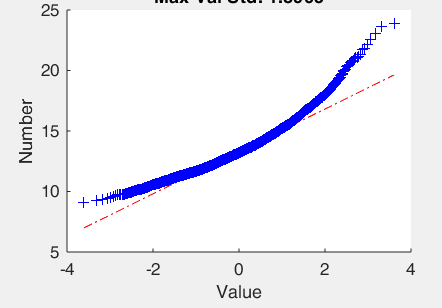
  

1. (b) (c)

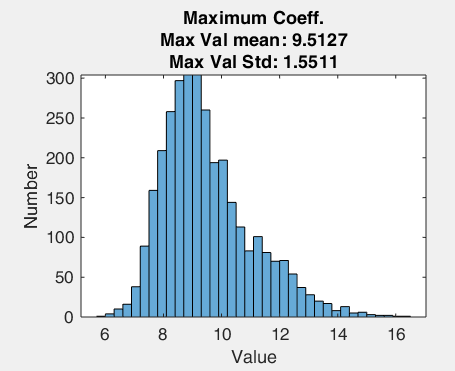
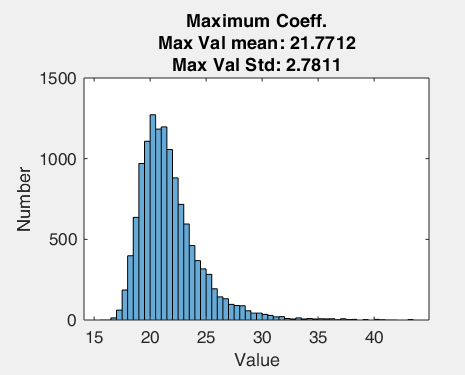
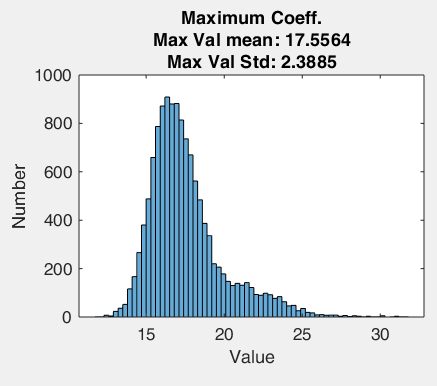
Figure 8 – histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing gaussian noise. (a) cube with 4 side size (b) cube with 8 side size(c) cube with 16 side size

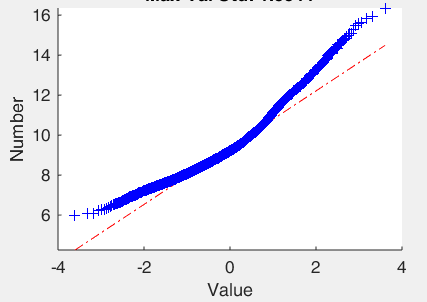
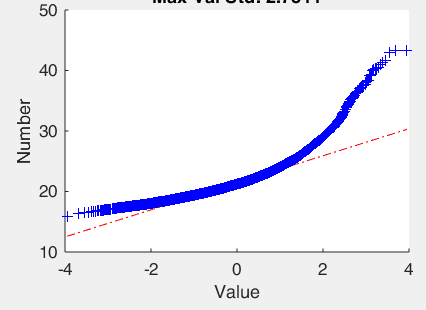
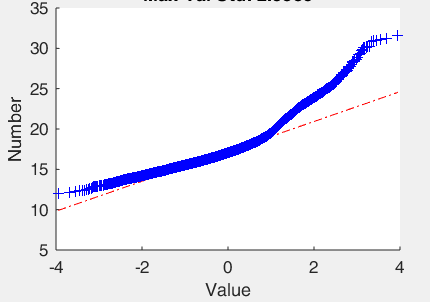
As seen in figures the distribution of the maximum coefficients for cubes with a line and gaussian noise is not normal.

**  **

**  **

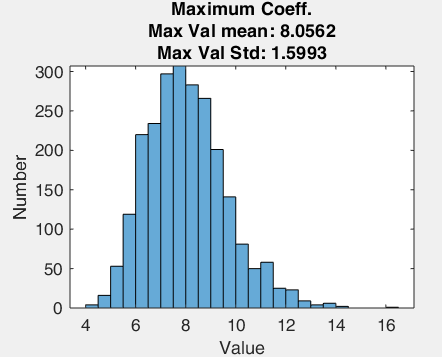
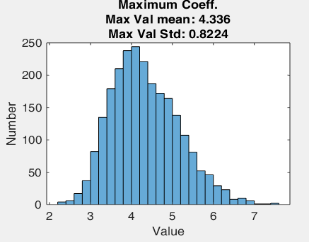
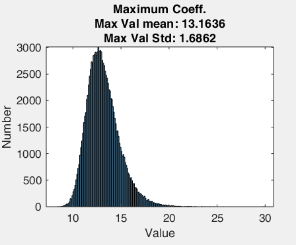
1. **(b) (c)**

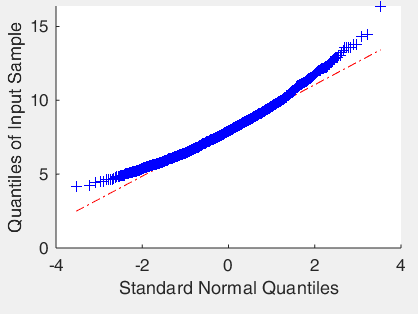
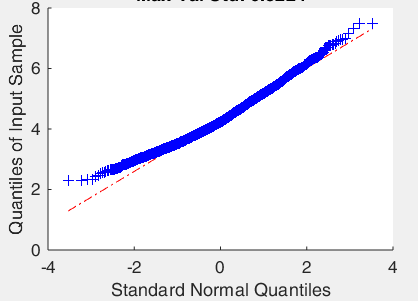
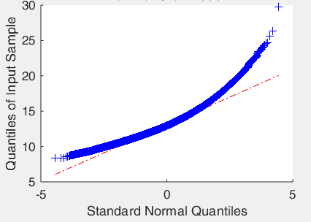
**  **

**  **

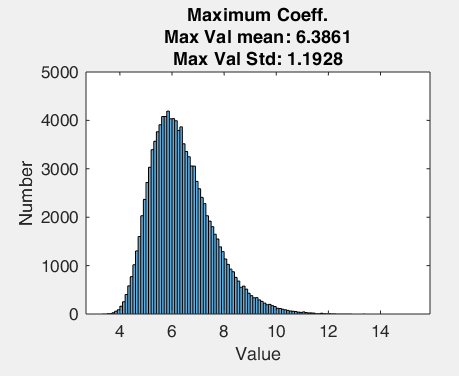
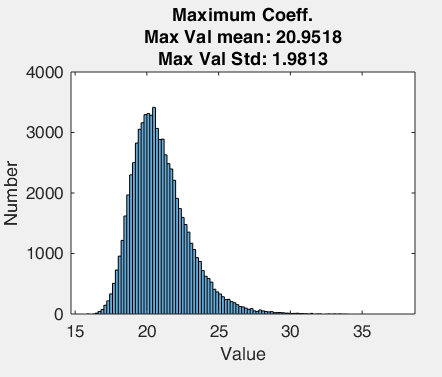
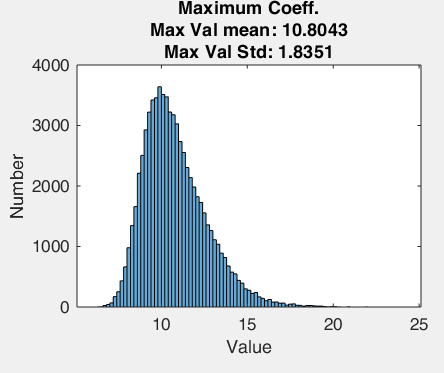
**(d) (e) (f)**

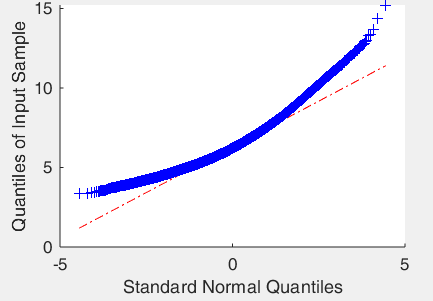
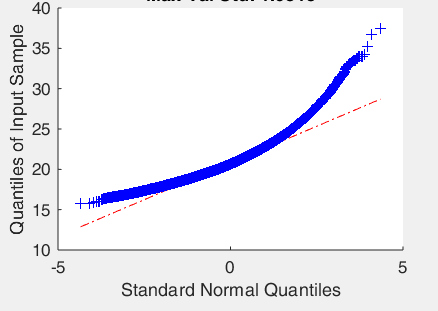
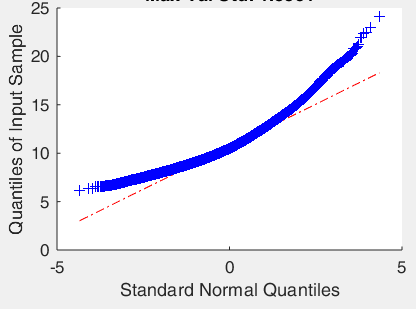
Figure 9- histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D straight lines aligned to cube center and additive gaussian noise. (a) cube with 4 side size, SNR 1 (b) cube with 4 side size, SNR 4 (c) cube with 8 side size, SNR 1 (d) cube with 8 side size, SNR 4 (e) cube with 16 side size, SNR 1 (f) cube with 16 side size, SNR 4

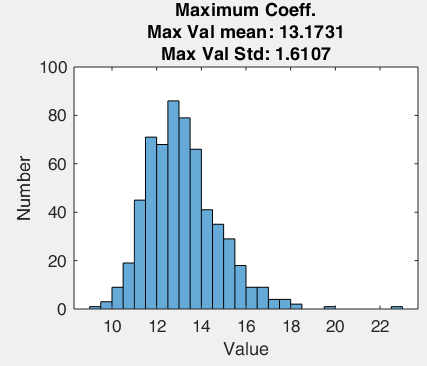
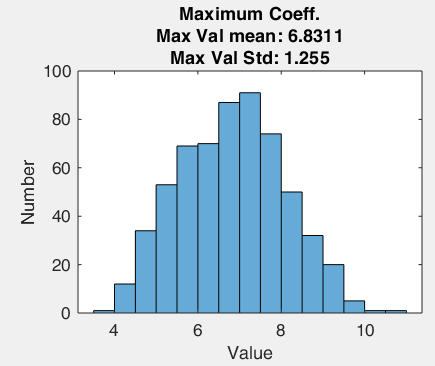
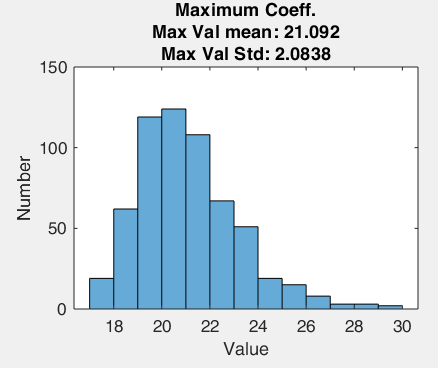
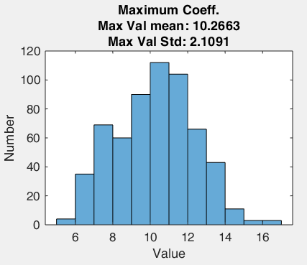
1. (b) (c)

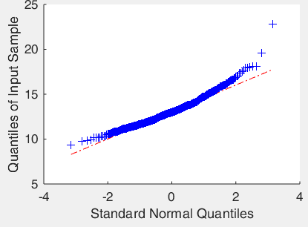
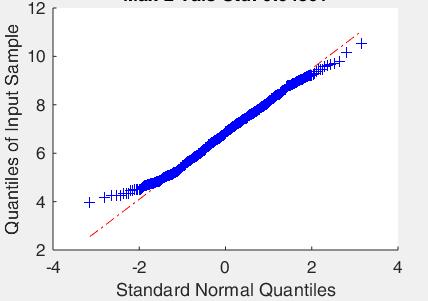
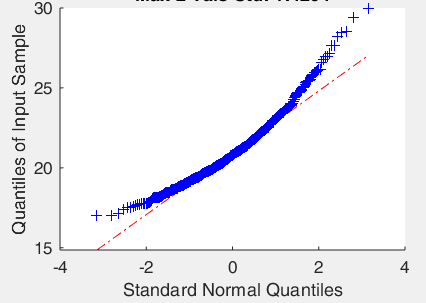
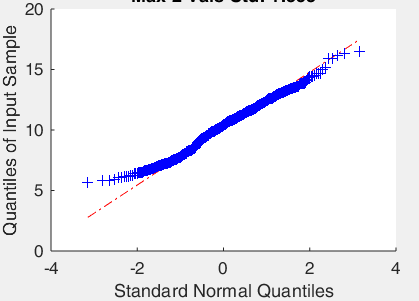
**  **

***  ***

***(d) (e) (f)***

Figure 10 - histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D curved lines aligned to cube center and additive gaussian noise. (a) cube with 4 side size, SNR 1 (b) cube with 4 side size, SNR 4 (c) cube with 8 side size, SNR 1 (d) cube with 8 side size, SNR 4 (e) cube with 16 side size, SNR 1 (f) cube with 16 side size, SNR 4

**   **

1. (b) (c) (d)

Figure 11 - histogram and quantile-quantile plots of maximum coefficients distribution for transform on 3D cube containing different 3D max. curved lines aligned to cube center and additive gaussian noise. (a) cube with 8 side size, SNR 1 (b) cube with 8 side size, SNR 4 (c) cube with 16 side size, SNR 1 (d) cube with 16 side size, SNR 4.

As seen above the maximum coefficient's distribution is not normal. So, we used the next paradigm to estimate the statistics of a maximum coefficients representing a 3D cube without a line.

.

related SNR

Algorithm 1 – Estimating the P value for a maximum coefficient representing a 3D cube without a line for specific noise level.

The 'multiple Factor' is used to estimate the P value (for 'multiple Factor' of 2 – p value equal 97.72%) - for coefficient representing a 3D cube without a line (for specific noise level).

We performed the above paradigm for:

1. Maximum coefficients
2. Second local maximum coefficients found by ('imregionalmax')
3. Maximum of coefficients difference (Coeff[sy, sz, ty, tz] - Coeff[sy, sz, ty+1, tz+1]) this one can be used if the line in the image is more than one pixel thick.

תמונה שמכילה טקסט, מפה

התיאור נוצר באופן אוטומטי

Figure 12 – Graphs showing the outcome of the paradigm to find the multiply factor of std for each SNR (noise level) – for not normalized maximum coefficients

תמונה שמכילה טקסט, מפה

התיאור נוצר באופן אוטומטי

Figure 13 - Graphs showing the outcome of the paradigm to find the multiply factor of std for each SNR (noise level) – for normalized by square root of the line length maximum coefficients

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| 2.7 | 2.4 | 2.1 | 1.7 | 1.2 | 0.4 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| 2.9 | 2.6 | 2.2 | 1.7 | 1.1 | 0.1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| >3 | >3 | 2.7 | 2.3 | 1.7 | 0.8 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| >3 | >3 | 2.9 | 2.4 | 1.7 | 0.7 |

1. Straight ,8 (b) Straight , 8 , Norm. Sqrt

(c) Straight ,16 (d) Straight , 16 , Norm. Sqrt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| 1.9 | 1.6 | 1.1 | 0.6 | <0.6 | <0.6 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| 1.8 | 1.4 | 0.8 | 0.2 | <0.2 | <0.2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| 2. | 2.0 | 1.5 | 0.9 | <0.9 | <0.9 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| 2.4 | 2.0 | 1.4 | 0.6 | <0.6 | <0.6 |

(e) Curved ,8 (f) Curved , 8 , Norm. Sqrt

(g) Curved, 8 (h) Curved, 16, Norm. Sqrt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| 2.2 | 1.9 | 1.4 | 0.9 | 0.1 | <0.1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| 2.2 | 1.8 | 1.3 | 0.6 | <0.6 | <0.6 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| 1.8 | 1.5 | 1.1 | 0.5 | <0.5 | <0.5 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 3.5 | 3 | 2.5 | 2 | 1.5 |
| 2.0 | 1.6 | 1.0 | 0.3 | <0.3 | <0.3 |

(i) Max Curved, 8 (j) Max Curved, 8, Norm. Sqrt

(k) Max Curved, 16 (l) Max Curved, 16, Norm. Sqrt

Table 1- SNR Vs Multiple Factor of Std -

**Tracking criteria –**

The evaluation of the data in those experiments is done using two groups:

1. Maximum Coefficients of transform on 3D cube with gaussian noise.
2. Maximum Coefficients of transform on 3D cube with gaussian noise and a line aligned to center of the cube by offsets [0, quarter of cube side size].
3. 3D cube with noise:

The experiment: For each of the cubes size we create 5000 3D cubes with randomly gaussian noise and find the percentage of getting maximum coefficient that represent a line aligned to center of the cube.

|  |  |
| --- | --- |
| cube (win) size | Prediction of Coeff. Representing line aligned to center |
| 4 | 5.38% |
| 8 | 1% |
| 16 | 0.22% |

|  |  |
| --- | --- |
| Cube (win) size | Prediction of Coeff. Representing line aligned to center |
| 4 | 10.94% |
| 8 | 4.3% |
| 16 | 1.96% |

|  |  |
| --- | --- |
| cube (win) size | Prediction of Coeff. Representing line aligned to center |
| 4 | 0.22% |
| 8 | 0% |
| 16 | 0% |

1. (b) (c)

Table 2 – percentage of predicting a coefficient that represent a line aligned to center of the cube for different cube size: (a) not normalized coefficients (b) normalized coefficients by square root of the line length (c) normalized coefficients by square root of the line length

From the above results we conclude:

1. It is preferable to normalize the coefficients to enhance the reliability that when we get a maximum coefficient representing a line aligned to center it is for 3D cube containing real line and not only noise.
2. The preferred way is to normalize the coefficients by line length.

ii) 1. 3D cube with lines aligned to the cube center

In this experiment:

For each of the lines, for each SNR, we create 50 - 3D cubes with the line and randomly additive gaussian noise in specific SNR and find the percentage of getting maximum coefficient that represent a line aligned to center of the 3D cube.

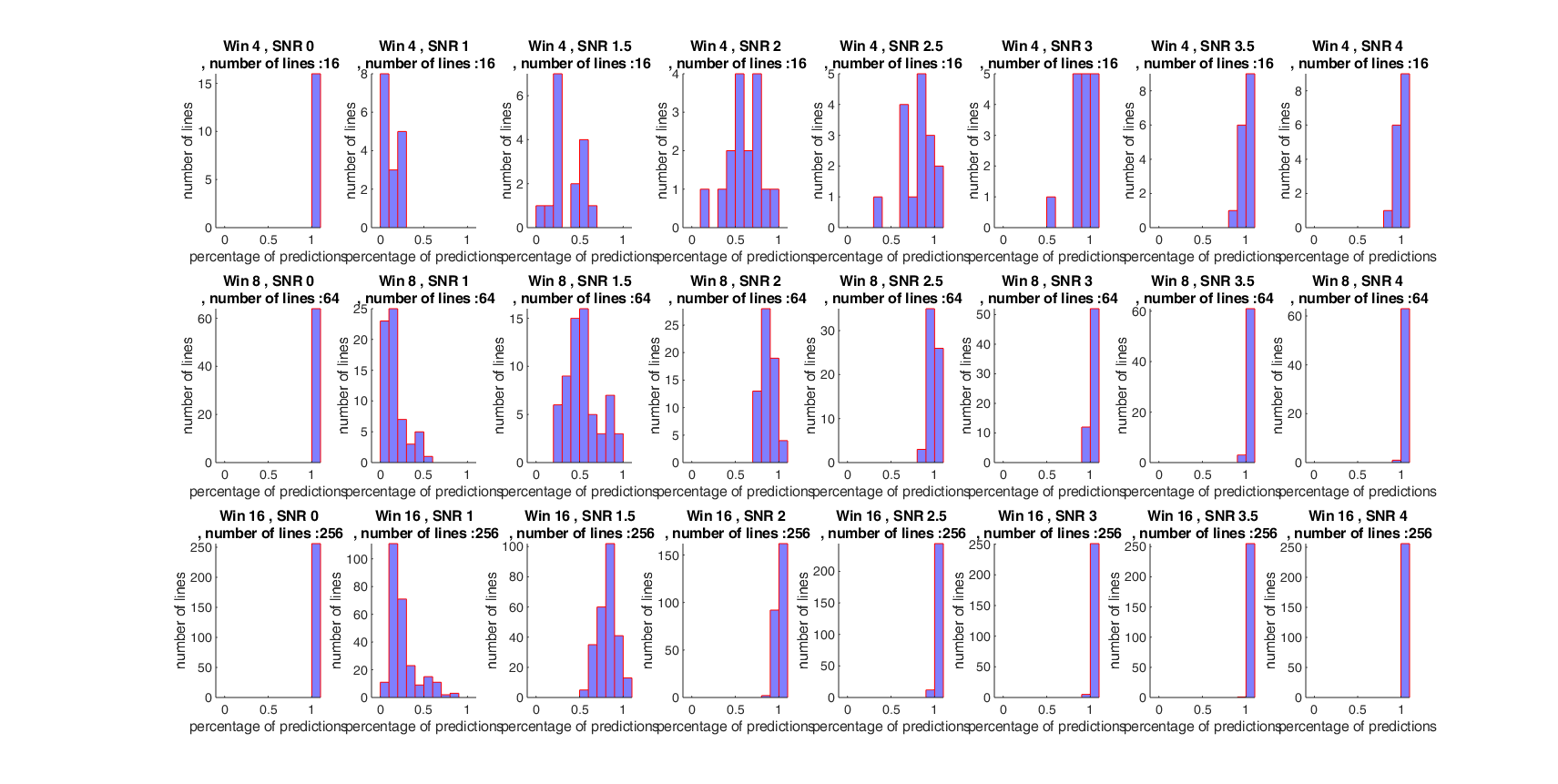


Figure 14- Number of lines Vs. percentage of predictions of max coefficient of a line aligned to cube center - for 3D Cube with 3D Straight lines no coefficients normalization

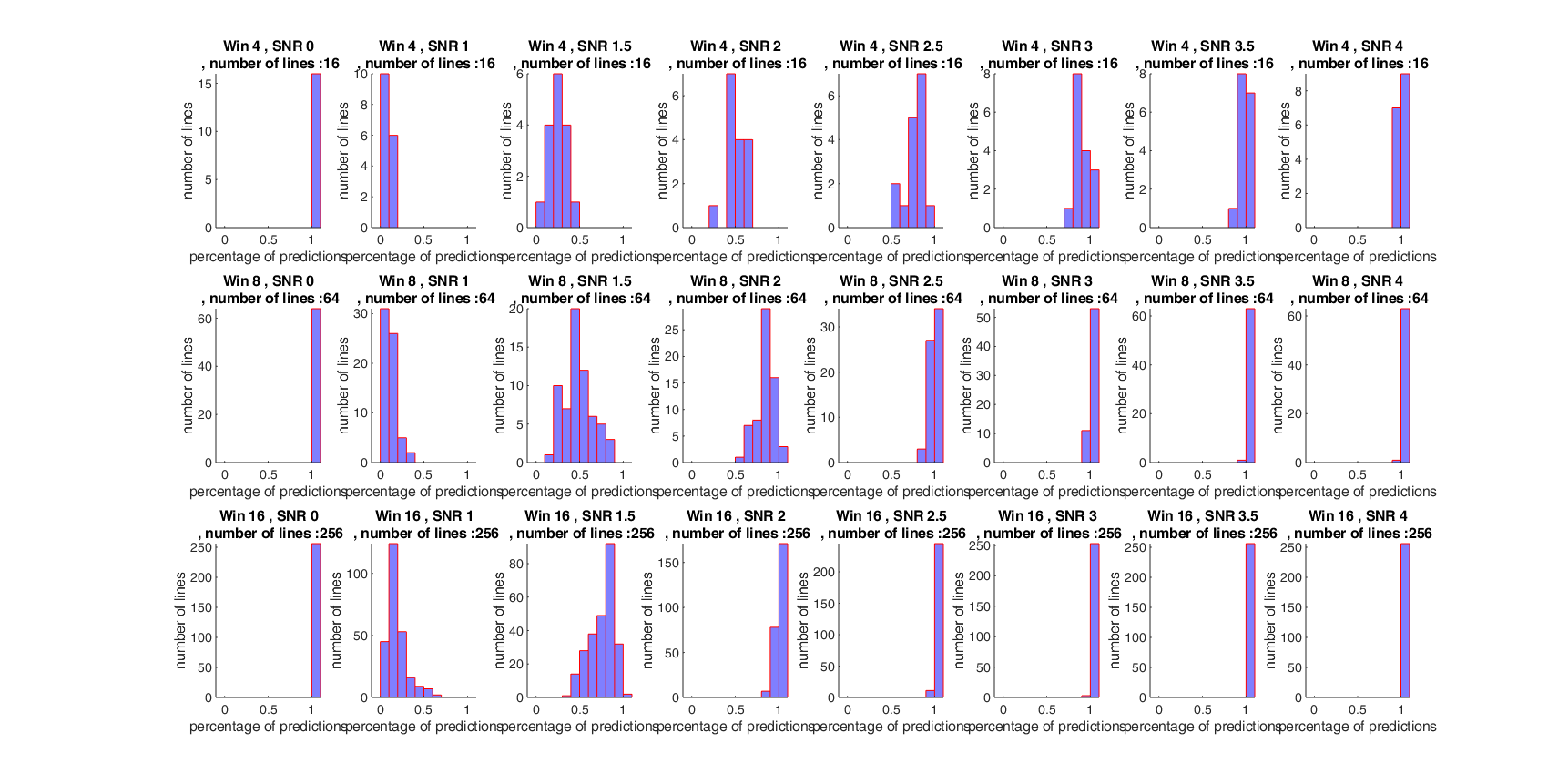


Figure 15 - Number of lines Vs. percentage of predictions of max coefficient of a line aligned to cube center - for 3D Cube with 3D Straight lines coefficients normalization by square root of line length

תמונה שמכילה צילום מסך

התיאור נוצר באופן אוטומטי

Figure 16 - Number of lines Vs. percentage of predictions of max coefficient of a line aligned to cube center alignment - for 3D Cube with 3D Straight lines coefficients normalization by line length

From the above graphs, we conclude:

1. 16 pixels cube[[3]](#footnote-3) has the best prediction (almost perfect for SNR 2.5)
2. The results for normalizing the maximum coefficients by square root of the line length has almost the same prediction performance as not normalized coefficients.
3. The results for normalizing the maximum coefficients by lines length are very low compared with not normalizing or normalizing by square root of the line length.

## Experiments with 3D dynamic SHAS

The experiments are divided to two kinds –

**Synthetic scenario** – a synthetic trajectory added with gaussian noise in different SNRs.

**Images with Clouds scenarios** – a synthetic trajectory added to real images with Clouds in different CSR.

The experiments are based on:

1. The next trajectories:

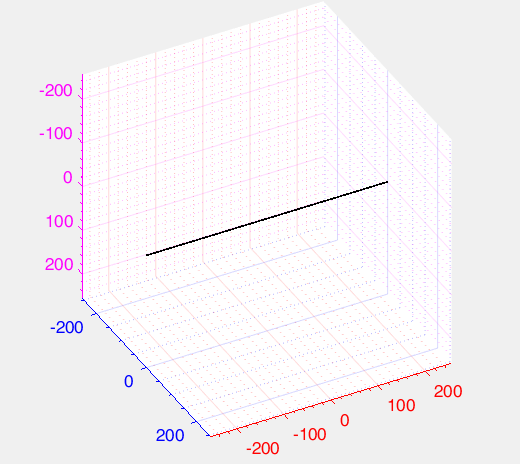
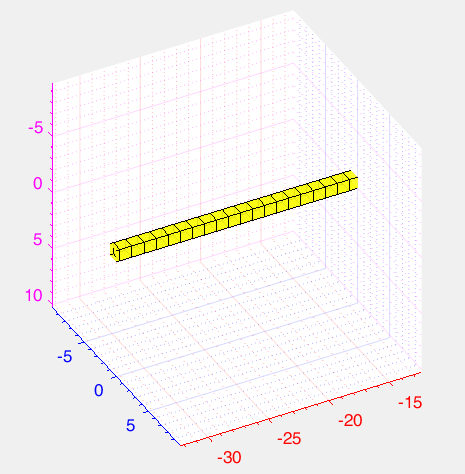
 

Figure 17 - trajectory of no velocity no acceleration

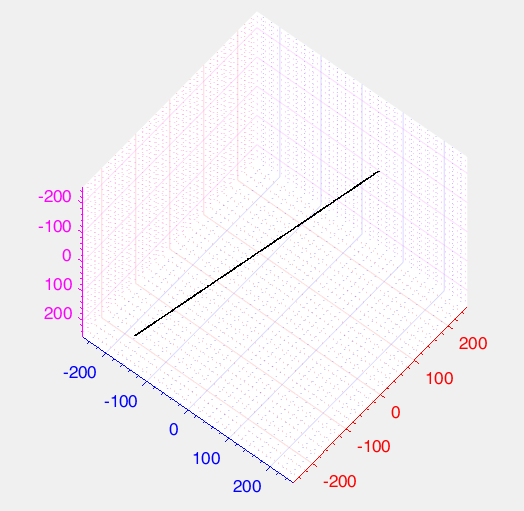
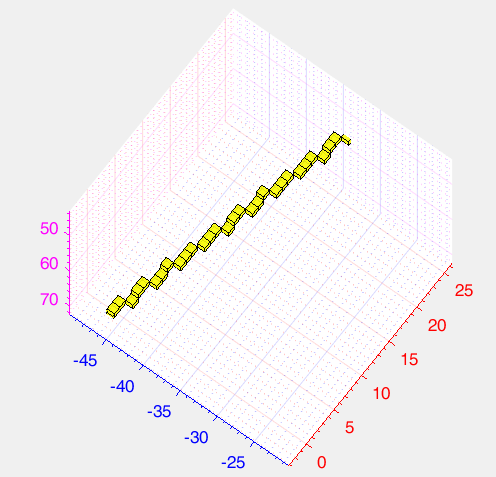
 

Figure 18 - trajectory of velocity of 0.33 pixel in y, velocity of -0.25 pixel in z no acceleration

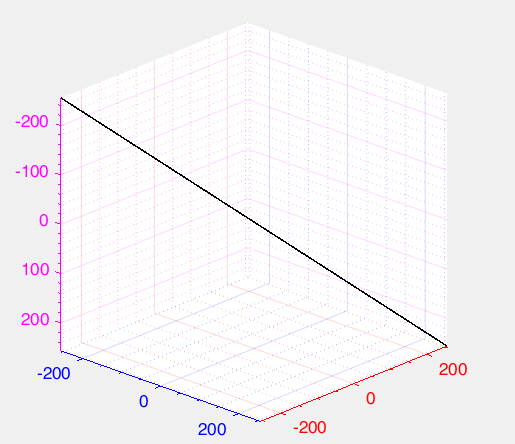
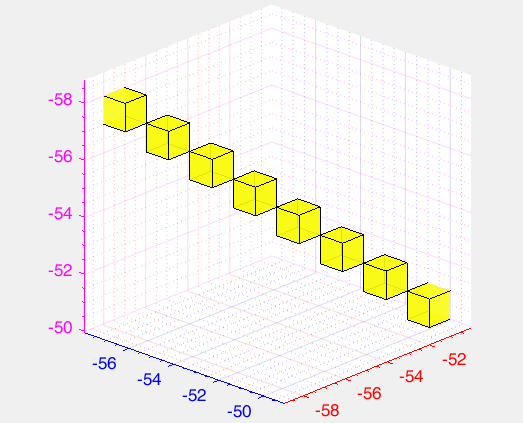
 

Figure 19 – trajectory of maximum velocity of 1 pixel in y, maximum velocity of -1 pixel in z no acceleration

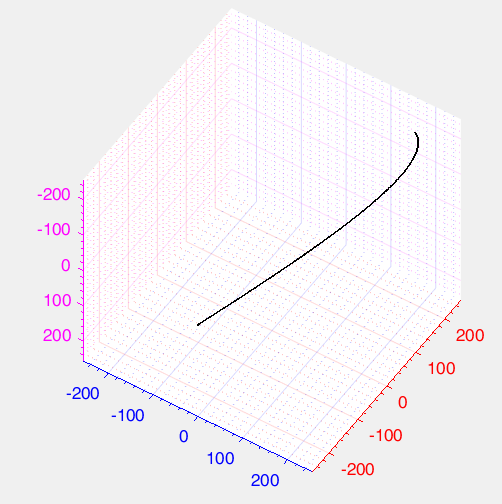
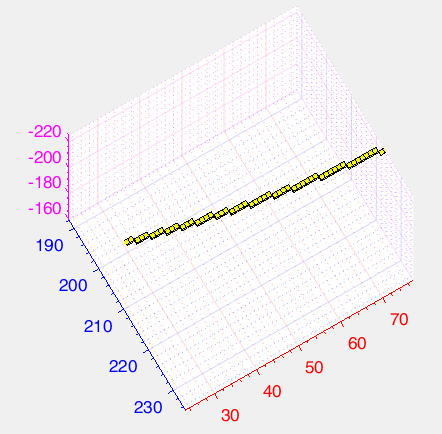
 

Figure 20 - trajectory of velocity from 1 pixel and acceleration decreasing logarithmic from -0.603 to -0.6145 pixel in y, velocity from -1 pixel and acceleration increasing logarithmic from 0.603 to 0.6145 pixel in z

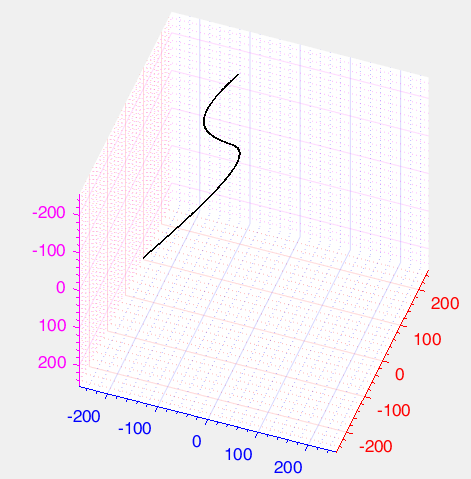
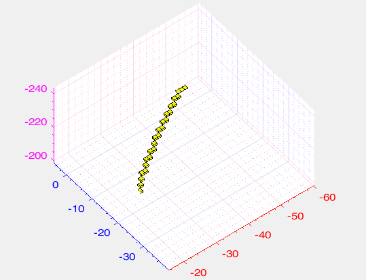
 

Figure 21 - trajectory of velocity from 1 pixel and acceleration decreasing in steps of -0.00006 pixel from 0 in y, velocity from -1 pixel and acceleration increasing from 0.00006 from 0 pixel in z

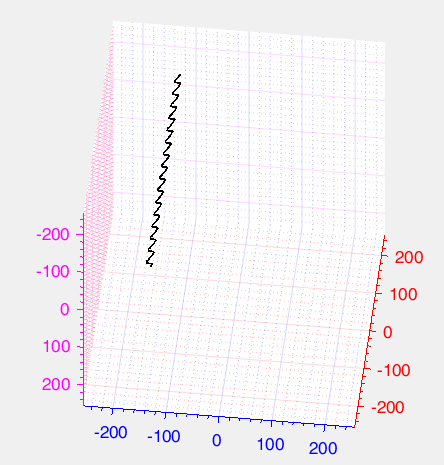
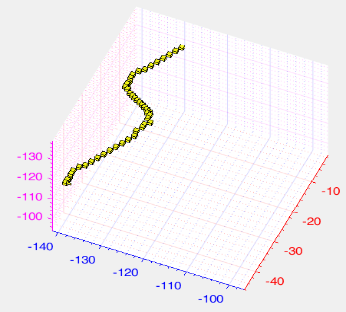
 

Figure 22 - trajectory of velocity from 1 pixel and acceleration changing every 8 steps in a loop [-0.25 0 0.25 0] pixel in y, velocity from -1 pixel and acceleration changing every 8 steps in a loop [0.25 0 -0.25 0] pixel in z

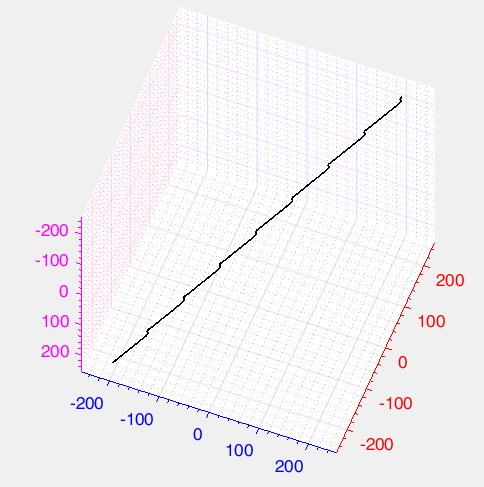
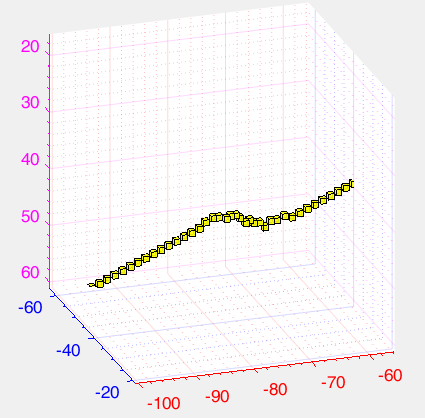
 

Figure 23- trajectory of velocity from 1 pixel and acceleration repeating every 64 steps where in the first 48 steps no acceleration in the last 16 step 8 of -0.25 and 8 of 0.25 pixel in y, velocity from -1 pixel and acceleration repeating every 64 steps where in the first 48 steps no acceleration in the last 16 steps 8 of 0.25 and 8 of -0.25 pixel in z

## Implementation and run-time

## Qualitative results

# Conclusions

**Future work.**

**References**

Both answer how far your values are spread around the mean of the observations.

An observation that is 1 under the mean is equally "far" from the mean as a value that is 1 above the mean. Hence you should neglect the sign of the deviation. This can be done in two ways:

Calculate the absolute value of the deviations and sum these.

Square the deviations and sum these squares. Due to the square, you give more weight to high deviations, and hence the sum of these squares will be different from the sum of the means.

After calculating the "sum of absolute deviations" or the "square root of the sum of squared deviations", you average them to get the "mean deviation" and the "standard deviation" respectively.

The mean deviation is rarely used.

Both measure the [dispersion](http://en.wikipedia.org/wiki/Statistical_dispersion) of your data by computing the distance of the data to its mean.

the **mean absolute deviation** is using norm L1 (it is also called [Manhattan distance or rectilinear distance](http://en.wikipedia.org/wiki/Manhattan_distance))

the **standard deviation** is using norm L2 (also called [Euclidean distance](http://en.wikipedia.org/wiki/Euclidean_distance))

<https://stats.stackexchange.com/questions/81986/mean-absolute-deviation-vs-standard-deviation>

<https://stats.stackexchange.com/questions/67337/mad-in-relation-to-95-confidence>

CNR vs SNR

[file:///D:/לימודים/computer%20vision/תזה/articls/mfd/On%20the%20Definition%20of%20Signal-To-Noise%20Ratio%20and%20Contrast-To-Noise%20Ratio%20for%20fMRI%20Data.pdf](D://לימודים/computer%20vision/תזה/articls/mfd/On%20the%20Definition%20of%20Signal-To-Noise%20Ratio%20and%20Contrast-To-Noise%20Ratio%20for%20fMRI%20Data.pdf)

As target = pixel + sigma \* K

The SNR - sigma\*k average is zero => (pixel average )/(sigma\*K)

### תקציר

### תוכן העניינים

**האוניברסיטה הפתוחה**

**המחלקה למתמטיקה ומדעי המחשב**

עבודת תזה זו הוגשה כחלק מהדרישות לקבלת תואר

"מוסמך למדעים" M.Sc. במדעי המחשב

באוניברסיטה הפתוחה

החטיבה למדעי המחשב

על-ידי

**גלית ולפרט**

העבודה הוכנה בהדרכתו של ד"ר עופר לוי

אוקטובר 2019

1. http://www.ipam.ucla.edu/programs/long-programs/multiscale-geometry-and-analysis-in-high-dimensions/ [↑](#footnote-ref-1)
2. Sparse Representation for Infrared Dim Target Detection via a Discriminative Over-Complete Dictionary Learned Online [↑](#footnote-ref-2)
3. In DSHAS the Cube is referred as the sliding window [↑](#footnote-ref-3)