Automatic Calibration and Neural Networks for Robot Guidance

Balaji Sethuramasamyraja, Masoud Ghaffari, and Ernest L. Hall

Center for Robotics Research
University of Cincinnati
Cincinnati, Oh 45221-0072

ABSTRACT

An autonomous robot must be able to sense its environment and react appropriately in a variable environment. The University of Cincinnati Robot team is actively involved in building a small, unmanned, autonomously guided vehicle for the International Ground Robotics Contest organized by Association for Unmanned Vehicle Systems International (AUVSI) each year. The unmanned vehicle is supposed to follow an obstacle course bounded by two white/yellow lines, which are four inches thick and 10 feet apart. The navigation system for one of the University of Cincinnati's designs, Bearcat, uses 2 CCD cameras and an image-tracking device for the front end processing of the image captured by the cameras. The three dimensional world co-ordinates were reduced to two dimensional image coordinates as a result of the transformations taking place from the ground plane to the image plane. A novel automatic calibration system was designed to transform the image co-ordinates back to world co-ordinates for navigation purposes. The purpose of this paper is to simplify this tedious calibration using an artificial neural network. Image processing is used to automatically detect calibration points. Then a back projection neural algorithm is used to learn the relationships between the image coordinates and three-dimensional coordinates. This transformation is the main focus of this study. Using these algorithms, the robot built with this design is able to track and follow the lines successfully.

Keywords: Calibration, robotics, neural network, robot guidance, robot vision

1. INTRODUCTION

Autonomous road vehicles, guided by computer vision systems, are a topic of intensive research in numerous places in the world. For effective functioning the robot must be capable of purposeful movement in the environment. A truly autonomous robot must sense its environment and react appropriately. These issues attain greater importance in an outdoor, variable environment. The rise in popularity of the single chip microcomputer and the drastic reductions in size and cost of integrated circuits in recent years have opened up huge new opportunities for intelligent systems and robotics. Building an unmanned ground robotic vehicle is a challenging task. A robot can be thought of as an intelligent connection of perception to action. Implementing this is a formidable task and might take on a wide variety of disciplines, ranging from mechanical logic to microprocessor control to networks of neuron-like gates. Mobile robots pose a unique challenge to artificial intelligence researchers. They are inherently autonomous and they force us to deal with key issues such as uncertainty, reliability and real time response. They also require an integration of mechanical strength, reliable control systems, and sensors for vision and obstacle avoidance. Navigation and mapping are crucial to all robotic systems and are an integral part of autonomous mobile robots. While it is possible for a robot to be mobile and not do mapping and navigation, sophisticated tasks require that a mobile robot build maps and use them to move around. Levitt and Lawton (1990) pose three basic questions that define mobile robot mapping and navigation:[1]

1. Where am I?
2. How do I get to other places from here?
3. Where are the other places relative to me?

Humans receive a large amount of their information through the human vision system, which enables them to adapt quickly to changes in their environment. The Center for Robotics Research at the University of Cincinnati has been involved in a worldwide competition to build a small-unmanned autonomous ground vehicle that can navigate around an outdoor obstacle course. Vision-based mobile robot guidance has proven difficult for classical machine vision methods because of the diversity and real time constraints inherent in the task. Vision for motion control must always be real time vision. In this context of unmanned guided vehicles, the vision system must enable the robot to perceive changes in its environment while they are occurring and soon enough to react to these changes and make decisions accordingly. For mobile systems, a speed of reaction similar to human beings is desirable. For this, a robot vision system should not introduce a delay of more than 100 ms in reporting an event in the environment or in providing data for some visible motion. There are several factors, which affect the functioning of the outdoor autonomous systems.

Some are variations of road type, appearance variations due to lighting and weather conditions, real time processing constraints and high level reasoning constraints. A general autonomous vehicle should be capable of driving on a variety of road surfaces like grass, concrete, sand, boards etc. The vehicle should function equally well inside, i.e., on a plane surface as well as outside on a varying terrain. The second factor making autonomous driving difficult is the variation in appearance that results from environmental factors. Lighting changes and deep shadows make it difficult for perception systems to pick up important and desired features during daytime driving. The threshold of the perception system has to be adjusted in such a way that the desired features are identified correctly. Any change to the light affects the threshold and performance of the system. Also to be considered is the fact that missing or obscured lane markers make driving difficult for an autonomous system even under favorable lighting conditions. Adequate computer hardware is a key to practical robot vision. General-purpose computers are definitely not adequate. Multi-processor systems containing a small number of processing elements, each of them based on a standard microprocessor of moderate performance have been demonstrated to outperform much more expensive computer systems in robot vision applications. Flexibility is a very important factor, including the flexibility of random access pixel data by the processing elements, and flexibility in dynamically concentrating the computing power of the system on those parts of an image containing the most relevant information at any moment. The system should also be flexible in restructuring under software control to match the inherent structure of the vision task. There is always a limited amount of time for processing sensor information. The speed of the front end processing system should be such that the vehicle reacts very quickly to the changes in the environment. For example at 5 miles per hour a vehicle is traveling nearly 7.5 feet per second. A lot can happen in 7.5 feet, like losing track of the lane or straying a significant distance from the lane or colliding with an obstacle if the system does not react accurately or act quickly enough.

The purpose of this paper is to simplify this tedious calibration using an artificial neural network. Image processing is used to automatically detect calibration points. Then a back projection neural algorithm is used to learn the relationships between the image coordinates and three-dimensional coordinates. This transformation is the main focus of this study. This paper also describes the design, development and exploratory research on the vision system for the autonomous guided vehicle Bearcat III.

The three dimensional (3-D) vision system makes use of 2 CCD cameras and an image-tracking device for the front end processing of the image captured by the camera. The camera reduces the three dimensional world co-ordinate system into two dimensional image co-ordinate system. After getting the information regarding image co-ordinates, at any time, the challenge is to extract three-dimensional information from them. A mathematical as well as geometrical transformation occurs via the camera parameters in transforming a 3-D coordinate system to a 2-D system. If these mathematical and geometrical relations are known, a 3-D coordinate point on a line can be autonomously determined from its corresponding 2-D image point. To establish these mathematical and geometrical relationships, the camera has to be calibrated. This is because if the vision system is well calibrated, accurate measurements of the coordinates of the points on the line with respect to the robot can be made. From these measurements, the orientation of the line with respect to the robot can be computed. With these computations, the next task is to guide the robot. The motion control of the AGV designed has the capability of turning about the center of its drive axis, which is called the zero turning radius feature. It is gaining popularity and expanding commercially in the U.S. mowing market. This feature provides exceptional maneuverability and can make sharp turns possible with relatively greater ease than those without the ZTR (Zero turning radius) feature. Rotating one wheel forward and the other wheel backward generally accomplishes the ZTR function. However in our design we instead vary the speeds of the left and right drive wheels while negotiating a curve. This enables the AGV to make a curved turning path parallel to the track lines.

2. BACKGROUND AND PREVIOUS RESEARCH

Calibration of a camera means determining the geometric properties of the imaging process i.e. the transformation that maps a 3-D point, expressed with respect to a reference frame onto its 2-D image whose co-ordinates are expressed in pixel units. This problem has been a major issue in photogrammetry and computer vision for years. The main reason for such interest is that the knowledge of the imaging parameters allows one to relate the image measurements to the spatial structure of the observed scene.[2] The fundamental theorem of robot vision says that manipulation of a point in space x_1 by either a robot manipulator that moves it to another point x_2 or through a camera system that images the point onto a camera sensor at x_2, is described by the a matrix transformation, which is of the form $X_2 = Tx_1$

The transformation matrix T describes the first-order effects of translation, rotation, scaling, and projective and perspective projections. Camera calibration is a complex problem because of the following problems:

(1) Calibration of internal parameters of a camera, the so-called intrinsic parameters, including the optical and mechanical (geometrical) properties of the camera, such as focal length, lens distortion parameters, the intersection point of the optical axis with the image plane etc. Sometimes the manufacturers supply these parameters but they are usually not accurate enough for computations. Some of them such as focal length vary with adjustments, while some of them such as the lens center are calibrated once and for all depending upon the optical stability of the camera.

(2) Estimation of the location of the camera (system) relative to the 3-D world reference system, including rotation and translation between these two systems is required. These are called extrinsic parameters. These parameters are not directly related to the camera itself, but the set up of a camera, which means they have to be calibrated at each set up.

Robert in his paper "Camera Calibration without Feature Extraction" has presented an approach to this problem using a calibration pattern.[3] The approach is different from the classical calibration techniques, which involve extraction of image features and computation of camera coefficients. A classical iterative technique is used to search for the camera parameters that best project 3-D points of a calibration pattern. Li and Lavest have thrown light on some aspects of zoom lens camera calibration.[4] A lot of care has to be taken in the electronic stability of the camera and frame grabber, and the way calibration points are measured and detected in images. In that paper they have addressed some practical aspects of camera calibration, in particular, of a zoom lens system. Through a systematic approach they describe all the keys points that have to be checked in order to obtain accurate calibration results.

A lot of caution is required during calibration. Hong, et al. list two points that should be considered in camera calibration:[5]

1. Reducing the location error of image features as far as possible, by exploiting image processing technique, and

2. Compensating system error by the optimal pattern of approximating residual error of image points, namely the posterior compensation of the system error.

Based on these two points, the calibration process discussed by Weng et al. are of three parts:[6] (1) The direct transformation error approximation camera calibration algorithm; (2) the sub pixel image feature location algorithm combined with the 3D control point field delicate design and fabrication; (3) The precisely movable stage, which provides the reliable means of accuracy checking. Tsai presented an algorithm that decomposes a solution for 12 transformational parameters (nine for rotation and three for translation) into multiple stages by introducing a radial alignment constraint.[7] The radial alignment constraint assumes that the lens distortion occurs only in the radial direction from the optical axis Z of the camera. Using this constraint, six parameters are computed first, and the constraint of the rigid body transformation is used to compute five other parameters. The remaining parameters are computed by radial lens distortion parameter and estimating it by a nonlinear optimization procedure.

Zhang et. al in “Analysis of a Sequence of Stereo Scenes Containing Multiple Moving Objects Using Rigidity Constraints” describe a method for computing the movement of objects as well as that of a mobile robot from a sequence of stereo frames.[8] Stereo frames are obtained at different instants by a stereo rig, when the mobile robot navigates in an unknown environment possibly containing some moving rigid objects.

Zhang et al. present a method for estimating 3D displacements from two stereo frames.[9] It is based upon the hypothesize-and-verify paradigm, which is used to match 3D line segments between the two frames. In order to reduce the complexity of the algorithm, objects are assumed to be rigid. In the experimental sections, the algorithm is shown to work on indoor and natural scenes. “A 3D World Model Builder with a Mobile Robot” - An article written by the same authors describes a system to incrementally build a world model with a mobile robot in an unknown environment.[12] The model is segment-based. A trinocular stereo system is used to build a local map about the environment. A global map is obtained by integrating a sequence of stereo frames taken when the robot navigates in the environment. Luong, et al. in their paper "Motion of an Uncalibrated Stereo Rig: Self-Calibration and Metric Reconstruction" address the problem of self-calibration and metric reconstruction (up to a scale) from one unknown motion of an uncalibrated stereo rig, assuming the coordinates of the principal point of each camera are known.[13] They also present a novel technique for calibrating a binocular stereo rig by using the information from both scenes and classical calibration objects. The calibration provided by the classical methods is only valid for the space near the position of the calibration object. Their technique takes the advantage of the rigidity of the geometry between two cameras. The idea is to first estimate precisely the epipolar geometry, which is valid for a wide range in space from all available matches.

During the execution of a task the vision-system is subject to external influences such as vibrations, thermal expansion etc. which affect and possibly render invalid the initial calibration. Moreover, it is possible that the parameters of the vision-system such as the zoom or the focus are altered intentionally in order to perform specific vision-tasks.

"Self-Maintaining Camera Calibration over Time" by Schenk et al. describes a technique for automatically maintaining calibration of stereovision systems over time without using again any particular calibration apparatus.[9] Worrall, Sullivan and Baker in the paper "A simple, intuitive camera calibration tool for natural images" present an interactive tool for calibrating a camera, suitable for use in outdoor scenes.[11] The motivation for the tool was the need to obtain an approximate calibration for images taken with no explicit calibration data. The method decomposes the calibration parameters into intuitively simple components, and relies on the operator interactively adjusting the parameter settings to achieve a visually acceptable agreement between a rectilinear calibration model and his own perception of the scene. Most of the previous research deals with the theoretical aspects of zoom lens camera calibration. The intrinsic parameters are obtained by building a pinhole camera model. The research here deals with designing a calibration algorithm keeping in mind the significant practical aspects.

3. THE CALIBRATION SYSTEM – SIGNIFICANCE

The objective of the vision system is to make the robot follow a line using a camera. In order to obtain accurate information about the position of the line with respect to the centroid of the robot, the distance and the angle of the line with respect to the centroid of the robot has to be known. The camera system reduces the 3-D information about the obstacle course into 2-D image co-ordinates. In order to obtain a relationship between the two co-ordinate systems, the camera needs to be calibrated. Camera calibration is a process to determine the relationship between a given 3-D coordinate system (world coordinates) and the 2-D image plane a camera perceives (image coordinates). More specifically, it is to determine the camera and lens model parameters that govern the mathematical or geometrical transformation from world coordinates to image coordinates based on the known 3-D control field and its image. The CCD camera maps the line from the 3-D coordinate system to the 2-D image system. Since the process is autonomous, the relationship between the 2-D system and the 3-D system has to be accurately determined so that the robot can be appropriately controlled to follow the line. The objective of this section is to explain the entire calibration process and its significance in this project.

The model of the mobile robot illustrating the transformation between the image and the object is shown in Figure 1. The robot is designed to navigate between two lines that are spaced 10 feet apart. The lines are nominally 4 inches wide but are sometimes dashed. This requires a two-camera system design so that when a line is missing, the robot can look to the other side via the second camera. Measurements are referenced to the robot centered as a global coordinate system. For navigation, the cameras must be located with respect to this centroid.

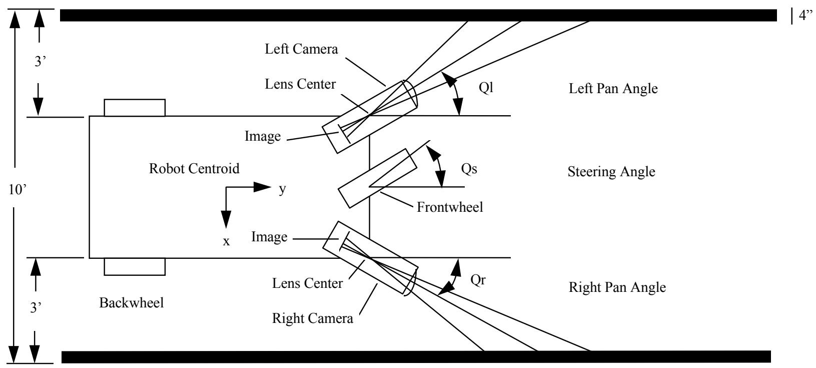

Figure 1. Top view model of the robot in the obstacle course

Camera calibration is considered very important in many computer vision problems. Camera calibration in the context of three-dimensional machine vision is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and/or the 3-D position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters). Camera projection is often modeled with a simple pinhole camera model. In reality, the camera is a much more complicated device, and if it is used as a measurement instrument, a proper calibration procedure should be performed. In order to follow a track, which is separated by two lines, which are 10 ft apart, 2 CCD cameras and an image-tracking device (ISCAN) are used. The ISCAN image tracking system finds the centroid of the darkest or the brightest region in an image and returns the co-ordinates of these points. These are the image co-ordinates. These co-ordinates are two-dimensional while the real world co-ordinates are three-dimensional. An algorithm is developed to establish a mathematical and geometrical relationship between the physical three-dimensional (3-D) and its corresponding digitized two-dimensional (2-D) co-ordinates. In an autonomous situation the challenge is to determine 3-D co-ordinates given the image co-ordinates. This is established by what is popularly known as "Calibration" of the camera. The objective is to find any corresponding ground co-ordinate given an image co-ordinate. What makes this the most important and crucial task is that the process of following the line is autonomous and dynamic and hence the relationship between these co-ordinates should be accurately determined. This, in turn, determines how closely the robot follows the line and hence the success of the robot.

The objective was to design a calibration method, which was not only accurate but also is easy and less time consuming. Some calibration methods are very accurate but are extremely time consuming. The purpose of this paper is to simplify this tedious calibration using an artificial neural network. Image processing is used to automatically detect calibration points. Then a back projection neural algorithm is used to learn the relationships between the image coordinates and three-dimensional coordinates.

4. BACKPROPAGATION

Backpropagation is the basis for training a supervised neural network. Static backpropagation is used to produce an instantaneous mapping of a static (time independent) input to a static output. These networks are used to solve static classification problems such as optical character recognition (OCR). At the core of all back propagation methods is an application of the chain rule for ordered partial derivatives to calculate the sensitivity that a cost function has with respect to the internal states and weights of a network. In other words, the term backpropagation is used to imply a backward pass of error to each internal node within the network, which is then used to calculate weight gradients for that node. Learning progresses by alternately propagating forward the activations and propagating backward the instantaneous errors.

A Backpropagation network consists of at least three layers of units: an input layer, at least one intermediate hidden layer, and an output layer. In contrast to the IAC and Hopfield networks, connection weights in a backpropagation network are one way. Typically, units are connected in a feed-forward fashion with input units fully connected to units in the hidden layer and hidden units fully connected to units in the output layer. When a backpropagation network is cycled, an input pattern is propagated forward to the output units through the intervening input-to-hidden and hidden-to-output weights. The data used to train the neural network is in Appendix C.

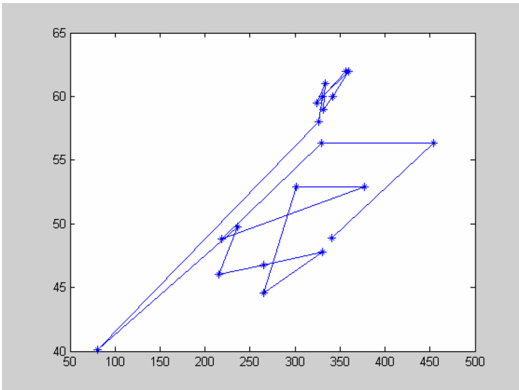

Figure 1: Training plot of *x* coordinate (Trained values are represented by * and actual values are represented by +)

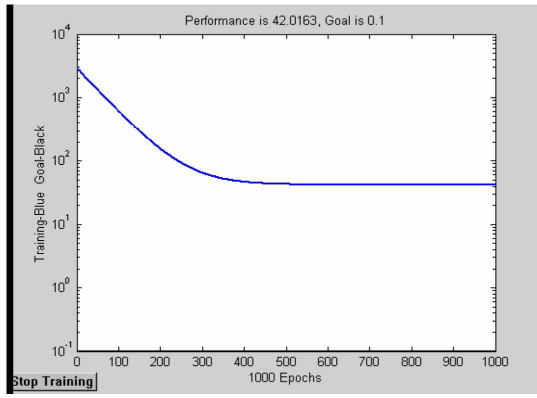

Figure 2: performance plot for *x*

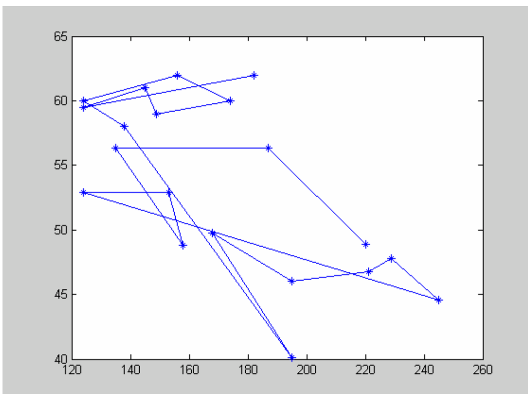

Figure 3: Training plot of *y* coordinate (Trained values are represented by * and actual values are represented by +)

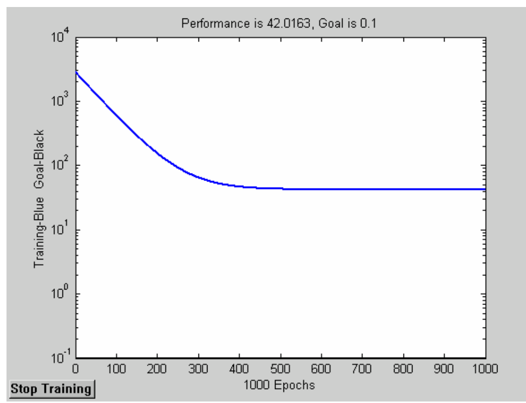

Figure4: performance plot for *y*

5. CONCLUSIONS

The back projection neural algorithm is used to learn the relationships between the image coordinates and three-dimensional coordinates. This transformation is the main focus of this study. Using these algorithms, the robot built with this design is able to track and follow the lines successfully.

REFERENCES

1. T.S. Levitt, and D. T. Lawton, "Qualitative Navigation for Mobile Robots," *Artificial Intelligence*, 44, pp. 305-360, 1990.
2. C. E. Thorpe, editor. "Vision and navigation: the Carnegie Mellon Navlab," The Kluwer international series in engineering and computer science, Robotics Boston: Kluwer Academic Publishers, pp. 4, 1990.
3. L. Robert, "Calibration Without Feature Extraction," *Proceedings of the International Conference on Pattern Recognition*, Jerusalem, Israel, Computer Society Press, pp. 704-706, Washington, DC, 1994.
4. M. X. Li and J.M. Lavest, "Some Aspects of Zoom-Lens Camera Calibration," *IEEE- Transaction on Pattern Analysis and Machine Intelligence*, pp. 1105-1110, 1996.
5. F. Hong and Y.Baozong, "An Accurate and Practical Camera Calibration System for 3D Computer Vision," *Chinese Journal of Electronics*, Vol. 1, No. 1, pp. 63-71, 1991.
6. J. Weng, P. Cohen and M. Herniou, "Camera calibration with distortion models and accuracy evaluation,"' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, No. 10, pp. 965-980, 1992.
7. R. Y.Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-The-Shelf Cameras and Lenses*," IEEE Transaction on Robotics and Automation*, Vol. 8, No. 2, pp. 129-139.
8. Z. Zhang, O. Faugeras, N. Ayache, "Analysis of a Sequence of Stereo Scenes Containing Multiple Moving Objects Using Rigidity Constraints," *Int. Con. on Computer Vision*, pp. 177-186, 1988.
9. Z. Zhang, R. Deriche, O. Faugeras, Q.T. Luong, "A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry," *Artificial Intelligence Journal*, Vol.78, pp. 87-119, October 1995.
10. Z. Zhang and Veit Schenk, "Self-Maintaining Camera Calibration Over Time," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'97)*, pages 231-236, Puerto Rico, June 17-19, 1997
11. A D Worrall, G D Sullivan and K D Baker, "A simple, intuitive camera calibration tool for natural images," *Proceedings of the 4th British Machine Vision Conf.*, University of Reading, Berkshire, UK, 1994.
12. Z. Zhang and O.D. Faugeras, "A 3D World Model Builder with a Mobile Robot," *International Journal of Robotics Research*, Vol. 11, No. 4, pp. 269-285, 1992.
13. Z. Zhang, Q.-T. Luong, and O. Faugeras, "Motion of an Uncalibrated Stereo Rig: Self-Calibration and Metric Reconstruction," *IEEE Trans. Robotics and Automation,* Vol.12, No.1, pp. 103-113, February 1996.

APPENDIX – A

Matlab code for training x of two dimensional image co-ordinates to arrive at the X of three dimensional co-ordinates.

```
p =[1:1:20];
t=[48.856,56.346,56.346,48.831,52.919,52.87,44.55,47.75,46.75,46,49.75,40.1,58,60,62,60,59,61,59.50,62];
plot(p,t,'b',p,t,'*');
p=[341,454,329,219,377,301,265,330,265,215,236,81,326,330,359,342,332,334,324,356];
net=newff(minmax(p),[3,1],{'tansig','purelin'},'traingd');
net.trainParam.show = 100;
net.trainParam.lr = 0.001;
net.trainParam.epochs = 1000;
net.trainParam.goal = 0.1
[net,tr]=train(net,p,t);
a = sim(net,p)
plot(p,t,'b',p,t,'*');
p=[219]
a = sim(net,p)
```

APPENDIX – B

Matlab code for training y of two dimensional image co-ordinates to arrive at the Y of three dimensional co-ordinates.

```
p =[1:1:20];
t=[48.856,56.346,56.346,48.831,52.919,52.87,44.55,47.75,46.75,46,49.75,40.1,58,60,62,60,59,61,59.50,62];
plot(p,t,'b',p,t,'*');
p=[220,187,135,158,153,124,245,229,221,195,168,195,138,124,156,174,149,145,124,182];
net=newff(minmax(p),[3,1],{'tansig','purelin'},'traingd');
net.trainParam.show = 100;
net.trainParam.lr = 0.001;
net.trainParam.epochs = 1000;
net.trainParam.goal = 0.1
[net,tr]=train(net,p,t);
a = sim(net,p)
plot(p,t,'b',p,t,'*');
p=[219]
a = sim(net,p)
```