Synthesizing images with out-of-plane transformations using stereo images

Ruitang Chen, Zhuodong He, Dai Shen

Stanford University

Abstract

Using a pair of stereo cameras, we can acquire the depth information, and thus synthesize images with 3D transformations. However, the resulting images often look uncanny because of data noises. In this project, we attempt to repair the generated stereo images. One way to address this problem is via unsupervised feature learning that could be achieved using linear autoencoders, also known as Reconstruction Independent Component Analysis (RICA). This algorithm has been shown to outperform other algorithms by penalizing lack of diversity among features. We improve the algorithm by adding an extension layer, and compare its performance with another image inpainting technique based on fast marching method. Our research shows that the inpainting method holds good promise.

I. Introduction

In computer vision, the process of obtaining data has often been time-consuming and expensive. Moreover, research outcome often heavily depends on both the quality and quantity of the data used. Consequently, novel ways of data generation has become an increasingly hot research topic. However, generated data often contains corruptions, which is inevitable in the simulation of reality from data at hand. Many of the corruptions fall into the category of inpainting problem. Inpainting problems occur when pixel values are missing,or images contain undesirable noises which hinder the use of the these data in other applications. This project aims to address this issue through Denoising Autoencoder (DA) specifically in the context of stereo view transformations.

One of the biggest motivations in choosing DA is that it employs deep learning that can capture complicated nonlinear structures hidden in the images. DA is a neural network that aims to reconstruct the original image from a noisy version of it through finding out the hidden structure in the image. In our case, the training data are obtained from two stereo cameras mounted on the top of a car. Images from the left camera are transformed and compared to those of the right. Some pixel values will be lost due to data inaccuracies (loss of depth information or occlusion), which will produce noises. DA is the used to restore the lost pixels and reconstruct the original images.

The specific DA algorithm we consider is Reconstruction Independent Component Analysis (RICA), which will be explained in detail in the next section. We also propose a nonlinear extension of RICA which improves the performance. In the end, we implement an inpainting Telea algorithm which outperforms RICA in the denoising process.

II. Methods

I. 3D transformation

The data set is obtained from Kitti[3]. Images are taken from cameras mounted on two extreme sides of a car. Images are rectified in advance. Depth maps are generated using Semi-Global Block Matching (SGBM) algorithm[4] which uses pixel-wise cost calculation. Disparity is the difference between intensity of two pixels at the same location of two images and is inversely proportional to depth, which is derived by the formula:

$$depth = T * f/disparity(x, y), \tag{1}$$

where T is the distance of two cameras, and f is focal length of camera. Hence we are able to get the real world depth data and map the pixel on the image to 3D location according to the formula:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & C_u \\ 0 & f_y & C_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix}, \tag{2}$$

where f_x, f_y are focal lengths, C_u,C_v are coordinates of the principal point of the camera, and (X, Y, Z) is the real-world coordinate.

Figure 1: *A disparity map.*

New image at a different angle can be obtained through imposing the following rotation matrix such as the one along z-axis,

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

In the original real world coordinate. We can make line transformation, and use the same method to project real world image data to 2D plane.

Figure 2: *a left image that has been transformed to match that of the right. Many pixels lost their values during the trasnformation.*

However, in this process some pixel values will lose due to inaccurate calculation of depth information. Hence we are trying to repair these black holes of images. We then propose three methods, RICA, deep RICA and inpainting.

II. RICA

Model In this section, we introduce the basic models of RICA which serve as the foundation in our denoising method. Compared to traditional autoencoder, RICA is able to train data with overcomplete features (number of features » dimentionality of data) and is insensitive to whitening (a preprosessing step that decorrelate the data). Therefore, this algorithm has been shown to outpeform many other autoencoder algorithms.
Suppose we have unlabeled training example set $\{x^{(1)}, x^{(2)}...\}$, in our case are vectors transformed by corrupted-image matrix, $\{y^{(1)}, y^{(2)}...\}$ are the original right image. We modified the problem to be training $\{x^{(1)}, x^{(2)}...\}$ to find an encoding matrix W^TW to restore $x^{(i)}$ to $y^{(i)}$.

Hence, RICA produce the following unconstrained optimization problem:

$$\min_W [\frac{\lambda}{m} \sum_{i=1}^{m} \|W^T W x^{(i)} - y^{(i)}\|_2^2 + \sum_{i=1}^{m} \sum_{j=1}^{n} g(W_j x^{(i)})],$$

where g is a nonlinear convex function, in our case is the sparsity cost.

$$g(W_j x^{(i)}) = \lambda * (\epsilon + W_j x^{(i)}))^{(\frac{1}{2})}$$

$W \in \mathbb{R}^{k \times n}$ is the weight matrix, where k is number of features. W is the encoding matrix, hence the encoding step is Wx^i. The activation function of RICA is proposed to be a linear function of $\{Wx^{(i)}\}$. Next we are going to propose a deep version of RICA, which use an extra layer of $f(\{Wx^{(i)}\})$ For convenience, we define that

$$H(W) = \sum_{i=1}^{m} \|W^T W x^{(i)} - y^{(i)}\|_2^2,$$

$$G(W) = \sum_{i=1}^{m} \sum_{j=1}^{n} g(W_j x^{(i)}),$$

$$J(W) = \frac{\lambda}{m} H(W) + G(W).$$

Process of RICA

1. Initialize parameters (λ, ϵ, number of hidden units).
2. Use backpropagation algorithm to calculate $\{a^{(2)}, a^{(3)}, \delta^{(1)}, \delta^2, \delta^{(3)}\}$.
   $a^{(l)}$ to denote the activation of layer l
3. Calculate $\bigtriangledown_W H(W)$,$\bigtriangledown_W G(W)$, $\bigtriangledown_W J(W)$.
4. Use the unconstrained optimizer L-BFGS to find $W^* = \underset{W}{\text{argmin}}\ J(W)$.
5. encode corrupted image by imposing $Wx^{(i)}$.
6. Calculate squared error $\|Wx^{(i)} - y^{(i)}\|_2^2$ to validate accuracy.

Data processing Using Kitti dataset, we are able to obtain a disparity map for each set of images. Then we artificially generate right image based on the camera calibration data we get. Due to limit of computer memory space, W cannot be high dimension. We divide the picture into 32 pixel $\times$ 32 pixel patches as training sets. Then we reshape image matrix into vector $\{x^{(1)}, x^{(2)}...\}$, the object vectors $\{y^{(1)}, y^{(2)}...\}$ are original right images obtained from Kitti data set. In this project, we run RICA on 25 pictures of 300 patches.

III. Deep RICA

Due to the linearity of RICA, data underfitting may occur. In this sense, we propose to add an additional hidden layer to the neural network. For inputs $x^{(i)}$, We preprocess it and calculate the new input as $sigmoid(Ux^{(i)})$. And we also use backpropagation on the additional layer. We train the data on the same data set as RICA.

Effect of parameter λ:
λ plays an important in the calculation of cost. Very small values of λ provide inadequate weight to the error term of $H(W)$. For our study, we chose λ to be 0.05.

Effect of number of hidden units (features):
A low number of hidden units make poor predictions. Whereas high number of hidden units may lead to overfitting. We run our algorithm at k = 400.

IV. Inpainting

Here we employ telea algorithm for inpainting. To inpaint a region Ω, we select a boundary point p and take a small neighborhood $B_\epsilon(p)$ of size ϵ of the known image around p . The inpainting of p should be determined by the values of the known image points close to p, i.e., in $B_\epsilon(p)$. For ϵ small enough, we consider a first order approximation $I_q(p)$ of the image in point p, given the image $I(q)$ and gradient $\bigtriangledown I(q)$ values of point q.

The inpainting method is described by:

$$I_q(p) = I(q) + \bigtriangledown I(q)(p - q) \tag{4}$$

$$I(p) = \frac{\sum_{q \in B_\epsilon(p)} \omega(p,q)[I(q) + \bigtriangledown I(q)(p-q)]}{\sum_{q \in B_\epsilon(p)} \omega(p,q)} \tag{5}$$

The above equation explains how to inpaint a point on the boundary of the to-be-inpainted region as a function of known image pixels. To inpaint the whole Ω, we iteratively apply Equation(4) to all the discrete pixels of $\partial\Omega$ and advance the boundary inside Ω until the whole region has been painted. Inpainting points in increasing distance order from $\partial\Omega$ ensures that images closest to known image are filled in first, thus mimicking manual inpainting techniques.
To implement this, we use a fast matching method which solves Eikonal equation:
$\| \bigtriangledown T\| = 1$ on Ω, with $T = 0$ on $\partial\Omega$. The soluion T is the distance map of Ω pixels to the boundary.

Data processing In the noisy images we synthesized, we define the to-be-inpainted region,i.e. the mask as all black pixels. and then we train each image with telea algorithm.

III. Results

The following are some of the images after rotation transformations. They show that more than 45 degree transformation would be accompanied by too many distortions and occlusion effects to be of practical usage.

Here a is the image rotated upwards by 10° repaired by telea, b is the image rotated downwards by 10° repaired by telea, c is the image rotated to the right by 30° repaired by telea, d is the image rotated to the right by 60° repaired by telea, e is the image repaired by linear RICA, f is the image repaired by nonlinear RICA. Images repaired by deep RICA has smaller square error than linear RICA. We are able to generate rather clear image by Telea till 30°. Image rotated 60° is comparatively distorted.

We validate the error rate of repaired images by calculating its mean square error by formula $\sum_{i=1}^{m} \|x^{(i)} - y^{(i)}\|_2^2$ comparing to the original images.

Table 1: *mean squared error of repaired images*

Mean-squared error of Methods (1e3)		
Telea	RICA	deep-RICA
2.2	2.4	2.4

IV. Discussion

I. Advantages and Limitations

One of the limitations of RICA is that it relies common inner structures of corrupted pictures. However, in our dataset, noises tend to center in certain areas (e.g. the sky) where disparity information is inaccurate due to far distance. Noises of this kind more often than not bear few similarities, if any at all. In fact, RICA works most effectively on Gaussian noises, which is not a very accurate model of the reality.

Deep RICA is a better model with a lower mean squared error than the linear RICA. However, the added complexity leads to a longer run time, especially for large data sets. In our data set, training time is still acceptable. Telea algorithm generates repaired images with the lowest mean squared errors. Instead of finding common features in images, this method is based on local optimization. Since an inpainting mask tells the algorithm which pixels correspond to the noises, this method only works in the predefined region. It is also more cost-effective in terms of implementation. Inpainting does not need other image information to train the data sets. It simply repairs the image based on the image itself. However, in some cases where noise is undefined, inpainting is impossible. Also, inpainting distance is a parameter that changes with respect to different image conditions. For large data set, this is also impossible to tune.

By comparison, RICA, which requires no predefinition of corrupted pixels, has broader applications. However, in synthesizing 2D images with out-of-plane transformations, Telea may be a better choice.

V. Conclusion

Through 3D transformation and image restoration process, we are able to generate images of different orientations and locations. In so doing, We can acquire data much faster and more efficiently. For future work, we would also like to extrapolate images beyond camera positions, which would be more difficult than our current interpolated images.

VI. Acknowledgement

We would like to thank Mr.Tao Wang, a phD student at Stanford CS department, for helping us with the project.

References

[1] *ICA with Reconstruction cost for efficient overcomplete feature learning.* Andrew Y.N. et. al.

[2] *An image inpainting technique based on the fast marching method.* Alexandru.T.

[3] *The KITTI Vision Benchmark Suite* `http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo`

[4] *Semi-Global Block-Matching Algorithm* `http://zone.ni.com/reference/en-XX/help/372916M-01/nivisionconceptsdita/guid-53310181-e4af-4093-bba1-f80b8c5da2f4/`

[5] *Image Denoising and Inpainting with Deep Neural Networks* `http://hebb.mit.edu/courses/9.641/lectures/Xie%20Xu%20Chen%20image%20denoising%20inpainting%20deep%20neural%20networks%2012.pdf`