ARBITRARY FACTOR IMAGE INTERPOLATION USING GEODESIC DISTANCE WEIGHTED 2D AUTOREGRESSIVE MODELING

Ketan Tang, Oscar C. Au, Yuanfang Guo, Jiahao Pang, Jiali Li

The Hong Kong University of Science and Technology
{tkt, eeau, eeandyguo, jpang, jiali}@ust.hk

ABSTRACT

Least square regression has been widely used in image interpolation. Some existing regression-based interpolation methods used ordinary least squares (OLS) to formulate cost functions. These methods usually have difficulties at object boundaries because OLS is sensitive to outliers. Weighted least squares (WLS) is then adopted to solve the outlier problem. Some weighting schemes have been proposed in the literature. In this paper we propose to use geodesic distance weighting in that geodesic distance can simultaneously measure both the spatial distance and color difference. Another contribution of this paper is that we propose an optimization scheme that can handle arbitrary factor interpolation. The idea is to separate the problem into two parts, an adaptive pixel correlation model and a convolution based image degradation model. Geodesic distance weighted 2D autoregressive model is used to model the pixel correlation which preserves local geometry. The convolution based image degradation model provides the flexibility to handle arbitrary interpolation factor. The entire problem is formulated as a WLS problem constrained by a linear equality.

Index Terms— interpolation, geodesic distance, autoregressive model, arbitrary factor

1. INTRODUCTION

Image interpolation aims to get a High Resolution (HR) image from a corresponding Low Resolution (LR) image through interpolation technique. Conventional linear interpolation methods include bilinear and bicubic interpolation. Although their complexity is relatively low, they have common drawback that they cannot adapt to varying pixel structures in an image because of the use of constant interpolators. As a result, they suffer from some inherent defects such as staircase effect, blurred details, and ringing artifacts.

To solve the common drawback of linear interpolation, adaptive interpolation methods have been proposed in which local geometric structures are better preserved. Li and Orchard proposed New Edge Directed Interpolation (NEDI), in which the interpolation coefficients are estimated from LR image and then applied on HR image, assuming geometric duality holds for a local window [1]. In [2] Zhang and Wu extended NEDI by proposing a Soft-decision Adaptive Interpolation (SAI) method, in which the interpolation coefficients are still estimated from LR image, but the pixel values of HR image are estimated by averaging multiple estimators from overlapped blocks.

One common problem with NEDI and SAI is that they use ordinary least squares (OLS) estimation for the interpolation coefficients. OLS has an intrinsic drawback that it is prone to outliers. Thus at regions of object boundaries, both NEDI and SAI have problems to accurately estimate HR pixel values. To solve this problem some weighted least squares (WLS) based methods have been proposed. In [3] Zhang et. al. suggested to use nonlocal means for weighting the residuals; In [4] Huang et. al. suggested using bilateral filter for weighting; In [5] Liu et. al. proposed combining bilateral filter and nonlocal means for weighting; In [6] Hung et. al. proposed using covariance to weight the residuals. In this paper we propose to use geodesic distance weight for both parameter and data estimation, in that geodesic distance can simultaneously measure both the spatial distance and color difference.

There are also some other problems with the above mentioned NEDI-based methods. Firstly, geometric duality may not always be true. For fine textures, the pixel correlations at different scales are different. Interpolation coefficients learned in a larger scale cannot be directly used in a smaller scale. Secondly, those NEDI-based methods can only handle zooming factor of two. For other zooming factors, they have to do adaptively interpolation multiple times, and then apply a conventional linear interpolation such as bicubic. For example to do 1.8 times zooming they have to first interpolate the image to 2 times large, then downsample it by 0.9 times, resulting in an image of 1.8 times large in the end. In this process, errors may accumulate and thus produce some artifacts such as blurring.

To solve the problem of estimating interpolation coefficients at different scales, we proposed AutoRegressive model and Gauss-Seidel optimization (ARGS) in our previous work [7]. In this paper we further improve the interpolation coefficient estimating by splitting the Gauss-Seidel iterations into two steps. At the first step (first Gauss-Seidel iteration) only LR pixels are used to estimate the interpolation coefficients, since all other pixels are untrustworthy. At the second step (following Gauss-Seidel iterations) all pixels are used to estimate the interpolation coefficients.

To solve the problem of being unable to handle arbitrary factor interpolation, we propose to combine convolution based interpolation and adaptive interpolation. The idea is to separate the target problem into two parts, an adaptive pixel correlation model serving as a cost function and a convolution-based image degradation model serving as a constraint. Piecewise autoregressive model (PAR) model is used for the adaptive pixel correlation modeling. The image degradation process is modeled as convolution-based image downsampling which is flexible enough to adapt to any zooming factor and any kernel, and can be used as a linear constraint to the PAR least square problem.

The rest of the paper is organized as follows. In section 2 we explain the algorithm, including the geodesic distance in 2.1 and the weighted 2-D autoregressive modeling in 2.2. We present the soluton of this problem in section 3. The experiment results are given in 4. And we conclude our work in section 5.

2. ALGORITHM

2.1. Geodesic Distance

The idea of geodesic distance weighted least squares is to incorporate the spatial correlation of pixels within a local block into the least square cost function. The strength of geodesic distance is that it simultaneously incorporates both spatial distance and pixel value distance, and is very robust to outliers [8]. Define the center pixel c of a local block as most important, i.e. its weight is 1. Then the importance of the cost of another pixel p is inversely proportional to its geodesic distance to the center pixel, $D(p,c)$. The geodesic distance $D(p,c)$ is defined as the shortest path that connects p with c:

$$D(p,c) = \min_{P \in \mathcal{P}_{p,c}} d(P)$$

where $\mathcal{P}_{p,c}$ is the set of all paths connecting p and c. A path P is defined as a sequence of spatially neighboring points in 8-connected neighborhood. Let the sequence P be $P = \{p_1, p_2, \ldots, p_n\}$, then the cost is computed by

$$d(P) = \sum_{i=2}^{n} d_C(p_i, p_{i-1})$$

with $d_C(p_i, p_{i-1}) = |I(p_i) - I(p_{i-1})|$ measuring the color difference between pixel p_i and p_{i-1} of image I. Intuitively, if there exists a path between pixel p to the window center c along which the intensity does not change much, the geodesic distance $D(p,c)$ is low. And larger weights should be given to the pixels that has smaller geodesic distances to the center pixel c:

$$\theta(p,c) = \exp\left(\frac{-D(p,c)}{\beta}\right)$$

where β is a user-defined parameter controlling the importance of geodesic distance weighting. A smaller β means a higher importance of geodesic distance weighting. Although computing the geodesic distance for all pixels in a local window is a NP-hard problem, a fast approximation algorithm is reported in [8].

2.2. Weighted 2-D Autoregressive Modeling

Assuming the geometric property of a local window in a natural image is constant, an image can be modeled as a piecewise autoregressive (PAR) process [2]

$$X(i,j) = \sum_{(m,n)\in T} \alpha(m,n) X(i+m, j+n) + v_{i,j} \quad (1)$$

where $T = \{(-1,-1), (-1,0), (-1,1), (0,-1), (0,1), (1,-1), (1,0), (1,1)\}$ is the set of 8-connected neighborhood, $v_{i,j}$ is a random perturbation independent of spatial location and the image signal. $\alpha(m,n)$ is the autoregressive coefficient for the (m,n)-th neighbor of pixel $X(i,j)$, which is assumed constant in local window T but different between different windows.

Let I_h be the HR image to be estimated by interpolating the observed LR image I_l, which is a downsampled version of the HR image by an arbitrary factor. Let $x_i \in I_l$ and $y_i \in I_h$ be the pixels of images I_l and I_h, $y_{i\diamond t}$ $(t = 1, 2, \ldots, 8)$ be the neighbors of pixel location i in the HR image. The HR image can be estimated block by block with the following weighted least squares problem

$$\begin{aligned} \min_{\{y,\alpha\}} \quad & F(y,\alpha) = \\ & \sum_{i\in W} (y_i - \sum_t \alpha_t y_{i\diamond t})^2 \theta(y_i, y_c) \\ \text{s.t.} \quad & DH\tilde{y} = x \end{aligned} \quad (2)$$

where W is a window of HR image I_h, y_i is the i-th pixels in W counted in scanning order, y_c is the center pixel of W, x is the vector of LR pixels that are inside W, H is anti-aliasing filter matrix, D is decimation matrix, and sampling matrix $S = DH$ computes LR pixel vector x from HR block. $\tilde{y}$ is the vector of all HR pixels related to x, including y and pixels outside W. Let v be the vector of pixels outside W but related to x, then we have $\tilde{y} = \begin{bmatrix} y^T & v^T \end{bmatrix}^T$. Note that with different sampling matrix S, the outer-pixel vector v can be different. For example when H is identity matrix, meaning there is no anti-aliasing filtering before downsampling, then $v = \mathbf{0}$. Let $S = [S_y, S_v]$ where S_y consists of the columns corresponds to y and S_v consists of the rest columns which correspond to v, the equality constraint can be written as

$$\begin{aligned} & \begin{bmatrix} S_y & S_v \end{bmatrix} \begin{bmatrix} y \\ v \end{bmatrix} = x \\ \iff \quad & S_y y = x - S_v v \end{aligned}$$

In our implementation we assume the downsampling kernel is bilinear for its simplicity. An optimal kernel may be estimated based on the test image's characteristics before interpolation, however we found generally the results of bilinear kernel are fairly good.

The problem with Eq. (2) is that it treats diagonal and horizontal/vertical (HV) neighbors equally. However since they have different spatial distance to the center pixel, they should have different influence strength. Therefore we further split the neighboring position set T into two sets $T = \{T_d, T_{hv}\}$:

$$\begin{aligned} T_d &= \{(-1,-1), (-1,1), (1,-1), (1,1)\}, \\ T_{hv} &= \{(-1,0), (0,-1), (0,1), (1,0)\}. \end{aligned}$$

Diagonal neighbors of $x(i)$ are defined as $x_t^d(i) = x(\boldsymbol{s}(i) + T_d(t)), t = 1, \ldots, 4$, HV neighbors of $x(i)$ are defined as $x_t^{hv}(i) = x(\boldsymbol{s}(i) + T_{hv}(t)), t = 1, \ldots, 4$, where $\boldsymbol{s}(i)$ is the 2D position vector of pixel $x(i)$. The interpolation coefficients are also split into two sets, $\alpha = \{a, b\}$ where a corresponds to diagonal neighbors and b corresponds to HV neighbors. Then our problem is reformulated as

$$\begin{aligned} \min_{\{y,a,b\}} \quad & F(y,a,b) = \\ \sum_{i\in W} & \left((y(i) - \sum_{t=1}^{4} a_t y_t^d(i))^2 + \lambda^2 (y(i) - \sum_{t=1}^{4} b_t y_t^{hv}(i))^2 \right) \theta(y_i, y_c) \\ \text{s.t.} \quad & S\tilde{y} = x \end{aligned} \quad (3)$$

where λ^2 controls the importance of HV neighbors over diagonal neighbors. This parameter serves as an additional degree of freedom.

3. JOINT OPTIMIZATION USING GAUSS-SEIDEL ITERATIONS

As shown in [7] that the ordinary least squares (OLS) problems can be effectively solved using Gauss-Seidel method. For weighted least squares (WLS) the optimization is quite similar, since here the weights $\theta(y_i, y_c)$ are precomputed and can be treated as constant during optimization.

Gauss-Seidel method is to alternatively fix one set of variables and optimize on the other set. The iterative equations of this problem are as follows.

$$\{a^{(n+1)}, b^{(n+1)}\} = \arg\min_{a,b} F(y^{(n)}, a, b) \quad (4)$$

$$y^{(n+1)} = \arg\min_{y} F(y, a^{(n+1)}, b^{(n+1)}) \quad (5)$$

Initial value of y can be obtained by bicubic interpolation.

3.1. Estimating interpolation parameters

For (4), since a and b are naturally decoupled, it can be divided into two sub-problems, and a and b have closed form solutions:

$$a^{(n+1)} = \arg\min_{a} \sum_{i \in W} (y_i - \sum_t a_t y^d_{i \diamond t})^2 \theta(y_i, y_c)$$

$$b^{(n+1)} = \arg\min_{b} \sum_{i \in W} (y_i - \sum_t b_t y^{hv}_{i \diamond t})^2 \theta(y_i, y_c)$$

The above weighted least squares minimization have closed form solution as follows [9].

$$a^{(n+1)} = (Y_a^T(\Theta Y_a))^{-1} Y_a^T(\Theta y)$$

$$b^{(n+1)} = (Y_b^T(\Theta Y_b))^{-1} Y_b^T(\Theta y)$$

where $\Theta = diag(\theta(y_1, y_c), \theta(y_2, y_c), \ldots, \theta(y_k, y_c)) \in \mathcal{R}^{k \times k}$ is a diagonal matrix of all weights. Y_a is the matrix of diagonal neighbors of y (the i-th row of matrix Y_a consists of the diagonal neighbors of y_i, i.e. $Y_a(i, \cdot) = [y^d_{i\diamond 1}, y^d_{i\diamond 2}, y^d_{i\diamond 3}, y^d_{i\diamond 4}]$). Y_b is the matrix of HV neighbors of y (the i-th row of matrix Y_b consists of the HV neighbors of y_i, i.e. $Y_b(i, \cdot) = [y^{hv}_{i\diamond 1}, y^{hv}_{i\diamond 2}, y^{hv}_{i\diamond 3}, y^{hv}_{i\diamond 4}]$). Note that all elements of y are inside W, however elements of Y_a and Y_b may be outside W. For the ease of representation we denote $a^{(n+1)}$ and $b^{(n+1)}$ by $\hat{a}$ and $\hat{b}$ when estimating image data.

At the beginning step of this Gauss-Seidel iterations, since most of the HR pixels are untrustworthy, we learn the parameters from LR pixels only. And based on the geometric duality assumption, the same parameters are used to estimate HR pixel values. Then at following iterations we learn the parameters from HR pixels until convergence.

After a, b are computed, the λ value in Eq. 7 can be estimated by the fitting error of a and b:

$$\lambda = \|y - Y_a a\|^2 / \|y - Y_b b\|^2 \quad (6)$$

3.2. Estimating image data

For (5), we have only $\tilde{y}$ as a variable in the cost function

$$\begin{aligned} F(y) &= \sum_{i \in W} ((y(i) - \sum_{t=1}^{4} a_t y^d_t(i))^2 \\ &\quad + \lambda^2 (y(i) - \sum_{t=1}^{4} b_t y^{hv}_t(i))^2) \theta(y_i, y_c) \\ &= (C^d \tilde{y})^T (\Theta C^d \tilde{y}) + \lambda^2 (C^{hv} \tilde{y})^T (\Theta C^{hv} \tilde{y}) \end{aligned}$$

Table 1: PSNR (dB) results of different interpolation methods applied on bilinear downsampled images.

Images	lanczos [11]	NEDI [1]	SAI [2]	proposed
baboon	24.04	23.02	23.15	**25.14**
bike	24.70	22.82	22.95	**26.72**
flower	22.48	20.78	20.96	**23.53**
lena	33.28	30.02	30.19	**35.76**
necklace	21.49	18.87	19.19	**23.16**
parrot	32.60	30.45	30.56	**35.02**
building	24.34	22.74	22.75	**27.65**
tree	27.50	25.04	25.32	**29.51**
average	26.30	24.22	24.38	**28.31**

where

$$C^d(i, j) = \begin{cases} 1, & \text{if } \tilde{y}_j = y_i \\ -\hat{a}_t, & \text{if } \tilde{y}_j = y^d_{i\diamond t}, \quad t = 1, \ldots 4 \\ 0, & \text{otherwise} \end{cases}$$

$$C^{hv}(i, j) = \begin{cases} 1, & \text{if } \tilde{y}_j = y_i \\ -\hat{b}_t, & \text{if } \tilde{y}_j = y^{hv}_{i\diamond t}, \quad t = 1, \ldots 4 \\ 0, & \text{otherwise} \end{cases}$$

By defining $C = \begin{bmatrix} (C^d)^T & \lambda (C^{hv})^T \end{bmatrix}^T = \begin{bmatrix} C_y & C_v \end{bmatrix}$, $\Theta' = diag(\Theta, \Theta)$, where C_y consists of the columns of C corresponding to y, C_v consists of the remaining columns of C corresponding to v, we have

$$\begin{aligned} y^{(n+1)} &= \arg\min_{y} \quad F(y) = (C\tilde{y})^T (\Theta' C \tilde{y}) \\ &\qquad \triangleq (C_y y + C_v v)^T (\Theta' C_y y + \Theta' C_v v) \\ \text{s.t.} &\qquad S_y y = x - S_v v \end{aligned} \quad (7)$$

This equality constrained quadratic optimization has a closed form solution by applying Karush–Kuhn–Tucker (KKT) conditions (§10.1.1 of [10]):

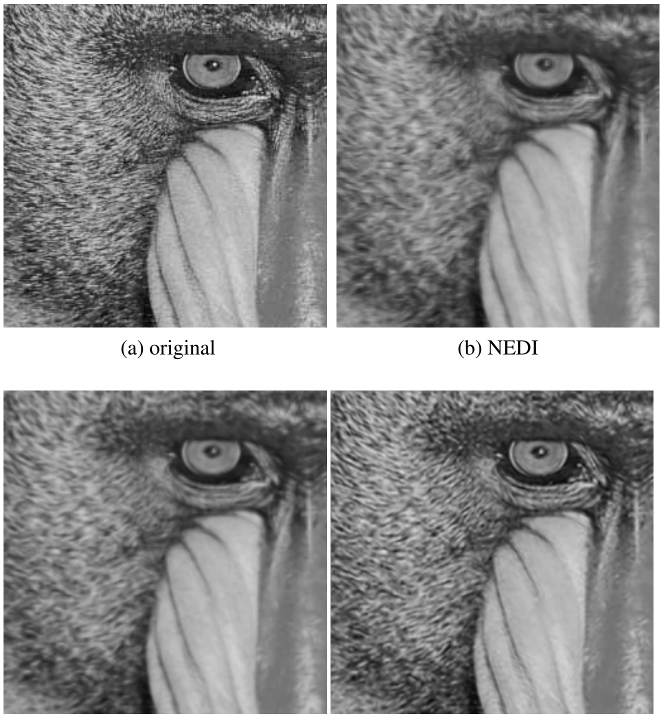
$$\begin{bmatrix} y \\ \mu \end{bmatrix} = \begin{bmatrix} C_y^T(\Theta' C_y) & S_y^T \\ S_y & 0 \end{bmatrix}^{-1} \begin{bmatrix} -C_y^T(\Theta' C_v) v \\ x - S_v v \end{bmatrix} \quad (8)$$

where μ is Lagrange multiplier which can be ignored.

4. EXPERIMENT RESULTS

In our experiments we test eight natural images, most of which contain complex textures which are difficult for interpolation. LR images are obtained by using bilinear downsampling on HR images and PSNR values are calculated as a measure of performance. Table. 1 shows the PSNR results of applying different interpolation methods for zooming factor of 2. We choose this zooming factor so that we can compare our algorithm with some existing ones. As can be seen that our proposed method performs best for all images. On average we have about 2dB gain over the lanczos which is the second best.

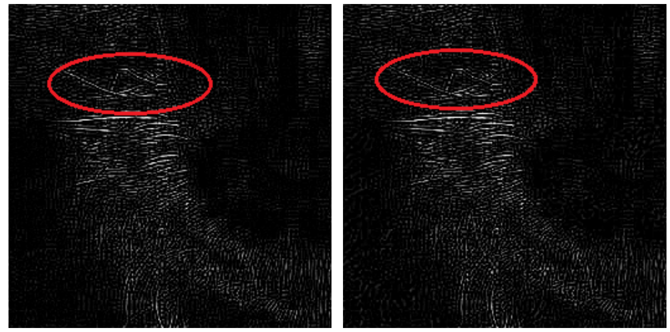
Fig. 1 shows the results of different interpolation algorithms. Lanczos interpolation methods generate blur HR images. SAI method preserves strong edge structures very well. The edges are also sharper than bicubic and lanczos. However, the results of NEDI and SAI are still a little blur. Compared to all the other methods, the



(a) original (b) NEDI

(c) SAI (d) proposed

Fig. 1: Results of different interpolation methods applied on bilinear downsampled **baboon** images. Due to space limit, only a part is shown.



(a) without weighting (b) with weighting

Fig. 2: Absolute residual multiplied by four.

(a) SAI + bicubic (b) proposed algorithm

Fig. 3: Results on *original* **lena** image with zooming factor 1.8. (a) SAI ($2\times$ enlarge) followed by bicubic ($0.9\times$ shrink). (b) the proposed algorithm.

proposed method generates much sharper and more natural result. Special attention may be paid on the hair region where our result contains more natural details, and also the face region where NEDI and SAI methods generate over-smooth results.

To evaluate the effectiveness of the geodesic distance weighting, we draw the residual image in Fig. 2. In the highlight region we can see that with weighting the strong edge is better recovered. However we also notice in regions with less strong edges, the result with geodesic distance weighting has larger residual. A possible reason may be that geodesic distance is sensitive to noise. Some region-based technique such nonlocal means may be used to further improve the geodesic distance weighting.

One may think that SAI interpolation (may be performed multiple times) followed by simple convolution based interpolation (e.g. bicubic) is enough for arbitrary factor interpolation (as suggested in [2]). We have found that this "SAI+bicubic" scheme is not good enough in general. Fig. 3 shows the interpolation results of applying different methods on original **lena** image with zooming factor 1.8 (no downsampling is involved in this example). For the proposed algorithm, we directly interpolate the original image to 1.8 times of original size. For SAI algorithm we first interpolate it into 2 times large and then downsample it by 0.9 using bicubic kernel, resulting in an image of 1.8 times of original size as well. We can see that Fig. 3(a) is quite blur. On the other hand, the proposed method generates much sharper result, and also the lines of hair are more continuous. The only problem is that there is more noise in the proposed algorithm's result, which is because the proposed algorithm tries to recover high frequencies from the original, including both the real image detail and unpleasant noise. We believe there is still room for improvement.

5. CONCLUSION

In this paper we propose an effective arbitrary factor interpolation algorithm based on geodesic distance weighted autoregressive model. Geodesic distance weighting makes the autoregressive model more robust to outliers and recovers better at object boundaries. The image degradation constraint ensures the result HR image agree with the LR input image, and it can adapt to arbitrary zooming factor and arbitrary convolution kernel. Thus the resulting HR images are smooth along edge direction and sharp across edge direction. Experiment results show that while achieving arbitrary factor interpolation, our proposed algorithm also achieves higher visual quality than conventional adaptive interpolation method SAI.

Acknoledgement

This work has been supported in part by the Research Grants Council (RGC) of the Hong Kong Special Administrative Region, China. (GRF 610109 and GRF 610112)

6. REFERENCES

[1] X. Li and M. T. Orchard, "New edge-directed interpolation," *Image Processing, IEEE Transactions on*, vol. 10, no. 10, pp. 1521–1527, 2001.

[2] X. Zhang and X. Wu, "Image interpolation by adaptive 2-d autoregressive modeling and soft decision estimation," *Image Processing, IEEE Transactions on*, vol. 17, no. 6, pp. 887–896, 2008.

[3] X. Zhang, S. Ma, Y. Zhang, L. Zhang, and W. Gao, "Nonlocal edge-directed interpolation," pp. 1197–1207, 2009.

[4] K.-W. Hung and W.-C. Siu, "Improved image interpolation using bilateral filter for weighted least square estimation," *ICIP*, 2010.

[5] X. Liu, D. Zhao, R. Xiong, S. Ma, W. Gao, and H. Sun, "Image interpolation via regularized local linear regression," *Image Processing, IEEE Transactions on*, vol. 20, no. 12, pp. 3455–3469, 2011.

[6] K.-W. Hung and W.-C. Siu, "Robust soft-decision interpolation using weighted least squares," *Image Processing, IEEE Transactions on*, vol. 21, no. 3, pp. 1061–1069, 2012.

[7] K. Tang, O. C. Au, L. Fang, Z. Yu, and Y. Guo, "Image interpolation using autoregressive model and gauss-seidel optimization," in *Image and Graphics (ICIG), 2011 Sixth International Conference on*, 2011, pp. 66–69.

[8] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, "Local stereo matching using geodesic support weights," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 2009, pp. 2093–2096.

[9] H. Wendland, *Scattered Data Approximation*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[11] C. E. Duchon, "Lanczos filtering in one and two dimensions," *Journal of Applied Meteorology*, vol. 18, pp. 1016–1022, Aug. 1979.