

Self-similarity Based Editing of 3D Surface Textures

Junyu Dong Lin Qi Jing Ren
Department of Computer Science
Ocean University of China
Qingdao, China, 266071
dongjunyu@ouc.edu.cn

Mike Chantler
Texture Lab
Heriot-Watt University
Edinburgh, UK, EH14 4AS
M.J.Chantler@hw.ac.uk

Abstract

This paper presents inexpensive methods for self-similarity based editing of real-world 3D surface textures. Unlike self-similarity based 2D texture editing approaches which only make changes to pixel color or intensity values, these techniques also allow surface geometry, reflectance and other representations of the captured 3D surface textures to be edited and relit using illumination directions that differ from those of the original. A single editing operation at a given location affects all similar areas and produces changes on all images of the sample rendered under different conditions. We perform painting, cloning and warping operations over two sets of 3D surface texture representations: (1) surface height and albedo maps, and (2) eigen base images. The surface height and albedo maps are commonly used for bump mapping, and eigen base images can be used for representing 3D surface textures with complex reflectance. The result representations can be efficiently used in modern computer graphics packages for real-time applications as rendering process only requires simple graphics calculations such as weighted sums of base images.

1. Introduction

As an effective way to produce new textures, texture editing has been paid a great deal of attention. In recent years, researchers focused on developing methods that can automate the editing process [1,10,12]. These approaches normally use a 2D texture image as input and can perform painting, warping; or can change the orientations, sizes, and randomness of “macrostructures” over the sample texture to generate new textures.

However, real-world surface textures normally comprise rough surface geometry and various reflectance properties. Their images can therefore vary dramatically with illumination. For example, Figure 1 (a) and (b) show two example images of a 3D surface

texture illuminated from two directions. Thus, if the subjects to be edited are 3D surface textures (such as brick, woven or knitted textiles, embossed wallpapers etc.), the 2D editing technique cannot provide the information required for rendering under other than the original illumination conditions. This presents obvious limitations for high-fidelity rendering of textures in augmented and virtual reality applications. It is therefore highly desirable to develop 3D surface texture editing methods.

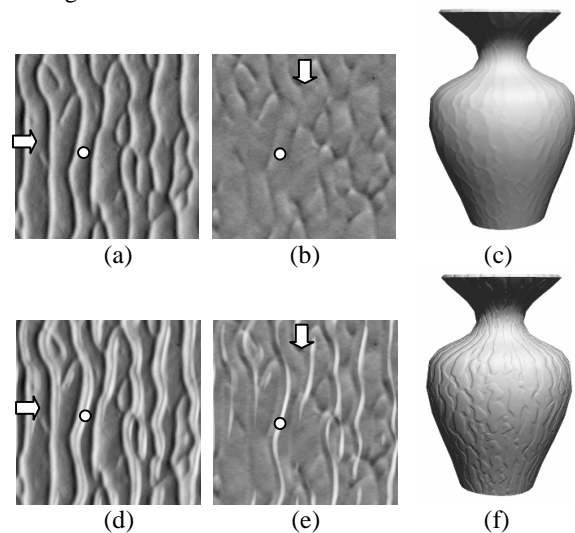


Figure 1. (a) and (b) are two images of a 3D surface texture (plaster surface) imaged under differing illumination. Block arrows show the illumination directions. (d) and (e) are the relit images produced by self-similarity based painting over eigen base images. The white dot shows the input location. (c) and (f) show the comparison of bump mapping effects produced by using the original and result surface height map, which is generated by self-similarity based editing.

The goal of this paper is to present inexpensive approaches that can edit real-world 3D surface textures using self-similarity. We extend Brooks and Dodgson’s work in [1] and propose a self-similarity based editing method that can perform painting, cloning and warping over two sets of surface representations: (1) surface height and albedo maps, and (2) eigen base images of

the sample texture. The output surface texture representations can be rendered to produce new texture images under varied illumination or viewing conditions.

Although synthesis of 3D surface texture synthesis has been investigated [13, 3], few publications are available regarding the editing of real-world 3D surface textures. The fundamental idea of our work is similar with Zelinka and Garland's surface modeling methods [9]; their approach however, operates on surfaces of vertex-based 3D models instead of 3D surface textures. The output of their method is a 3D model while ours is a new surface texture representation. In [10], Dischler et. al. mentioned that by geometrically modeling the surface height their method can change orientations, sizes, and randomness of "macrostructures" specified by the user over a texture bump map. It did not, however, consider surface reflectance properties and cannot edit other 3D surface representations such as surface normal and eigen-base images [3,13]. In contrast, our approach uses simple methods to represent the surface geometry as well as reflectance properties and can automatically edit the whole surface with minimum user input.

The rest of this paper is organized as follows. Section 2 describes the selection of 3D surface texture representations for editing. Section 3 introduces our approach on self-similarity based editing of 3D surface textures. Finally, we conclude our work in section 4.

2. Representing 3D surface textures for editing

2.1. Selection of representations

One of the characteristics of 3D surface textures is that they exhibit different appearances under varied lighting and viewing conditions. Thus, multiple images are required to describe 3D surface textures. Directly editing the original images is not practical, as calculation of self-similarity is difficult in multidimensional space. We therefore need to:

1. capture images of the sample texture that will enable us to extract a suitable representation of the 3D surface,
2. edit the surface representation to produce a new description of the required dimensions, and
3. render (or relight) the surface representation in accordance with a specified set of lighting and viewing conditions.

In the context of this paper, the term *representation map* refers to a set of images that are extracted from the original samples and can be used to generate new images under arbitrary illumination directions.

As previously discussed our aim is to develop inexpensive techniques for the editing of 3D surface textures that can provide real-time rendering when implemented in a desk top machine. This limits us to approaches that (a) use low dimensional representations, and (b) use simple or common graphics calculations such as weighted sums of base images or dot-products. We therefore choose two sets of representations: surface height and albedo maps as well as eigen-base images. The surface height and albedo maps are widely used in computer graphics applications for bump mapping and are compatible with almost every graphics software package. The eigen-base images can be used to represent surface textures with complex reflectance and has been used as a compression format of Bidirectional Texture Functions (BTFs) [3, 2, 14]. Thus, editing and rendering eigen base images can be seen as an approximation of editing BTFs.

We exploit the PhoTex texture database (www.macs.hw.ac.uk/texturelab/database) for extracting surface texture representations. The database contains many sample textures; each texture is represented using 36 images captured under a variety of different illumination conditions.

2.2. Generating surface height and albedo maps

Surface height and albedo maps are the most compact representation of Lambertian surface textures. We first use photometric stereo [4] to estimate the surface normal and albedo maps by solving an over-constrained linear system. The surface height map is then generated by using a global integration technique in Fourier space [15], which minimizes the problems associated with integrating errors. This is an efficient, non-recursive approach, but suffers problems when the gradient data has been affected by shadows.

For 3D surface texture with complex reflectance, surface height and albedo maps may be generated using many other methods [7, 6]. The height and albedo maps are used for directly editing and rendering.

2.3. Generating eigen base images

According to [13] and [8], we have elected to use three eigen base images to represent 3D surface texture with Lambertian reflectance, while more eigen base images can further capture complex reflectance components such as specularities and shadows. The eigen-base images are generated by applying Singular-Value Decomposition (SVD).

Since we have 36 images of each sample texture, we perform Singular-Value Decomposition (SVD) on the 36D sample image space. Due to the fact that

singular values derived from SVD decrease rapidly and the first few eigenvectors account for most of the information, these eigenvectors are used to approximate the original 36D sample image space [13]. Eigen base images are obtained by multiplying eigenvectors by their correspondent singular values.

The rendering process simply consists of generating linear combinations of these base images.

3. Self-similarity based editing of 3D surface textures

3.1. The framework

For 2D texture, self-similarity based editing on a particular pixel causes changes on color or intensity of all pixels that exhibit similar local neighborhoods [1]. For 3D surface texture editing, a single editing operation will result in changes on all images of the sample texture rendered under different illumination and viewing conditions. Following Brooks and Dodgson's work [1], we perform editing operation over 3D surface textures in three aspects: painting, cloning and warping. Each of these operates on the two previously selected sets of surface texture representations, namely (1) surface height and albedo maps, and (2) eigen base images.

Obviously, editing of the surface height and albedo representation operates in 2D space, while editing of eigen base images may require more dimensions if more base images are used to represent complex reflectance.

The editing operation comprises four steps:

(i) Select a pixel location on images of the sample surface texture.

(ii) Choose a proper neighborhood size and calculate a similarity map.

We use a single-scaled square neighborhood for calculating the similarity map. The size of the neighborhood should be selected in accordance with the size of all the similar elements based on human's visual perception. For each pixel location on the representation maps, the similarity between its neighborhood and the neighborhood around the input location is calculated using the Euclidian distance between the two neighborhoods. A Euclidean distance is then converted to a similarity value in the range of [0, 1] using the method described in [1]. Note that when editing operates in multi-dimensional space, the neighborhoods comprise pixel values in all representation maps at the corresponding locations. In this way, we obtain a similarity map for 3D surface texture representations.

(iii) The value at each pixel location on the representation maps is replaced by the multiplication of the user input value at the input pixel location and the similarity value obtained from the previous step.

(iv) The result representation maps are rendered at different illumination and viewing angles.

It should be noted that the similarity map calculated using eigen base images is not consistent with the one produced by using all rendered images under varying illumination directions. However, there exists implicit relation between eigen base images and surface geometry and reflectance [5]. We therefore use surface height and albedo maps to generate a similarity map, which is then used for the whole editing process of other 3D surface texture representations.

3.2. Painting on 3D surface textures

We define painting on 3D surface textures as the editing operation that changes surface geometry and reflectance as well as pixel intensities or colors of all similar areas. The direct way of editing surface geometry and reflectance is to use surface height and albedo maps as input. If we use eigen base images for editing, we may produce color or intensity changes in all images of the sample texture under varied lighting and viewing directions.

Painting on surface height and albedo maps

In order to edit surface height and albedo maps, we simply modify the self-similarity based editing algorithm proposed by Brooks et. al. in [1] so that it can take two dimensional vectors as input to generate a similarity map. Since height and albedo are heterogeneous values, they are first transformed into values in the range of [0, 255] respectively. This makes the two maps in the same scale when used for calculating the similarity map.

Obviously, this similarity map is normally different from the one produced by separately editing a height or albedo map. We use equation (1) to generate new surface height and albedo maps. It should be noted that only one threshold is employed as described in [1].

$$\begin{pmatrix} h''(x,y) \\ a''(x,y) \end{pmatrix} = \begin{pmatrix} h(x,y) \\ a(x,y) \end{pmatrix} * s(x,y) + \begin{pmatrix} h' \\ a' \end{pmatrix} * (1 - s(x,y)) \quad (1)$$

where:

(x, y) is a pixel location in the spatial coordinate

$h(x,y)$ and $a(x,y)$ are surface height and albedo maps at a pixel location (x,y) respectively

$s(x,y)$ is the similarity value at location (x,y) of the similarity map

h' and a' are the constant input height and albedo value that the user wants to change

Figure 1(f) and Figure 2 show the results produced by self-similarity based painting over surface height

and albedo maps. A visualization of a similarity map is also shown in Figure 2.

Painting on eigen base images

A single painting operation on eigen base images may change the color or intensity values of pixels exhibiting similar neighborhoods in all images of the sample rendered under varied illumination directions. Thus, a single painting on eigen base images is equivalent to painting on every relit image separately.

However, we cannot simply change eigen base images by adding a value according to the similarity and then render the result base images, as this will destroy the relationship between the base images and produce wrong results. Instead, we use equation (2) to obtain new images under different illumination directions.

$$i(x, y) = (1 - s(x, y))\mathbf{c} \cdot \mathbf{e}(x, y) + s(x, y) * V \quad (2)$$

where V is the color or pixel value to be painted at the selected location (x, y)

$\mathbf{e}(x, y)$ is the eigen vector comprises pixel values extracted from base images at location (x, y)

\mathbf{c} is the coefficient vector for rendering under the required illumination direction.

Since the eigen base images are only multiplied by a scalar $(1-s(x, y))$, the relationship between the base images is kept and only the rendering process is changed according to (2).

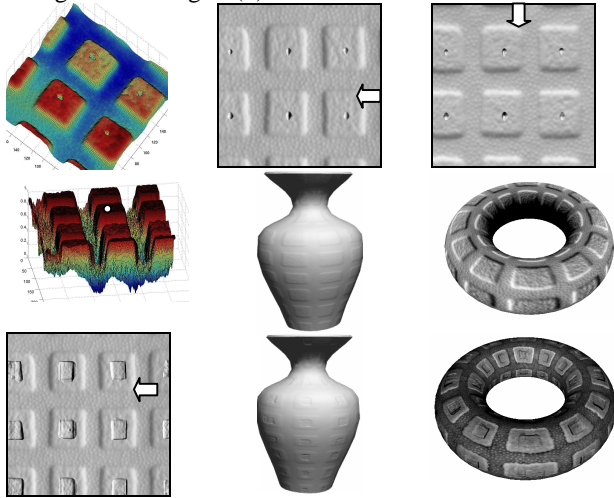


Figure 2 Results produced by self-similarity based painting on surface height and albedo maps. Top row: creating a dimple on the centre of each square. Middle row: a similarity map and images of the sample surface and its bump mapping on two 3D models. Bottom row: creating a small square in the centre of each square on the sample.

Figure 1 (d) and (e) show the result images produced by self-similarity based painting over eigen base images. We require all the “bumps” on the surface to be automatically painted with white ribbons by

selecting a location on a single “bump” (the white dot). In Figure 3, we made changes to all square bumps on the surface by painting eigen base images with two different values.

3.3. Similarity-based cloning

Similarity-based cloning of 3D surface textures allows users to mix two surface textures according to the self-similarity of one sample. We extend replicated cloning approach proposed in [1] to include surface height and albedo changes so that new 3D surface textures carrying the geometric and reflectance properties originated from the two samples can be produced. We also use eigen base images to perform cloning based on pixel intensity values to produce new texture images under different illumination directions.

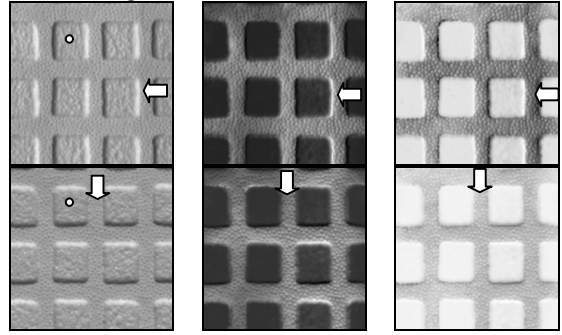


Figure 3 Painting on eigen base images using different values. Left: two original images. Middle: two relit images with the change value set to 2000 at the selected location (the white dot). Right: the change value is set to 3500. These values will be scaled to the range of [0, 255] for display.

The surface height and albedo cloning is plainly a linear combination of two sets of representation maps using the self-similarity and the global opacity values as the coefficients (i.e. h' , a' in equation (1) are replaced by height and albedo values of another surface texture). Figure 4 shows results produced by cloning texture “acd” and “ace”.

It is also a similar case for cloning eigen base images. We only need to replace V in equation (2) with another set of eigen vector and its correspondent coefficient vector. Figure 5 shows the result images produced by cloned eigen base images of two surface textures (Texture “acd” and “aar”).

3.4. Similarity-based warping of 3D surface textures

In [1], Brooks et. al. proposed to use neighborhood similarity as a measurement to expand or shrink interested area on the sample images by using a non-linear magnification technique [11]. Warping of 3D surface textures can produce the similar effect on all

images of the sample rendered under different lighting and viewing directions.

For warping over surface height and albedo maps as well as eigen base images, it is a straightforward extension of the non-linear magnification algorithm from 1D space to multi-dimensional space. Figure 7 shows rendering results of warped surface height and albedo as well as eigen base images.

Since self-similarity based warping only requires the movement of pixel positions instead of changing pixel values, it will preserve the interrelationship between some particular surface representation maps, e.g. surface normal maps. Therefore, we may use surface normal and albedo maps for magnification and shrinking. This can avoid the integration error associated with generating height data from surface normal.

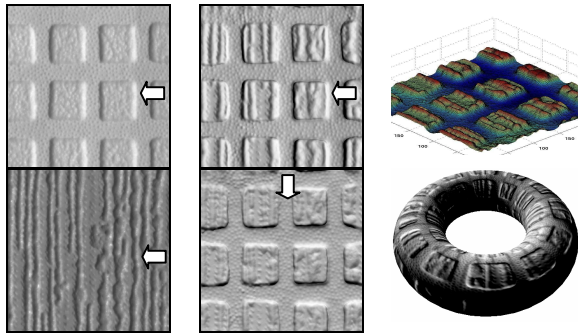


Figure 4 Results produced by cloning surface height and albedo maps. Left column: two original surface texture images. Middle column: result images illuminated from two directions. Right column: the upper image shows the plot of surface height data produced by cloning; the lower image is a bump mapping result using the cloned surface.

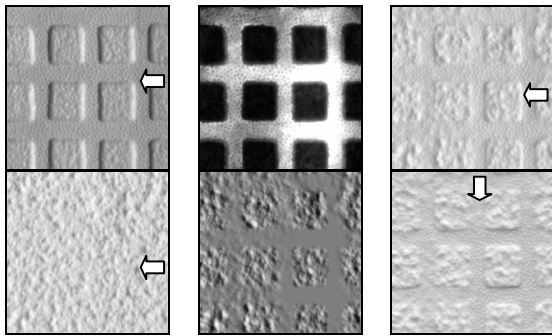


Figure 5 Results produced by cloning eigen base images. Left column: two sample texture images. Middle column: the first two result eigen base images. Right column: final relit images with different illumination directions.

However, warping over surface normal maps should be paid additional attention when borders of magnified areas comprise high frequency. The high

frequency of surface normal maps causes pixel intensity changing in all images when the sample surface is illuminated or viewed from different directions. The original algorithm may suppress the high frequency when magnifying the inside area. This results in the loss of edges in the relit images, whereas there is no such effect in the case of 2D texture warping.

In order to solve this problem, we use an additional parameter to control the suppression so that the high frequency can be preserved. The parameter, which restricts the nearest distance of two neighbor points in the magnification field, defines the number of pixels that participate in suppression. If more points can contribute to the required space, the lost of the details may be effectively controlled. Thus, an appropriate parameter will restrict the lost of detail on the borders. Figure 6 shows the comparison of the relit images using magnified surface normal without and with controlling suppression.

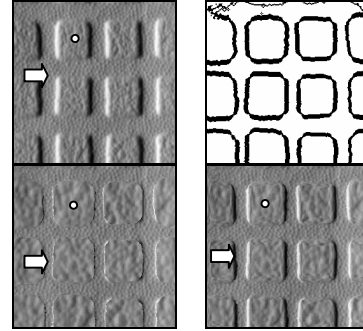


Figure 6 Comparison of the relit images using magnified surface normal without and with controlling suppression. First row: the original image and its magnification field. The black frames in the right image show the areas that the square bumps should be expanded to. Second row: relit images without (left) and with (right) controlling suppression.

4. Conclusion and discussion

In this paper we introduced inexpensive methods for editing 3D surface textures using self-similarity. The 2D texture editing method proposed by Brooks *et al.* were adapted and applied to surface height and albedo maps, as well as eigen base images. The resulting 3D surface textures can be rendered under varied illumination and viewing angles.

Although self-similarity based editing of 3D surface textures can be seen as an extension of its 2D counterpart, several problems remain unsolved and require attention. The most important is to find a suitable similarity measurement that is transform-invariant between the spaces spanned by surface representation maps and all rendered images. In

addition, our approach has not used the super-resolution synthesis techniques [1] when performing magnification operations. Thus, large amount of warping over particular 3D surface textures might cause a loss of high frequency detail. Future work may improve this by developing new methods for super-resolution synthesis of 3D surface texture representations.

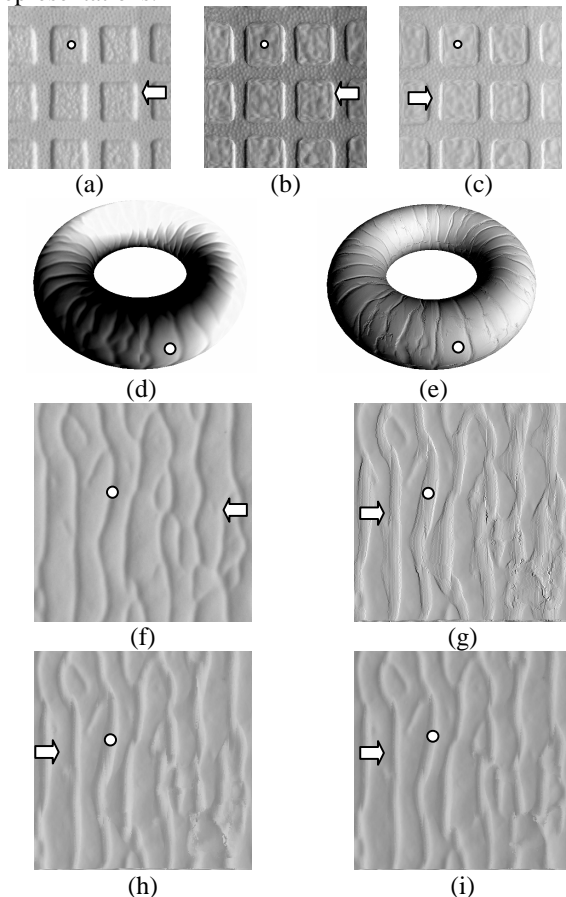


Figure 7 Results of self-similarity based warping. (b) and (c) are relit images produced by warping of eigen base images while (a) is an original image. The square bumps are magnified in (b) and (c). (e) is the result produced by rendering magnified height and albedo maps while (d) is produced using the original maps. The bottom four images show the sample image (g) and three relit images produced by warped surface height and albedo (g), surface normal and albedo (h), and eigen base images (i) respectively. Block arrows show the illumination directions and the white dots show the selected locations for editing.

References

- [1] S. Brooks and N. A. Dodgson. Self-Similarity Based Texture Editing. *ACM Transactions on Graphics*, 21(3), July 2002, pp. 653-656.
- [2] K. J. Dana, B. Van Ginneken, S. K. Nayar and J. J. Koenderink. Reflectance and Texture of Real-World Surfaces. *ACM Transactions on Graphics*, 1999 Vol. 18, No. 1, pp. 1-34.
- [3] M. L. Koudelka; S. Magda; P. N. Belhumeur; D. J. Kriegman. Acquisition, Compression, and Synthesis of Bidirectional Texture Functions. The 3rd international workshop on texture analysis and synthesis. 2003. pp59-64.
- [4] R. Woodham. Analysing images of curved surfaces. *Artificial Intelligence*, 1981. 17:117-140.
- [5] J. Dong, and M. Chantler. On the relations between four methods for representing 3D surface textures under multiple illumination directions. *Proceedings of the 2004 International Conference on Computer and Information Technology*, Sep. 2004, pp.807-812.
- [6] J. Dong, and M. Chantler. Estimating Parameters of Illumination models for the synthesis of 3D surface texture. *Proceedings of the 2004 International Conference on Computer and Information Technology*, Sep. 2004, pp716-721.
- [7] G. Kay and T. Caelli. Estimating the parameters of an illumination model using photometric stereo. *Graphical models and image processing*, Vol. 57, No. 5, September 1995.
- [8] R. Epstein, P. W. Hallinan and A. L. Yuille. 5±2 eigenimages suffice: an empirical investigation of low-dimensional lighting models. In *Proceedings of the Workshop on Physics-Based Modeling in Computer Vision*, 1995, pp. 108-116.
- [9] S. Zelinka, Similarity-Based Surface Modelling Using Geodesic Fans. In *Proceedings of the Second Eurographics Symposium on Geometry Processing* (2004), pp. 209-218.
- [10] J. Dischler and D. Ghazanfarpour,. Interactive Image-Based Modeling of Macrostructured Textures. *IEEE Computer Graphics and Applications*, 19, 1, 66-74.
- [11] A. Keahey, and E. Robertson. Nonlinear Magnification Fields. *IEEE Symposium on Information Visualization*, 51-58.
- [12] K. Perlin, and L. Velho. Live Paint: Painting With Procedural Multiscale Textures. *ACM SIGGRAPH 95*, 153-160.
- [13] J. Dong, and M. Chantler. Capture and synthesis of 3D surface texture. *International Journal of Computer Vision*. April-May 2005, Vol.62: Nos.1-2, pp177-194
- [14] Z. Zhang. Modeling geometric structure and illumination variation of a scene from real images, In *Proceedings of International Conference on Computer Vision*, Bombay, India, 1998, pp. 1041-1046.
- [15] R. T. Frankot, and R. Chellapa. A method for enforcing integrability in shape from shading algorithms. *IEEE transactions on pattern analysis and machine intelligence*, 10(4), pp. 439-451.