V^2ID: Virtual Visual Interior Design System

Zhibin Lei Yufeng Liang Weicong Wang
Bell Laboratories Rutgers University

Abstract

In this paper we propose a novel system of semantic feature extraction and retrieval for interior design and decoration application. The system, V^2ID (Virtual Visual Interior Design), uses colored texture and spatial edge layout to obtain simple information about global room environment. We address the domain specific segmentation problem in our application and the techniques for obtaining semantic features from a room environment. We also discuss heuristics for making use of these features (color, texture or shape) to retrieve objects from an existing database. The final resynthesized room environment with original scene and novel object from database is created for the purpose of animation and virtual room walkthrough.

1 V^2ID: Virtual Visual Interior Design

We present in this paper a novel electronic commerce application using multimedia content, V^2ID (i.e., Virtual Visual Interior Design). Such a system uses a image (or sketch) of an existing room environment as input and allows user to query a database of custom designs or different vendor databases for interesting/matching objects (furniture etc.). The query criterions are the visual features of individual objects (color, texture, shape) which can be computed directly from input image. The final output is a set of novel virtual scenes with retrieved objects stitched to the original room scene. We make minimum assumptions about a room environment so that novel scene composition under different camera movement can be approximately computed. The system's approach is purely visual-feature based and it focuses on the visual matching/searching effects only. It requires little modifications to the current available interior design and retail industry databases.

As shown in Fig. 1, V^2ID roughly consists of three modules: room understanding module, visual query module and image synthesize module. In order to obtain useful information about a room environment (for example simple room structure and configuration) from input image, we need to segment/partition input image properly. This is the necessary first step. We propose a novel approach of semantic feature extraction for the purpose of partitioning a room environment. We address the domain specific segmentation problem in our application and introduce techniques for obtaining semantic features from a room environment. We also discuss heuristics for making use of these features to retrieve objects from an existing database.

Traditional content based image retrieval research (CBIR) has been focused on global image content that consists of various image features: color, texture, shapes etc. Other iconic indexing approaches make use of the object spatial relationship as well as image features to compute the similarity between two images. While image features capture the global content of an image, they fail to characterize the individual object or scene structure. In many applications, there is a need to represent a single object. Current systems usually require user's input to help locating the object in the image. To go beyond the simple global image features to individual object features requires image understanding (or image segmentation) in general. Unfortunately domain free segmentation problem has not been solved yet and it still presents a challenge for many applications.

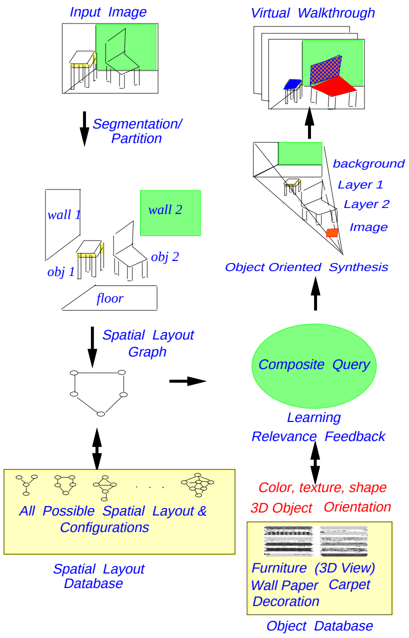

Figure 1: V^2ID: Virtual Visual Interior Design.

Our visual query module uses object visual features for content-based object/image retrieval. In our application, images of individual object (for example furniture) have already been captured and stored in the database and its feature vectors have been precomputed and stored as key index as well. However, for the query input image, we still need to partition or segment the image in order to locate the interesting objects in the scene or estimate the room spatial layout.

An image of the interior room scene consists of several different layers. Each layer serves a different purpose for V^2ID. For example, background layer is the least important layer if user is interested in purchasing new furniture. But it becomes very important when user's goal is to change or buy new curtain or carpet. We represent a room scene with following layers: background, object 1, ..., object n, image plane. Background consists of ceil, walls, floor and whatever object(s) that are attached to them and do not belong to the object n layers. Object n layers are different objects (sofa, armchair, and other furnitures that are of interest to the user). Image plane layer is the layer where the whole room scene gets mapped onto. With such a muti-layered representation, we can retrieve similar/matching objects from vendor databases and recompose novel scene based on this knowledge of room environment. In the following sections we discuss various modules employed in V^2ID.

2 Room Environment Understanding

There are three different steps in our system for semantic room feature extraction. The first step is the object/background segmentation or scene partitioning. Our goal in this module is to locate as many background planes as possible, i.e., we need to find the number of ceil (0 or 1), number of walls (0, 1, 2, 3, 4), number of floor (0 or 1) that exist in a scene.

Our second step is to find the approximate spatial layout of a scene. After the detection of various homogeneous texture/color regions, edges in the texture region will be removed and only the directional edges of walls, ceil and floor will be used to obtain the spatial layout information. Hough transform is used in a much cleaned room scene to find structural straight lines or line segments, which can be used to locate the vanish point – an important feature for room scene understanding. We store all the possible spatial layout and configurations in our database and this knowledge will help to find the location of background planes (walls, floor, and ceil).

The last step uses the segmentation result and spatial layout information to locate potential interesting objects in the scene and create object n layers. Simple heuristics can be used in this stage. For example, curtains and windows are objects attached to wall. Carpet is on the floor and furniture is in general "blended" into wall background. Object specific geometric shape will be used to detect object n layers too. We classify different kind of furnitures based on their geometric appearance. Object shapes are represented as implicit algebraic curves [1] and their algebraic invariants are used as feature vectors. In the current implementation, we still need user input to finalize the object layer results. We compute object specific features (color histogram, texture, shape) based on the final result. The object retrieval is then done based on individual object features instead of the whole image.

2.1 Color texture quantization

We use a texture segmentation algorithm to locate homogeneous texture/color regions in the image. The image feature we use combines the spectral information with the spatial information of an image and represent them in an unified way.

The images in our database are RGB images with each color channel quantized in 8 bits. First, we need to re-quantize the colors to K representative colors. The problem of selecting the K representative colors from N total colors is a specific instance of the more general problem of vector quantization(VQ). Often, the Linde-Buzo-Gray (LBG) algorithm can be used to iteratively achieve a local minimum with respect to the MSE criterion. Due to the computational high cost of the VQ algorithms, a simple heuristic that has been used for color quantization is the popularity algorithm, which works by forming a 3-D histogram of the true color image colors and assigning the K most frequently occurring colors in the histogram as the representative colors.

The modes of a continuous probability distribution are the local maxima of that distribution [3]. They represent the most probable values of the random variable.

In this paper, we use a combination of mean-shift mode seeking and Partition Around Medoids(PAM) to do the color quantization. In order to ensure the isotropy of the feature space, the La^*b^* uniform color space is used, in which the Euclidean distance is the perceived color difference. First, the mean-shift mode seeking algorithm is used to find all the modes in the feature space. Once all the convergent points of the data are found, we use the PAM algorithm on the convergent points and find the clusters. The centroids of the clusters are numbered and used to represent the pixels in the cluster. In our application, images are quantized to 16 colors.

2.2 Texture feature and segmentation

Grey-scale co-occurrency matrix is used on the re-quantized image to extract the texture information from the image. If the input image $I(i, j)$ is quantized to N levels, the spatial grey-scale co-occurrence matrix $P(d, \theta)$ is an $N \times N$ matrix whose (m, n)th element $P_d(m, n)$ is defined as the number of times the gre-scale m and n occur separated by d pixels at an angle θ within the image.

$$P_{d,\theta}(m, n) = \{(i, j), (k, l) \in S | I(i, j) = m \ \ , \ \ I(k, l) = n\} \tag{1}$$

where S is the set of all pairs of pixels in the given spatial relation.

Since it is desired to have the feature invariant under rotations, a co-occurrence matrix will be utilized averaging over all possible θ and represented by $P_d(m, n)$. In our application, the co-occurrency matrix is calculated on $0°, 45°, 90°, 135°$ and then summed up.

$P_d(m, n) = [P_{d,0°}(m, n) + P_{d,45°}(m, n) + P_{d,90°}(m, n) + P_{d,135°}(m, n)]/4$

We define the distance between two co-occurrence matrices C_1 and C_2 as:

$$D = \sum_{i=1}^{N} \sum_{j=1}^{N} |C_1(i, j) - C_2(i, j)| \tag{2}$$

The image is divided into 16×16 blocks and the co-occurrence matrix is calculated for each block. PAM algorithm is then used to cluster the similar matrices into a uniform area. One example of the color texture segmentation algorithm is shown in Fig. 2.

3 Visual Search Engine

Color and texture have been studied extensively in the previous CBIR research. Our current implementation of color primitives involves computation and matching in the perceptual color space [2]. Color texture is extremely important in interior design applications. Not only do we need to address the illumination and viewing direction invariants of color texture features, we also need to take into consideration the psychological and perceptual effects of color and texture. Actually, the definition of texture itself for matching patterns in interior design is not well defined. Our current work on texture primitives is geared towards a general representation and matching for texture as an image feature. We chose a multi-level illumination invariant color feature vector based on the correlation function of three spectral channels of the input color texture. (See [4] for details)

For object shape structure modeling, we use implicit algebraic curves. These curves are natural extensions of straight lines, ellipses, conic sections etc and are very useful for modeling natural man-made object shapes.

$$f(x,y) = \sum_{i,j \geq 0; i+j \leq d} a_{ij} x^i y^j$$

Based on this representation, shape similarity matching metrics can be computed as [1]:

$$\begin{aligned} dist(Z_1, Z_2) &= \| f_1 - f_2 \|_{Z_1 \cup Z_2} \\ &\equiv \tfrac{1}{N_1} \textstyle\sum_{(x,y) \in Z_1} (f_1(x,y) - f_2(x,y))^2 + \\ &\tfrac{1}{N_2} \textstyle\sum_{(x,y) \in Z_2} (f_1(x,y) - f_2(x,y))^2 \end{aligned} \tag{3}$$

where data sets Z_1 and Z_2 containing N_1 and N_2 points, respectively. The algebraic curve models for them are $\{(x,y) : f_1(x,y) = \sum_{0 \leq i,j; i+j \leq 4} a_{1ij} x^i y^j = 0\}$ and $\{(x,y) : f_2(x,y) = \sum_{0 \leq i,j; i+j \leq 4} a_{2ij} x^i y^j = 0\}$ (for simplicity we use 4th degree curves here. The definition can be easily extended to any degree).

Without loss of generality, we combine Z_1 and Z_2 into a single data set $Z = Z_1 \cup Z_2$ which has $N_{1,2} = N_1 + N_2$ point s and write $dist(Z_1, Z_2)$ as

$$\begin{aligned} dist(Z_1, Z_2) &= \tfrac{1}{N_{1,2}} \textstyle\sum_{(x,y) \in Z} [f_1(x,y) - f_2(x,y)]^2 \\ &= \tfrac{1}{N_{1,2}} \textstyle\sum_{(x,y) \in Z} [\sum_{0 \leq i,j; i+j \leq 4} (a_{1ij} - a_{2ij}) x^i y^j \ \ ]^2 \end{aligned} \tag{4}$$

Denote $\mathbf{a} = (a_{00} \ a_{10} \ a_{01} \ a_{20} \ ... \ a_{04})^t_{15}$, $\mathbf{X} = (1 \ x \ y \ x^2 \ ... \ y^4)^t_{15}$ and denote the elements of Z as $\{(x_n, y_n) : 1 \leq n \leq N_{1,2}\}$.Then eqn. 4 can be rewritten as

$$dist(Z_1, Z_2) = \tfrac{1}{N_{1,2}} \textstyle\sum_{n=1}^{N_{1,2}} (\Delta \mathbf{a}^t \mathbf{X}_n)^2 = \tfrac{1}{N_{1,2}} \sum_{n=1}^{N_{1,2}} (\Delta \mathbf{a}^t \mathbf{X}_n)(\mathbf{X}_n^t \Delta \mathbf{a}) = \tfrac{1}{N_{1,2}} \sum_{n=1}^{N_{1,2}} \Delta \mathbf{a}^t (\mathbf{X}_n \mathbf{X}_n^t) \Delta \mathbf{a} = \Delta \mathbf{a}^t (\tfrac{1}{N_{1,2}} \sum_{n=1}^{N_{1,2}} \mathbf{X}_n \mathbf{X}_n^t) \Delta \mathbf{a} \equiv \Delta \mathbf{a}^t M \Delta \mathbf{a} \geq 0$$

The final object/image matching score is a weighted sum of color histogram distance, texture distance, and shape similarity distance.

4 Image Synthesize and Virtual Walk Through

Once the semantic features are established from the previous processing stages, image synthesize techniques can be used to resyhtehsize novel images based upon original scene and new objects retrieved from database and generate animation sequence of images of the same room seen from different position and view angles thus provide a on-line virtual *walk through* of the room.

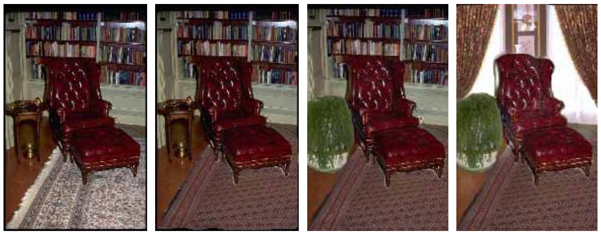

Figure 2: Image synthesize with new objects from database.

The general projective map of 3D world to 2D image scene is

$$\begin{pmatrix} xr \\ yr \\ r \end{pmatrix} = \begin{pmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & m_9 \end{pmatrix} * \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

where (X, Y, Z) is 3D coordinate and (x, y) is image coordinate. In order to provide a correspondence map of 3D to 2D and 2D image to 2D image, 9 parameters are needed. To simplify the process, we assume the perspective transformation (Fig. 4) and thus reduce the total number of unknowns. This assumption is valid especially for the 2D like objects as seen in our application (carpets, wall paper etc ...). The estimation of transformation parameters is by using the structural straight lines or line segments from the image after the coarse level image partition module. We want to minimize the sum of the squared intensity errors from many different sets of line segments.

$$E = \sum (I'(i', j') - I(i,j))^2 = \sum e^2$$

Fig. 2 shows the final results of image composition based on new objects from database. (The left image is the original input image.)

To walk through the new scene and generate the animation sequence, we need to use the background layout information as well as collect virtual camera information such as position, direction and focal length. We use a rather simplified approach for camera calibration since in our application we do not have very rigid constraints and we can allow certain range of variations.

In our system, three types of spatial layout of the background is supported, which applies to most cases of interior scene. The possible spatial layouts and to which of the three type they belong are illustrated in Fig. 3.

We need to estimate the vanish point and calibrate the virtual camera for each type of background. Assume the initial camera position and the focal length of the camera to be $\vec{V_c} = \{X_c, Y_c, Z_c\}$ and f respectively.

Take the spatial layout type 1, which is the most often seen layout in the interior design application, for

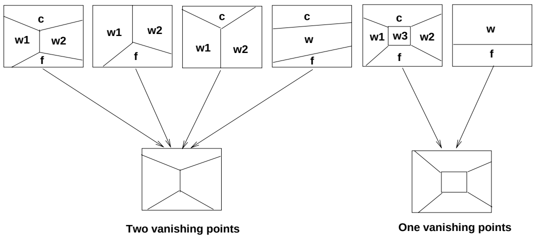

Figure 3: Possible spatial layout types.

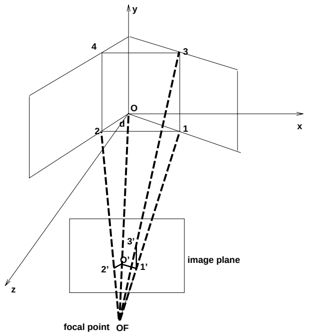

Figure 4: Perspective transformation.

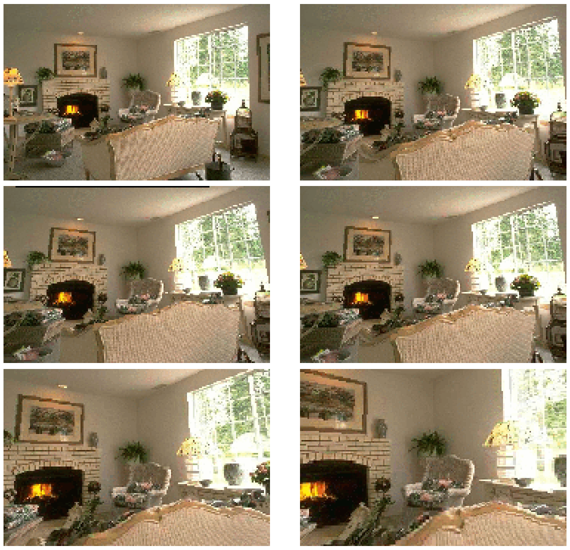

Figure 5: Virtual scene animation by changing camera location.

example, the coordinate system is set up as shown in the Fig. 4. The image plane is assumed to be perpendicular to the z-axis. A rectangle 1234 is specified with an arbitrary unit distance d to the origin O. The slope of $O1$ is assumed to be k.We have proved that $\vec{V_c} = \{X_c, Y_c, Z_c\}$,$f$ and k can all be represented as a function of d. Consider point 1 with the perspective transform point $1^{'}$ on the image plane

$$\vec{O1'} = O\vec{O}F + t(\vec{O1} - O\vec{O}F) \qquad 0 < t < 1 \qquad (5)$$

With the known relation

$$z_{1'} = z_{OF} - f \qquad (6)$$

we can solve the t in equation 5. The points 2,3 are used in the same way to combine with the point 1, which gives a equation set with 6 equations and 6 unkonwns($X_c, Y_c, Z_c, f, k, h = ||\vec{12}||$). By solving this equation set, the virtual camera can be calibrated.

Once the virtual camera and the foreground objects have been calibrated, a virtual walk through can be achieved by specifying a sequence of actions of the virtual camera and rendering the video sequence based on these virtual camera actions. Fig. 5 shows a sequence of novel images of a room scene.

5 System Architecture and Future Work

We select a client-server architecture for our system. In order to be able to access the system over the intranet or internet, we design the front end user interface using the HTML and Java technology. Most user interaction (including input image taken and manipulation) can be done very efficiently using Java on the fly at the client side. However, the vendor database still resides on a remote server side. We have pre-computed all the image features for the database images and store them along side with image itself. The client-server communication is through CGI (Common Gateway Interface). We are currently working on developing the system to use CORBA technology. There are many design issues that we are facing in implementing such a system. For example, segmentation and animation can be done both at the server and client side. It is necessary to consider computer power, network bandwidth as well as user satisfaction and interaction in order to make an intelligent decision for the many tradeoffs involved. There are many other issues need to be addressed. For example, We want to have as little user input for room understanding as possible. A machine learning of visual query engine is also very important for the user satisfaction. Another area needs further research is how to bring the CAD model and available images together for such a image based visual application.

References

[1] Z. Lei, T. Tasdizen, and D.B. Cooper. "PIMs and Invariant Parts for Shape Recognition," *Proceedings of International Conference on Computer Vision,* Bombay, India, January 1998.

[2] Z. Lei, S. K. Ganapathy and R.J. Safranek. "MIRACLE: Multimedia Information Retrieval by Analysing Content and Learning from Examples," *Proceedings of Fourth IEEE Workshop on Applications of Computer Vision,* Oct. 1998, Princeton, NJ.

[3] Y. Liang, Z. Lei, S.K. Ganapathy and J. Wilder. "Multi-resolution Colored Texture Detection and Analysis," *Proceedings of First International Workshop on Computer Vision, Pattern Recognition and Image Processing,* Oct. 1998, Research Triangle Park, NC.

[4] Z. Lei and Y. Liang. "Multi-Layered Semantic Feature Extraction for Interior Environment Understanding and Retrieval," *IS&T/SPIE's 11th Annual Symposium, Multimedia Processing and Applications, Storage and Retrieval for Image and Video Databases VII,* Jan. 1999, San Jose, CA.