Generation of N-Parametric Appearance-Based Models Through Non-uniform Sampling

Luis Carlos Altamirano[1], Leopoldo Altamirano Robles[2], and Matías Alvarado[1]

[1] Instituto Mexicano del Petróleo, PIMAyC, Eje Central Lázaro Cárdenas 152,
San Bartolo Atepehuacán, C.P. 07730, México, DF
(laltamir,matiasa)@imp.mx

[2] Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro No. 1, C.P. 7200, Puebla, Pue., México
robles@inaoep.mx

Abstract. In this work, a generalization of non-uniform sampling technique to construct appearance-based models is proposed. This technique analyses the object appearance defined by several parameters of variability, determining how many and which images are required to model appearance, with a given precision ε. Throughout non-uniform sampling, we obtain a guideline to spend less time on model construction and to diminish storage, when pose estimation no matters. The proposed technique is based on a scheme of N-linear interpolation and SSD (Sum-of-Squared-Difference) distance, and it is used in conjunction with the eigenspaces method for object recognition. Experimental results showing the advantages are exposed.

1 Introduction

Appearance-based approaches were proposed as an alternative to the geometrical ones for object recognition. They use the image data, i.e., the pixel intensities, directly without previous segmentation process. Several appearance-based methods have been proposed in the literature [1], [2], [3], [4], [5].

So far in the current literature, efforts to determine characteristic views under illumination changes have been made [6, 7]. However, if viewing position (instead of illumination) is changing, no characterization of the image set is done. To work on viewing position changes, diverse techniques on image synthesis have been introduced [8, 9]. However, there are several differences between the appearance-based approach and image synthesis [10]. Usually, appearance-based approaches require a big quantity of images in order to build object models. For this reason, several techniques for modeling objects with fewer views have been introduced, for example, aspect graphs [11] or eigenspaces [1]. However, calculating these characteristic views is an expensive computational process. Recently, some works to determine how many and which images are necessary to model an object have been presented. However, their application is restricted to specific objects as faces [12], or they only work with the object shape [13].

Non-uniform sampling was introduced in [14] as an alternative for reducing the quantity of necessary images for object modeling and object recognition. Experimen-

A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 132–139, 2004.

tal results showed that a reduction in the quantity of required views is possible, if the behavior of the object appearance is taken into count, and no pose estimation is necessary. If pose estimation is required, reduction is also possible but it could be not significant.

However, non-uniform sampling cannot model objects under more than one parameter. For this reason, in this work, a generalization to N parameters of the basic technique proposed in [14] is developed. This generalized technique allows determining the strictly necessary images for modeling any object, under N parameters and for a precision ε. Image election is guided by N-linear interpolation and SSD (Sum-of-Squared-Difference) criterion. We show how this technique can be jointed to eigenspaces one to support object recognition.

2 Preliminaries

2.1 Eigenspaces Overview

By using Principal Component Analysis, the eigenspaces technique [1] comprises a set **I** of training images into a compact representation which can be used for object recognition. This compression process is computationally intensive, and its complexity depends upon the size of **I** (if the size of **I** is smaller than the number of pixels that constitutes the images; otherwise, depends upon the number of pixels). For this reason, reducing the size of **I** is desired.

2.2 Interpolation and SSD Distance

Definition 1: A parameterized surface is a function ϕ: $D \subset R^N \rightarrow R^M$, where D is any domain in R^N [15]. Notice that a parameterized surface is a generalization of a parameterized trajectory, because if N=1 the surface becomes a trajectory.

Definition 2: Let ϕ_1: $D \subset R^N \rightarrow R^M$ be a parameterized surface. Let $C=\{(x_1,y_1), (x_2,y_2),\ldots,(x_m,y_m)\}$ be such that $\phi_1(x_i)=y_i$, for i=1,...,m. A parameterized surface ϕ_2: $D \subset R^N \rightarrow R^M$ interpolates $\phi_1(x)$ on points x_i in C, if $\phi_2(x_i)=y_i$, for i=1,...,m. Because $\phi_2(x)$ is not determined uniquely, for an ε given, $\phi_2(x)$ is usually selected such that $|\phi_1(x) - \phi_2(x)| < \varepsilon$, for all x in D. Such ε is named the associate interpolation error.

Sometimes, it is impossible to find a parameterized surface $\phi_2(x)$ such that it meets the error criterion inside the interval. In this case, the interpolation problem of $\phi_1(x)$ is changed for the $\phi_1(x)$ *piecewise interpolation problem*: it is necessary to find a set of m-1 parameterized surfaces $\phi_2(x)$, $\phi_3(x)$,..., $\phi_m(x)$, such that interpolate $\phi_1(x)$, respectively, in a partition of D, within the tolerance ε.

Definition 3: Let $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$ be two vectors in R^n. The SSD distance (Sum-of-Squared-Difference) between X and Y is defined by:

$$\| X - Y \|^2 = \sum_{1}^{n} (x_i - y_i)^2$$

A detailed explanation of the properties of SSD distance on appearance-based models can be founded in [16].

3 Generalized Image Acquisition

In order to build the appearance-based model of an object, it should be in the turntable's center, and rotates in front of the camera. The camera is in a fixed position respect to turntable, and it cannot move. Finally, the illumination source is fixed with respect to the camera and the turntable. Under these conditions, images taken around the object belong to a trajectory, which is parameterized for the angle θ of the object rotation in front of the sensor; this trajectory is called in the literature the trajectory determined by object appearance, or simply, the object appearance. The object model is constructed by sampling this trajectory. Notice that the sampling can be uniform or non-uniform, but uniform sampling requires a lot of space and time for building models. Non-uniform sampling was introduced for reducing space and time requirements. However, the object appearance would be defined by several parameters. For example, if we let illumination source moves or camera moves. In this case, the technique exposed in [15] is no more applicable.

Then, we extend the previous non-uniform sampling technique to work with several parameters. Some parameters that we could consider are: turntable angle, camera elevation angle, position of illumination source, position of any part of an articulated object, etc. To experiment with the proposed technique, in this work we will use only two parameters of those: turntable angle and camera elevation angle, but we emphasize that the same technique can work with whichever number of parameters, as explained below.

3.1 Generalized Uniform Image Acquisition

If we use uniform sampling to acquire images for building the model of an object defined by N parameters, we need sampling on the whole workspace determined by N parameters. One image parameterized with N parameters, i.e., $I(p_1, p_2, \ldots, p_n)$, is obtained for each point of this sampling. Whenever all images acquired in this manner are represented as n×m matrixes, and matrix's columns are stacked to form vectors $v \in R^{nm}$, looking at Definition 1, we can see that such vectors belong to a surface $\phi: D \subset R^N \rightarrow R^{nm}$ defined by N parameters. This parameterized surface is called in the literature the surface determined by object appearance, or simply, object appearance. However, as mentioned before, uniform sampling implies to use a big quantity of images and spend too much computing time. To avoid these problems, the usage of non-uniform sampling is proposed.

3.2 Non-uniform Image Acquisition

The basic non-uniform sampling technique is based on the observation that object appearance can be approximated by means of piecewise linear interpolation. In this work, the use of linear interpolation to approximate the object's appearance is also proposed. Of course, because we are considering N parameters, we need to approximate a parameterized surface (see Definition 2) instead of a trajectory. For this reason, we propose to approximate the parameterized surface with N-linear interpolation, instead of linear interpolation.

N-linear interpolation is the straight generalization of linear interpolation, because if N=1, we obtain linear interpolation (1-linear interpolation). Linear interpolation interpolates between points b_0 and b_1 by means of the straight line (simplest curve) between them:

$$g(t_1) = (1-t_1)b_0 + t_1 b_1 \quad ; \quad 0 \le t_1 \le 1 \; . \tag{1}$$

In bi-dimensional case, bilinear interpolation interpolates between four points b_{00}, b_{01}, b_{10} and b_{11} by means of the simplest surface between these points (a hyperbolic paraboloid); it is obtained by the following expressions [17]:

$$\begin{aligned} B_{00}^{10} &= (1-t_1)b_{00} + t_1 b_{10} \quad ; \quad 0 \le t_1 \le 1 \\ B_{01}^{11} &= (1-t_1)b_{01} + t_1 b_{11} \quad ; \quad 0 \le t_1 \le 1 \\ g(t_1, t_2) &= (1-t_2) B_{00}^{10} + t_2 B_{01}^{11} \quad ; \; 0 \le t_1 \le 1; 0 \le t_2 \le 1 \end{aligned} \tag{2}$$

We can observe that it is possible to obtain the generalized expression for N-linear interpolation between 2^N points. Those expressions are obtained easly by means of an algorithmic procedure [17]. So, we do not show it here.

As we can observe, N-linear interpolation can be used to approximate parameterized surfaces inside an error ε, according to Definition 2. So, N-linear interpolation can be used in the basic non-uniform sampling technique for obtaining a generalized technique.

To clarify these ideas, in Algorithm 1 the generalized algorithm to determine the necessary images to build the model of an object is given; this is defined by several parameters. The algorithm uses N-lineal interpolation to determine when a new image should be added as a part of the object model. A new image is added if the current model cannot interpolate this image appropriately (inside the error ε). Notice that the technique just guarantees good approximation on the middle point of interval analyzed. Some technical details are beyond the scope of this work, for example, for acquiring an image just once, we need to have special data structures and extra code.

An important aspect to be considered occurs in step 6 of the Algorithm 1. Here, split can be done homogeneously or heterogeneously. Homogeneously split is desired if all parameters are defined initially with the same bounds and the mechanical system has the same resolution for all parameters. If it is not the case, heterogeneous split is desired and it is necessary to prove the correct interpolation in each parameter. It can be done with N linear interpolations, one for each parameter. Throughout the following experiments, heterogeneous split is used.

```
Algorithm 1:
   0. Set $\varepsilon$ to the desired precision. Set low and high
      bounds for each one of N parameters.
   1. Acquire $2^N$ images, one for each vertex of the zone
      to analyze.
```

```
 2. Compute middle point of the zone to analyze (Lm).
 3. Acquire image for position Lm.
 4. Interpolate N-linearly between $2^N$ images acquired
    in step 3 using expression(1), (2), or the appro-
    priate expression to obtain: $g(t_1, t_2, ..., t_N)$; $0 \le t_i \le 1$.
 5. If it is possible to interpolate image Lm with
    g(0.5, 0.5, …, 0.5) within error $\varepsilon$, using sum-of-
    squared-difference (Definition 3) as criterion to
    compare Lm and g(0.5, 0.5, …, 0.5), i.e.,
    $|| Lm - g(0.5, 0.5, …, 0.5) ||^2 \le \varepsilon$
    then go to step 8.
 6. Split zone in $2^N$ sub-zones. Push in the stack
    these $2^N$ sub-zones.
 7. Go to 9.
 8. Keep images acquired in step 3, as necessary im-
    ages to build object's model.
 9. If stack is not empty, then pop a zone to analyze
    and go to step 2.
10. END.
```

4 Experiments

The software system that acquires the necessary images for modeling the object appearance defined by several parameters was developed. The software was coupled to a mechanical system (turntable) that rotates the object. Additionally, for this work, the mechanical system was able to elevate and descend the camera over turntable plane. The software system determined how many and which images are required to satisfy the precision criterion ε, for each object studied, respectively, for 1 (rotation) and 2 (rotation, elevation) parameters. For this work, we use the proposed generalized technique to analyze the objects showed in Fig. 1.

4.1 One Parameter

In this case, just one parameter (rotation) was used. Results about this case were reported in [14] over the Columbia Object Image Library [16], and the number of images founded by the proposed algorithm represented a significant reduction with respect to traditional approaches. Also, results about object recognition rate were typically very good because model precision was improved.

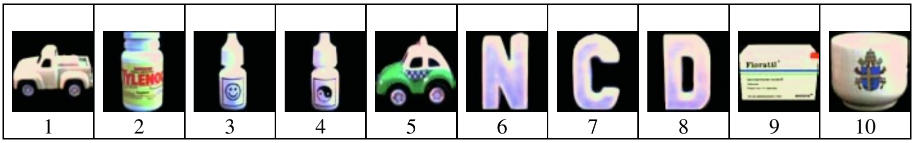

Fig. 1. Objects used to test the proposed technique

4.2 Generalized Case

In this case, the mechanical system was enabled with the capacity of ascending and descending the camera over the turntable plane. This extra capacity implies that the software system was required to analyze the appearance of the objects on 2 parameters: turntable rotation and camera elevation. In these experiments, the turntable rotation covered all 360° but camera elevation was limited to 0°-20° range.

Table 1. Results obtained to objects in Fig. 1, applying the proposed technique using two parameters (turntable rotation and camera elevation)

		OBJECT									
		1	2	3	4	5	6	7	8	9	10
PRECISION	ε = 2000	+	+	+	+	+	+	+	+	+	+
	ε = 2500	+	+	8	8	+	+	+	+	+	+
	ε = 3000	+	+	8	8	+	+	+	+	+	+
	ε = 3500	+	15	6	6	+	92	64	81	+	+
	ε = 4000	+	8	6	6	+	64	50	64	+	+
	ε = 4500	+	8	6	6	+	55	46	50	+	+
	ε = 5000	+	6	6	6	69	46	37	43	+	+
	ε = 5500	+	6	6	6	63	40	30	35	+	6
	ε = 6000	+	6	6	6	55	32	29	32	48	6
	ε = 6500	+	6	6	6	41	29	29	31	34	6
	ε = 7000	54	6	6	6	38	24	24	27	26	6
	ε = 7500	44	6	6	6	29	24	21	24	23	6
	ε = 8000	38	6	6	6	22	21	15	17	21	6

We applied the generalized technique over 50 objects, but we just documented results for objects in Fig. 1. We obtain the results shown in Table 1. The algorithm determined the strictly necessary images for modeling each object to specified precision ε, shown in column 1 of Table 1. In these experiments, we restricted our system to rotate or descend/elevate the camera more than 5° between consecutive images. For this reason, in Table 1, a + symbol means that to obtain the required precision ε, the system needed to rotate, descend or elevate the camera less than 5° between consecutive images (in the literature are reported precisions between 10° - 12°, typically). Notice the important reduction on the image quantity respect to a uniform sampling (36x3=108 images, 10° between consecutive images).

Finally, the precision of the models generated with the proposed technique was tested, by comparing the interpolated images obtained from the model, with real images obtained on corresponding positions. We show results in Fig. 2 for the model of object 1 in Fig. 1. The SSD between real and synthetic images is usually less than ε, showing that the estimated model is typically a good approximation of the object appearance (see definition 3), as reported in [14].

5 Non-uniform Sampling and Eigenspaces

The eigenspaces technique can be faster if it uses the images determined by the proposed algorithm as its training set of images, instead of the larger number used cur-

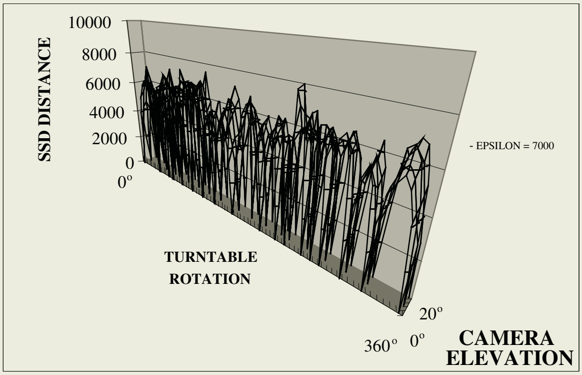

Fig. 2. Testing a 2-parametric model

rently, and it can be safety done, because both measure of model precision and criterion to select the strictly necessary object images for a given precision ε, are preserved in eigenspace [14]. Notice that the generalized proposed technique should be used in applications where pose estimation no matters, because the computation of eigenspaces will be faster using non-uniform sampling.

6 Conclusions

Non-uniform sampling to build N-parametric appearance-based models was presented. The method determines the strictly necessary images to capture the object appearance defined by N parameters, within a precision ε. With this technique, more complete object models are obtained than 1 parameter, and reduction of image quantity to build object models is achieved by eliminating unnecessary images.

References

1. Murase, H., Nayar, S.K.: Visual learning and recognition of 3-D objects from appearance. International Journal of Computer Vision. Vol. 14 No. 1 (1995) 5-24
2. Nelson, R. C., Selinger, A.: Experiments on (Intelligent) Brute Force Methods for Appearance-Based Object Recognition. DARPA Image Unders. Worksh. (1997) 1197-1205
3. Pauli, J., Benkwitz, M., Sommer, G.: RBF Networks Appearance-Based Object Detection. Proceedings of ICANN, Paris, Vol. 1 (1995) 359-364
4. Poggio, T., Beymer, D.: Regularization Networks for Visual Learning. In Early Visual Learning, Oxford University Press (1996)
5. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object representation. In Early Visual Learning, Oxford University Press (1996)
6. Belhumeur, P. N., Kriegman, D. J.: What is the Set of Images of an Object Under All Possible Ilumination Conditions?. Int. J. C. Vision. Vol. 28, Issue 3 (1998) 245-260
7. Epstein, R., Hallinan, P., Yuille, A.: 5±2 Eigenimages suffice: An empirical investigation of low-dimensional lighting models. Proc. IEEE Worksh. on physics-based modeling in computer vision (1995)

8. Glassner, A.S.: Principles of digital image synthesis. Morgan-Kaufmann Pub. (2000)
9. Seitz, S.M., Dyer, C.R.: Photorealistic Scene Reconstruction by Voxel Coloring. Int. Journal of Computer Vision, 35(2) (1999)
10. Epstein, R., Yuille, A.L., Belhumeur, P.N.: Learning Object Representations from Lighting Variations. Object Representation in Computer Vision Vol. II. In Proceedings of ECCV'96 International Workshop, Cambridge, U.K. (1996)
11. Koenderink, J.J., Van Doorn, A.J.: The internal representation of solid shape with respect to vision. Biological Cybernetics, Vol. 32, (1979)
12. Cootes, T.F., Wheeler, G.V., Walker, K.N., Taylor, C.J.: Coupled-View Active Appearance Models. The Eleventh British Mach. Vis. Conf. U. of Bristol (2000)
13. Mokhtarian, F., and Abbasi, S.: Automatic Selection of Optimal Views in Multi-view Object Recognition. The Eleventh British Mach. Vis. Conf., U. of Bristol (2000)
14. Altamirano, L.C., Altamirano, L., Alvarado, M.: Non-Uniform Sampling For Improved Appearance-Based Models. Pattern Rec. Letters, V. 24, Issue 1-3, (2003) 529-543
15. Marsden, J.E., Tromba, A.J.: Vector Calculus. W.H. Freeman and Co., Eds. (1976)
16. Nayar, S., Murase, H., Nene, S.: Parametric Appearance Representation. In Early Visual Learning, New York Oxford, Oxford University Press (1996)
17. Farin, G.: Curves and Surfaces for Computer Aided Geometric Design. A. Press, Inc, (1988)