

# 计算机辅助动画系统 (GKDCAS)

张 骏 李 彦 万良君 唐振初 胡衡江

(电教中心)

**摘 要** GKDCAS 是一个基于微型计算机环境, 使用 C 语言设计的一个实用动画制作系统。本文阐述了 GKDCAS 的软件结构与设计方法, 介绍了它的功能, 并给出了一个用该系统制作动画的实例。

**关键词** 计算机动画, 交互式计算机图形, 数据库, 键帧动画

**分类号** TP399

自从 1963 年美国制作出第一部计算机动画电影以来, 计算机动画技术得到了迅速发展。今天, 在巨型机、中小型机、专用图形硬件和显示设备 (如 SGI IRIS 4D 系统) 等强支持环境下, 计算机可以产生出许多奇妙的动画来, 如美国伊马克斯系统公司用 40 名程序员花 5 年时间用巨型计算机制作了一部名叫“太阳的回波”的影片, 该片获得了惊人的三维效果。不过, 这些系统费用都大大超出了大多数用户的承受能力。

我们自行研制的新型实用计算机动画系统 GKDCAS 是在通用微型计算机系统上设计的。目前除三维实体造型之外, 整个系统已完成。它将向用户提供有力的动画制作手段, 大大提高传统动画制作的效率和质量, 并降低制作成本。

## 1 GKDCAS 系统描述

如图 1 所示, GKDCAS 的硬件环境包括: COMPAQ-386 微型计

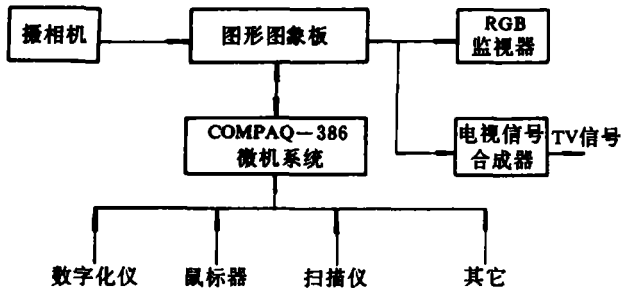


图 1 GKDCAS 硬件环境

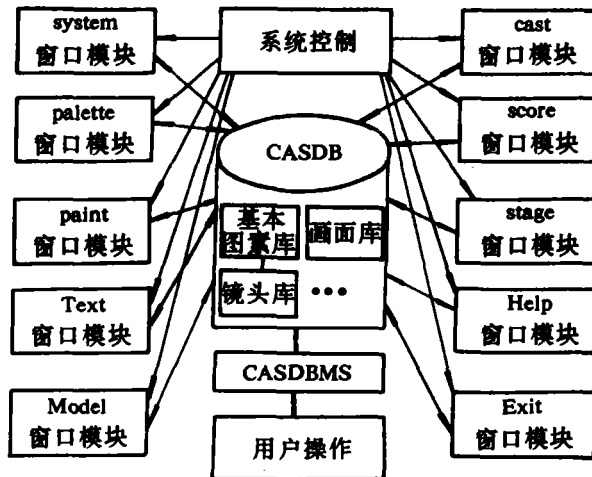


图 2 GKDCAS 软件结构

• 1991 年 3 月 21 日收稿, 1991 年 7 月 3 日收修改稿

算机、图形图象板以及数字化仪、鼠标器、扫描仪等交互工具。

整个系统是在 PC 机上最基本的 DOS 环境下用 C 语言实现的。系统软件由一个系统控制模块、十个窗口控制模块和动画制作数据库管理系统 CASDBMS 构成（如图 2 所示），其中系统控制模块控制各窗口模块的功能调度，各窗口模块可以向 CASDB 查询和存储数据。用户也可退回到 CASDBMS 对动画数据库内容进行管理，如输入、删除、修改或查询有关的动画数据等。CASDB 包括：基本图素库、画面库、镜头库……等。

## 2 GKDCAS 图形数据模型及结构

整个动画系统的管理、控制及状态记录都是通过系统数据结构标志的。系统数据结构包含两部分：(1) 系统描述及状态表（表 1）；(2) 图形数据存储结构。

表 1 系统描述及状态

名称	功能	表项
菜单表	定义菜单数组	共 20 项，包括： 主菜单、Score 菜单，……
屏幕分区表	定义菜单，对话框及特定功能的区域坐标值	共 68 项，包括： 主菜单区、主窗口区，……
指针与标志表	定义表示当前状态的指针与标志	共 70 项，包括： 当前菜单项，文件滚动指针，当前帧号，……

本系统图形数据模型(GDM)定义如下：

$\langle \text{GDM} \rangle ::= \text{SET\_OF} (\langle \text{镜头} \rangle); \quad \langle \text{镜头} \rangle ::= \text{STUC\_OF} (\langle \text{块} \rangle);$

$\langle \text{块} \rangle ::= \langle \text{块属性} \rangle + \text{STUC\_OF} (\langle \text{帧} \rangle);$

$\langle \text{帧} \rangle ::= \text{STUC\_OF} (\langle \text{图片} \rangle);$

$\langle \text{图片} \rangle ::= \langle \text{图片属性} \rangle + \text{STUC\_OF} (\langle \text{图原} \rangle) | \langle \text{基本图素} \rangle;$

$\langle \text{基本图素} \rangle ::= \text{STUC\_OF} (\langle \text{图原} \rangle);$

$\langle \text{图原} \rangle ::= \langle \text{图原属性} \rangle + \langle \text{图原实体} \rangle;$

根据上述模型，本系统的图形数据存储具体结构如下：

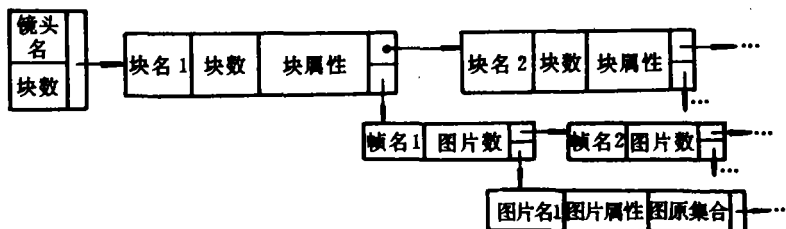


图 3 图形数据结构

图形数据类型(GDT)定义如下：

$\langle \text{GDT} \rangle ::= \langle \text{经典数据类型} \rangle | \langle \text{图原数据类型} \rangle;$

$\langle \text{经典数据类型} \rangle ::= \text{integer} | \text{real} | \text{logic} | \dots;$

$\langle \text{图原数据类型} \rangle ::= \langle \text{矢量图原} \rangle | \langle \text{位图图原} \rangle;$

$\langle \text{矢量图原} \rangle ::= \text{铅笔图原} | \text{刷子图原} | \text{喷枪图原} | \text{橡皮图原} | \text{直线图原} | \text{矩形图原} | \text{椭圆图原} | \text{英文图原} | \text{汉文图原} | \dots;$

位图图原由一个位图数组和该数组的尺寸定义。

图形数据结构的组织与管理主要是通过 CASDBMS 完成的。CASDBMS 按功能可划分为若干模块，通过菜单提示操作亦可在数据库管理系统支持软件环境（如 DBASE III 等）下直接工作。后一种工作方式主要是为系统生成与管理人提供的。一般用户（动画设计人员）利用 CASDBMS 提供的菜单窗口界面的图形操作方式即可。CASDBMS 总体结构如图 4 所示。

作为本动画系统数据库的基本成分，一般由系统生成与管理人提供为 CASDB 建立下述功能库：(1) 基本图素库。包括：人物类子库、动物类子库、植物类子库、山水类子库等；(2) 镜头库。包括：少儿子库、电教子库、文宣子库、科学实验子库等；(3) 画面库，主要保存上述功能库中重复使用率较高的画面。

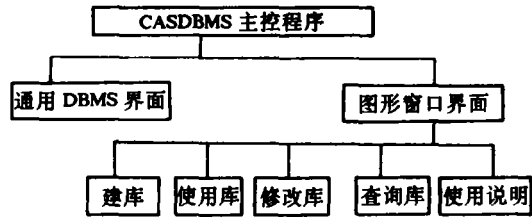


图 4 CASDBMS 基本结构

### 3 GKDCAS 的功能及特点

如图 1 所示，GKDCAS 含有十个窗口模块，各窗口分别对应一类功能：

System：系统资源和动画作者情况介绍；

Palette：模拟画家调色板功能，可自由调配颜色；

Paint：模拟真实世界的自由作画环境，提供如铅笔、刷子、喷枪、橡皮擦等多种绘图手段；

Text：英文汉字各种字体的文字画面编辑手段；

Model：三维实体造型，可提供自动消隐、明暗处理等功能；

Cast：高效的图片管理与铅笔画稿直接审查功能。图片栏上可直接显示整个文件的图片的缩小符号，并可以放大浏览和直接选用；

Score：提供直观简洁的动画序列编辑功能。用户可在时间和空间轴上直观地进行编辑，空间上各通道按前景优先级迭加成连续排列的帧，按时间顺序依次播出就可得到动画效果；

Stage：按设计要求正式播出总潜窗口的动画文件；

Help：系统提示与帮助；

Exit：退出 GKDCAS 系统，返回 DOS 系统。

本系统的人机界面为：用户在使用本系统制作动画时，首先启动计算机、电视信号合成器、图象监视器，然后运行批处理程序 GKDCAS.BAT，再通过图形界面，使用 mouse 作选择操作，最终完成动画产品的制作。

### 4 GKDCAS 中的软件技术

一个好的动画系统应使得操作尽可能简单，不能要求动画设计者对计算机技术和计算机图形学有多大程度的了解。对使用者最基本的要求就是，能象使用绘画工具那样使用辅助电子工具（输入笔、鼠标器等）。所以，许多复杂画面的制作技术要通过计算机图

形算法来完成和提供。GKDCAS 中提供了下述软件技术：

(1) 角色与背景的造型与输入。角色设计和背景造型是动画设计师的主要工作之一，本系统为这项工作提供了如下几种灵活简便且有效的手段：①利用扫描仪或摄影机输入一些高质量的图片；②利用数字化仪作简单轮廓设计；③在动画数据库中查询使用有关资料。

(2) 矢量与点阵图的混合表示。动画设计过程中，图片总是可以分解为基本图原的。图原采用图形学中通用的代数矢量表示，只是把任意位图作为一类特殊图原，从而按图原标志区别，解决了矢量与点阵两种不同图形混合使用的难题。

(3) 画家画笔与喷涂绘图。画家常用的画笔有多种，如各类铅笔、毛笔等，而具体画笔在涂绘时着笔轻重的不同产生不同的效果。软件模拟种种笔型效果采用了光栅模块、灰度调节等技术。

(4) 胶片镜像与旋转。在胶片制作过程中，经常用到的各种翻转效果，可通过对图象的数学变换实现。矢量图原结构不变，代数描述施以矩阵运算；位图图原改变结构，通过逻辑运算完成。

(5) 影片摄制的镜头调节效果，即通常的所谓 Zoom。直接通过图形学中的几何变换实现。

(6) 画面着色技术。采用图形学中扫描转换和种子扩散法实现。

(7) 套马索操作。所谓“套马索”是指设计者在画面上特别选定的一块图，然后对它重新进行定位或选定再设计时使用。运用矢量离散化结合扫描转换实现上述操作。

(8) 键帧动画。所谓“键帧动画”是指从外部指定主动作画面，而由计算机自动形成中间动作的画面。本系统内部对矢量图和位图分别进行插补。

矢量图的插补采用“运动点约束的插补”方法 (Reeves, 1981)。其特点是在时间和空间上都可根据用户要求进行非线性插补，以保证运动的连续性。其中一条与运动物体上某些点相关联的用以指定插补速度和轨迹的曲线被称为一个运动点。具体算法采用了 Miura 算法 (Miura et al., 1967)。

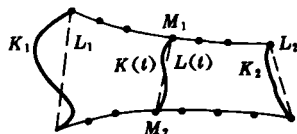


图 5 用运动点约束的插补

用前述矢量图原表示各帧画面，如  $K_1$ 、 $K_2$ 、 $K(t)$  等。

如图 5 所示，已知关键帧  $K_1$ 、 $K_2$  和运动点  $M_1$ 、 $M_2$  (用户指定)，求  $t$  时刻的插补曲线  $K(t)$ 。首先求出近似于  $K_1$ 、 $K_2$  的直线  $L_1$ 、 $L_2$  和近似于  $K(t)$  的  $L(t)$ ，再求变换  $T_1$ 、 $T_2$  (由平移、旋转、变尺寸构成的复合变换)，满足：

$$L(t) = T_1 \cdot L_1, \quad L(t) = T_2 \cdot L_2$$

则  $K(t)$  可表达为：

$$K(t) = (1 - t)T_1 \cdot K_1 + tT_2 \cdot K_2$$

其中， $t=0, \frac{1}{N}, \dots, \frac{N-1}{N}$  ( $N$  为插补总帧数)。

本系统中位图的插补只对整幅位图进行平移、旋转 (旋转中心可指定) 和变尺寸变换的插补。同样，平移时位图图片中心轨迹可由用户指定。

(9) 实时动画播放。将编辑的动画序列按设计要求正式播出，达到至少每秒 12 帧的动画效果。为了达到这一效果，软件首先完成离散点阵化，然后进行区域划分覆盖。只要对关键帧加以一定约束因子，其动画效果是令人满意的。

## 5 动画制作实例

由本系统制作的一个动画例子如图 6 所示。图中前景（行人）和背景（山）发生相对移动，同时人的姿势也发生了变换。具体做法是，先在 Paint 窗口中画出背景全貌和人的两种姿势图，然后在 Score 窗口中进行动画序列编辑，指定背景和前景的相对位置，最后在 Stage 窗口模块中播出动画序列。

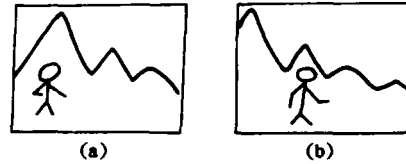


图 6 动画制作实例

此时计算机内部的实际操作过程为：背景全图存入一个  $1024 \times 1024$  的帧存中；保存图 6(a) 前景位置的背景；显示图 6(a) 前景；擦去图 6(a) 前景，恢复背景；漫游背景到图 6(b) 位置；保存图 6(b) 前景位置的背景；显示图 6(b) 前景。

## 6 结束语

用计算机辅助完成动画制作，这个课题在近年的 SIGGRAPH 和 Eurographics 年会上极受重视，已经形成热门研究项目。对于动画设计家而言，使用计算机确实为他们开拓了新的自由创作空间，可以最大限度地发挥其艺术创造力。GKDCAS 是建立在通用微机系统上的一个实用动画系统，尽管在三维真实造型方面还不完善，但整个系统适应我国的国情，它的推广应用将有力地改变目前在教育、娱乐等方面动画制作的落后状况。

### 参 考 文 献

- 1 Magnenat-Thalmann N, Thalmann D. Computer Animation Theory and Practice. Tokyo, 1985
- 2 Neal weinstock. Computer Animation. Addison-Wesley Publishing Co., 1989

## A Computer Aided Animation System (GKDCAS)

Zhang Jun Li Yan Wang Liangjun Tang Zhenchu Hu Hengjiang  
(Visual and Audio Center)

### Abstract

GKDCAS is a practical animation system designed in C-language and based on micro-computer environment. This paper is to present the software structure of GKDCAS and its design methods and to introduce its functions. An example showing how to make animation with this system is also given.

**Key words** computer animation, interactive computer graphics, data base, keyframe animation