



UNIVERSIDADE DA CORUÑA

Generación automática de secuencias en animación a partir de vídeo

Autor: Diego Moreda Rodríguez
Director: Emilio José Padrón González

Proyecto de Fin de Carrera de Ingeniería Informática
Depto. Electrónica y Sistemas, Facultad de Informática, UDC

21 de septiembre de 2016

Introducción

Generación automática de secuencias en animación a partir de vídeo

- ▶ Creación de un sistema que facilite generación de secuencias de animación a partir de fotogramas vídeo: *rotoscopía*.
- ▶ Desarrollo de un **método automático** para **interpolar** los fotogramas intermedios de una forma suave y visualmente atractiva. Problemas clave: **seguimiento** (*tracking*) y **correspondencia** entre curvas (*curve matching*).

Introducción

Generación automática de secuencias en animación a partir de vídeo

- ▶ Creación de un sistema que facilite generación de secuencias de animación a partir de fotogramas vídeo: *rotoscopía*.
- ▶ Desarrollo de un **método automático** para **interpol**ar los fotogramas intermedios de una forma suave y visualmente atractiva. Problemas clave: **seguimiento** (*tracking*) y **correspondencia** entre curvas (*curve matching*).

Rotoscopía

Utilización de secuencias de vídeo real sobre las que un artista dibuja manualmente los fotogramas de una animación. [Fle17]

Problemas: coste, coherencia temporal



Objetivos de nuestro trabajo

Desarrollar un sistema que *automaticamente* la generación de los fotogramas situados entre dos fotogramas clave, consiguiendo:

1. obtener una **interpolación** visualmente atractiva y coherente, utilizando algoritmos de **seguimiento** y **correspondencia**.
2. dotar al artista de la máxima **flexibilidad** en los trazos iniciales y posibilitando personalizar el resultado.
3. desarrollar e implementar una **aplicación** multiplataforma con un interfaz sencillo e intuitivo, y funcionalidad avanzada.

Visión general del sistema



Visión general del sistema



1. Artista:

1.1 añade *fotogramas* (imágenes)

1.2 importa *trazos* en fotogramas clave



Visión general del sistema



1. Artista:

1.1 añade *fotogramas* (imágenes)

1.2 importa *trazos* en fotogramas clave

2. Sistema: realiza *seguimiento automático*:

2.1 Calcula *correspondencia* ente las curvas.

2.2 Realiza *interpolación* de las curvas de los fotogramas intermedios.



Visión general del sistema



1. Artista:

1.1 añade *fotogramas* (imágenes)

1.2 importa *trazos* en fotogramas clave

2. Sistema: realiza *seguimiento automático*:

2.1 Calcula *correspondencia* ente las curvas.

2.2 Realiza *interpolación* de las curvas de los fotogramas intermedios.

3. Artista: modifica manualmente curvas en cualquier fotograma y vuelve a #2.



Método de seguimiento basado en fotogramas clave

Función de energía

Objetivo seguimiento \Rightarrow minimizar la **función de energía** E :

$$E = \underbrace{\omega_L E_L + \omega_C E_C + \omega_V E_V}_{\text{términos de forma}} + \underbrace{\omega_I E_I + \omega_G E_G}_{\text{términos de imagen}}$$

Los términos iteran sobre **curvas y muestras**, comparando una muestra i de una curva c con la muestra correspondiente i de la curva siguiente $c + 1$.

$$E_x = \sum_{c=c_j}^{c_f-1} \sum_{i=0}^N f_x(\zeta_i^c, \zeta_i^{c+1})$$

Método de seguimiento basado en fotogramas clave

Función de energía

Objetivo seguimiento \Rightarrow minimizar la **función de energía** E :

$$E = \underbrace{\omega_L E_L + \omega_C E_C + \omega_V E_V}_{\text{términos de forma}} + \underbrace{\omega_I E_I + \omega_G E_G}_{\text{términos de imagen}}$$

Los términos iteran sobre **curvas y muestras**, comparando una muestra i de una curva c con la muestra correspondiente i de la curva siguiente $c + 1$.

$$E_x = \sum_{c=c_j}^{c_f-1} \sum_{i=0}^N f_x(\zeta_i^c, \zeta_i^{c+1})$$

T.	Nombre	Descripción
E_L	Longitud	Longitud del vector que une 2 muestras consecutivas
E_C	Curvatura	Cambio en la curvatura (aproximando la 2ª derivada)
E_V	Velocidad	Cambios bruscos en la posición de las muestras
E_I	Intensidad	Cambios en la intensidad de la imagen (<i>ventanas</i> de px)
E_G	Gradiente	Diferencia en la magnitud del gradiente de la imagen

Método de seguimiento basado en fotogramas clave

Optimización

Solución \Rightarrow encontrar las **posiciones de los puntos de control** de las curvas que minimicen la función de energía

$$E = \sum_K \omega_k \|\mathbf{f}_k(\vec{x})\|^2 \quad \vec{x} \equiv \text{Vector de incógnitas (ptos. control)}$$

Es **problema de mínimos cuadrados no lineales (NLLS)** que se resuelve utilizando una variante del método de Levenberg-Marquardt (Steihaug) por niveles (4 pirámides de Gauss).

Método de seguimiento basado en fotogramas clave

Optimización

Solución \Rightarrow encontrar las **posiciones de los puntos de control** de las curvas que minimicen la función de energía

$$E = \sum_K \omega_k \|\mathbf{f}_k(\vec{x})\|^2 \quad \vec{x} \equiv \text{Vector de incógnitas (ptos. control)}$$

Es **problema de mínimos cuadrados no lineales (NLLS)** que se resuelve utilizando una variante del método de Levenberg-Marquardt (Steihaug) por niveles (4 pirámides de Gauss).

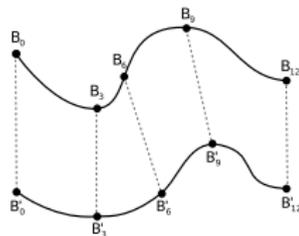
1. $Z_0 \leftarrow$ posiciones de los ptos. control actuales.
2. Para cada nivel de pirámide de Gauss:
 - 2.1 Recalcula muestras en función de Z_0 .
 - 2.2 Calcula matriz jacobiana J .
 - 2.3 $Z_1 \leftarrow$ solución al problema NLLS $E(J)$.
 - 2.4 $Z_0 \leftarrow Z_1$
3. Finaliza: Z_0 contiene los nuevos puntos de control.

Método de seguimiento basado en fotogramas clave

Diferencias con el trabajo original

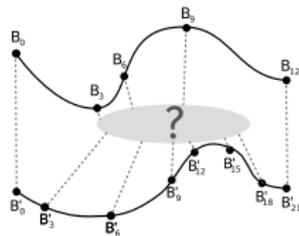
En el trabajo original [AHSS04], la curva final es **siempre una copia** de la curva inicial \Rightarrow se asume que:

- ▶ Cada punto (de control, muestra) de la curva c representa siempre la misma característica de la curva $c + 1$.



En nuestro trabajo, curvas inicial y final son **totalmente diferentes** (nº y posición ptos. de control) \Rightarrow se necesita:

- ▶ Desarrollar la **correspondencia** entre curvas.
- ▶ Otros ajustes (término E_C , muestreo no uniforme).



Algoritmos de correspondencia (1)

Propiedades ideales algoritmo correspondencia:

- ▶ monotonicidad
- ▶ invarianza a transformaciones
- ▶ coincidencia de extremos
- ▶ **preservación de *features***

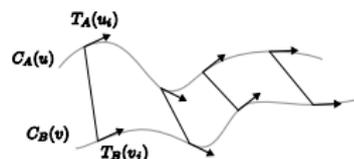
Algoritmos de correspondencia (1)

Propiedades ideales algoritmo correspondencia:

- ▶ monotonicidad
- ▶ invarianza a transformaciones
- ▶ coincidencia de extremos
- ▶ **preservación de *features***

1. Basado en correspondencia entre vectores tangentes [CEBY97]

Compara los *vectores tangentes normalizados* de las curvas, que representan los cambios en la curvatura.



Aplica programación dinámica para resolver la asignación ($\mathcal{O}(mn)$).

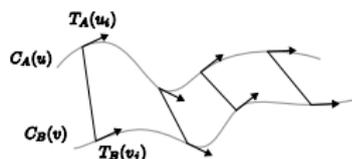
Algoritmos de correspondencia (1)

Propiedades ideales algoritmo correspondencia:

- ▶ monotonicidad
- ▶ invarianza a transformaciones
- ▶ coincidencia de extremos
- ▶ **preservación de *features***

1. Basado en correspondencia entre vectores tangentes [CEBY97]

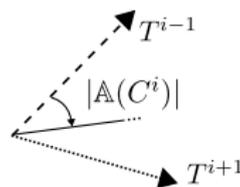
Compara los *vectores tangentes normalizados* de las curvas, que representan los cambios en la curvatura.



Aplica programación dinámica para resolver la asignación ($\mathcal{O}(mn)$).

2. Método basado en la similitud de las curvas [JKN00]

Añade términos de *índice* (posición de la muestra en la secuencia) y de *ángulo* (entre vectores tangentes de muestras anterior y sig.).



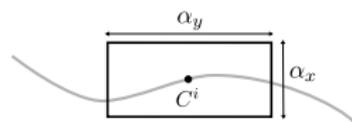
Mejor varianza a rotación y sensibilidad a *outliers* ($\mathcal{O}(\log_2(m)n^2)$).

Algoritmos de correspondencia (2)

3. Método guiado por imagen [LL02]

Añade un *término de imagen* basado en la luminancia relativa de una *ventana* centrada en la muestra.

Mayor coste computacional ($\mathcal{O}(mn(\alpha_x\alpha_y)^2)$).

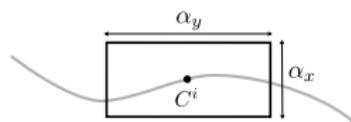


Algoritmos de correspondencia (2)

3. Método guiado por imagen [LL02]

Añade un *término de imagen* basado en la luminancia relativa de una *ventana* centrada en la muestra.

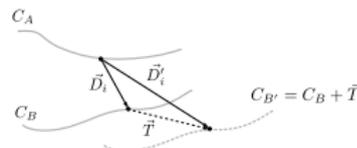
Mayor coste computacional ($\mathcal{O}(mn(\alpha_x\alpha_y)^2)$).



4. Método basado en el vector de desplazamiento [GGCV95]

Utiliza la *distancia euclídea* entre las muestras y las *posiciones relativas* de las mismas.

Invarianza a traslación, pero no a rotación ($\mathcal{O}(mn)$).



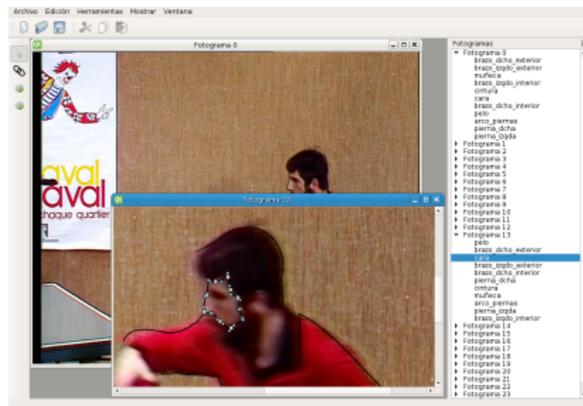
Aplicación ARAS

Introducción

La herramienta **ARAS** (*Automatic Rotoscoping for Animated Sequences*) implementa el desarrollo teórico de nuestro trabajo con un interfaz ajustado a las necesidades prácticas del artista.

Objetivos:

- ▶ Interfaz sencillo e intuitivo
- ▶ Integración con otras aplicaciones
- ▶ Multiplataforma, software libre
- ▶ Seguir mejores prácticas (técnicas, estándares, i8n, documentación, ...)



Aplicación ARAS

Diseño

OO con **patrones** (MVC, Observer, Command, Strategy, ...). Retos:

- ▶ reutilización y adaptación de la aplicación original
- ▶ desarrollo de componentes de visualización (*OpenGL* + *Qt*)
- ▶ módulos de importación/exportación (*parsers PostScript/SVG*, XML)

Aplicación ARAS

Diseño

OO con **patrones** (MVC, Observer, Command, Strategy, ...). Retos:

- ▶ reutilización y adaptación de la aplicación original
- ▶ desarrollo de componentes de visualización (*OpenGL + Qt*)
- ▶ módulos de importación/exportación (*parsers PostScript/SVG, XML*)

Implementación

Énfasis en **estándares abiertos, software libre y multiplataforma**:

- ▶ C++ (*STL*), compilado con *GCC*
- ▶ Librería *Qt 5*; otras: *boost, ImageMagick, log4cxx, ragel*

Aplicación ARAS

Diseño

OO con **patrones** (MVC, Observer, Command, Strategy, ...). Retos:

- ▶ reutilización y adaptación de la aplicación original
- ▶ desarrollo de componentes de visualización (*OpenGL* + *Qt*)
- ▶ módulos de importación/exportación (*parsers PostScript/SVG*, XML)

Implementación

Énfasis en **estándares abiertos, software libre y multiplataforma**:

- ▶ C++ (*STL*), compilado con *GCC*
- ▶ Librería *Qt 5*; otras: *boost*, *ImageMagick*, *log4cxx*, *ragel*

Metodología

Metodología ágil sin adscribirnos a una variación en concreto:

- ▶ adecuación a la naturaleza “abierta” del trabajo y requisitos difusos.
- ▶ desarrollo iterativo con *sprints*, hitos, entregables.

Resultados experimentales

Escenas y métricas utilizadas

Métricas - comparan curva “ideal” con curvas interpoladas:

- ▶ Distancia media (normalizada): $|\mathbf{DM}| = \frac{1}{m} \sum_{i=0}^{m-1} d(C_A^i, C_B^i) \cdot \text{Diag}$
- ▶ (Peak) signal-to-noise ratio: $\mathbf{PSNR} = 10 \log_{10} \left(\frac{\text{Diag}^2}{\text{ECM}} \right)$

Secuencias:

- ▶ **“Amira”**:
10 fotogramas (640x480),
2 curvas

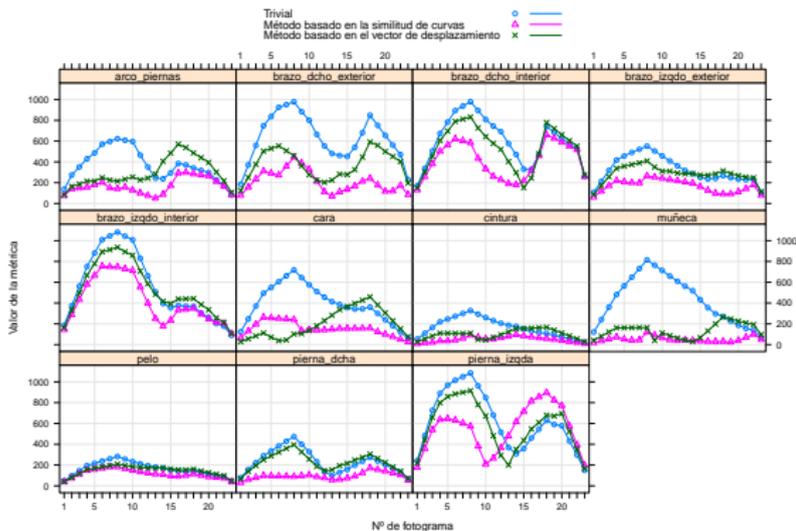


- ▶ **“Table-tennis”**:
25 fotogramas (720x576),
11 curvas



Resultados experimentales

Resultados (1): bondad de los resultados

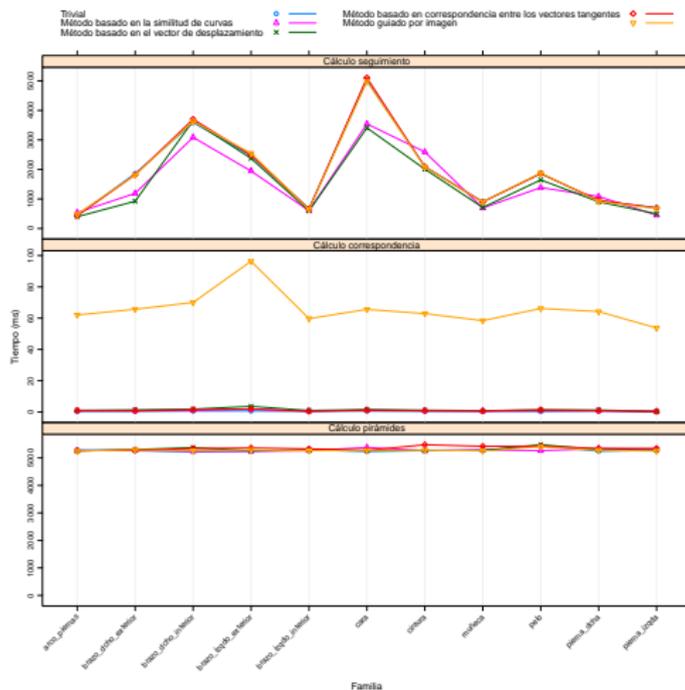


DM familias secuencia "Table-tenis"

- ▶ Cualquier correspondencia influye positivamente en el resultado final.
- ▶ Correspondencia basada en similitud de curvas > basada en vector de desplazamiento > trivial.
- ▶ Algunas curvas son "difíciles" (cambios bruscos, longitud, oclusión, continuidad) independientemente de la correspondencia.

Resultados experimentales

Resultados (2): tiempos



- ▶ Cálculo de pirámides gaussianas representa la mayor parte del tiempo ($\sim 77.14\%$).
- ▶ Cálculo de correspondencia lleva muy poco tiempo porcentualmente (máximo 1.21%).
- ▶ Cálculo del seguimiento muy variable ($7.02\% - 48.94\%$) dependiendo de las características de la curva e iteraciones.
- ▶ Entre las correspondencias, el método basado en imagen es el que lleva más tiempo ($\sim 20x$) al necesitar iterar sobre la imagen.

Tiempo por etapa familias secuencia "Table-tennis"

Conclusiones

- ▶ Hemos revisado y mejorado un **método de seguimiento** para generación de curvas intermedias, utilizando diferentes **algoritmos de correspondencia** como el principal método para **incrementar la calidad** de los resultados.
- ▶ Hemos **desarrollado una aplicación** que permite al usuario realizar todo el proceso de una forma intuitiva y controlar el resultado final.
- ▶ Combinando ambas, hemos **umentado la flexibilidad** al eliminar limitaciones del método original; y hemos reducido sustancialmente el tiempo empleado por el usuario en el proceso.

Trabajo futuro

- ▶ Aplicación de nuevos algoritmos de correspondencia (método basado en propiedades geométricas de un conjunto característico de puntos, *feature matching*, desarrollado en A.1).
- ▶ Preprocesado inteligente de los trazos inicial y final:
 - ▶ División de curvas en igual número de segmentos
 - ▶ Simplificación de curvas
- ▶ Paralelización del proceso de seguimiento.

TODO

- [AHSS04] A. Agarwala, A. Hertzmann, D.H. Salesin, and S.M. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics (TOG)*, 23(3):584–591, 2004.
- [CEBY97] S. Cohen, G. Elber, and R. Bar-Yehuda. Matching of freeform curves. *Computer-Aided Design*, 29(5):369–378, 1997.
- [Fle17] Kax Fleischer. Method of producing moving picture cartoons, October 9 1917. US Patent 1,242,674.
- [GGCV95] D. Geiger, A. Gupta, L.A. Costa, and J. Vlontzos. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3):294–302, 1995.
- [JKN00] H. Johan, Y. Koiso, and T. Nishita. Morphing using curves and shape interpolation techniques. *Proceedings of Pacific Graphics 2000, 8th Pacific Conference on Computer Graphics and Applications*, pages 348–358, 2000.
- [LL02] T.Y. Lee and C.H. Lin. Feature-guided shape-based image interpolation. *Medical Imaging, IEEE Transactions on*, 21(12):1479–1489, 2002.