

Video Narrative Authoring with Motion Inpainting

Timothy K. Shih

Department of CSIE, National Central
University
No.300, Jhongda Rd., Jhongli City,
Taoyuan County 32001, Taiwan
timothykshih@gmail.com

Joseph C. Tsai

Department of CSIE, Tamkang
University
No. 151, Ying-chuan Rd., Tamsui,
Taipei County 25137, Taiwan
kkiceman@gmail.com

Kuan-Ching Li

Department of CSIE, Providence
University
No. 200, Chung Chi Rd., Taichung
County 43301, Taiwan
kuancli@pu.edu.tw

ABSTRACT

Storytelling and narrative creation are recent interests in the areas of interactive media designs. Instead of using virtual reality-based 3-D models, we propose a system which uses video technologies to generate video story from existing avatars and videos, with moderate avatar control technologies. The user only needs to involve in two steps: (1) select a background video as video scene; and (2) pick an “object track” and set up its trajectory. In order to plan for a realistic narrative, several issues such as computing relative sizes of avatars, object pasting, and video scene generating are considered. We use an algorithm for aligning motion of object tracks in order to make object movement smoother. For producing a video static scene including calibrating all layers, we maintain a motion map for each video frame and use the maps as guidance when removing objects in video and combining all frames back to a video scene. To generate a dynamic background, we proposed motion inpainting to create more dynamic textures and insert the new patch into inpainting area to generate a new dynamic background. An authoring tool equipped with special functions to integrate different motion tracks for the generation of video narratives is also presented in this paper.

Categories and Subject Descriptors

I.4.4 Restoration, I.4.5 Reconstruction, I.4.3 Enhancement

General Terms

Algorithms, Design, Experimentation

Keywords

Motion interpolation, Special Effect, Video Inpainting, Video Narrative Generation, Video Planning

1. INTRODUCTION

Special effect production in movie industry is time consuming and challenge. Augmented Reality is commonly used in digital movies. The creation of 3D models and animations could rely on exiting software tools and hardware tracking devices. However, it takes a tremendous amount of human effort for post processing. Alternatively, special effect can be created by asserting actors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM Workshop on Multimodal Pervasive Video Analysis, October 25–29, 2010, Firenze, Italy.

Copyright 2010 ACM 1-58113-000-0/00/0010...\$10.00.

extracted from a blue background. It is relatively realistic to create special effects, within a studio for real persons as compared to using 3D models. However, to create outdoor special effects is hard, especially for those in a dangerous zone. We propose a series of methods to plan video narratives by using existing video clips, which can be combined, extended, interpolated, or altered.

Narrative and story planning [3] could use a mixed reality approach by tracking an avatar in video. The avatar can be combined with a pre-defined 3-D virtual reality model. The work discussed in [3] uses an artificial intelligence technique for interactive story telling. A critical step to make the story looks real is the precision of object tracking in a video. For real-time applications, efficient solutions such as those proposed in [4, 7] seems to be useful. However, a simple but efficient tracking technique is not enough if one consider subdividing a video into multiple layers due to diversity of layer combinations. Our research is based on two earlier contributions, video inpainting [6, 10, 13] and video forgery [9]. In this paper, the new contribution is on the generation of a multilayered video schematic, which includes distances of different layers due to our new tracking and segmentation mechanism. In addition, a spatiotemporal placement algorithm is proposed to precisely place avatars in the schematic. Color properties are adjusted such that combining layers from different video sources can be presented realistically.

The challenge in building a video narrative authoring tool is to generate a spectacular dynamic background. Dynamic textures are very hard to create. For dynamic background panorama generation [17], Rav-Acha et al. proposed dynamic mosaics by sweeping the aligned space-time volume of the input video by a time front surface and by generating a sequence of time slices. The algorithm can control dynamic background speed. Texture of the dynamic background is used to extend the length of video. Inspired by this approach, we propose a new method called “motion inpainting.” Our algorithm can generate extra features of a dynamic background and thus is uses in our authoring tool.

In section 2, we talk about video scene generation. There are two parts in the section, the first is static background generation and the second is dynamic background processing. We discuss how to merge the background and foreground smoothly and naturally in section 3. Section 4 shows the experimental results and the final section is the conclusion.

2. VIDEO SCENE GENERATION

The first issue in generating a video narrative is to produce a visually pleasant video scene. There are two kinds of background to be generated. One is the static background and another is the dynamic background. We will discuss these two parts in the section. In generating static background, given a selected video

clip, the first step is to construct motion maps of each frame. After the motion maps are computed, the procedure of frame referencing can find out appropriate information from other frames to replace the area of possible missing foreground object through the assistance of motion vectors. The procedure of frame combination can also be achieved by using motion information to determine the overlap areas between consecutive frames. In order to generate extra dynamic background, we propose a new method called “motion inpainting” to choose a patch with dynamic texture and try to extend the video to an arbitrary length. After selecting new patches, we insert them in the inpainting area and blend the inserted patch with the source background.

2.1 Static Background Generating

To generate static background, there are three steps. The first is to compute the motion map for every video frame. After motion map computing, if there were some objects to be removed, we can rely on the motion map to finish the process. Finally, we have to generate a panorama. The panorama allows the user to control the background moving speed.

In order to maintain temporal continuity, we propose a motion map algorithm. The following are the main steps of the algorithm:

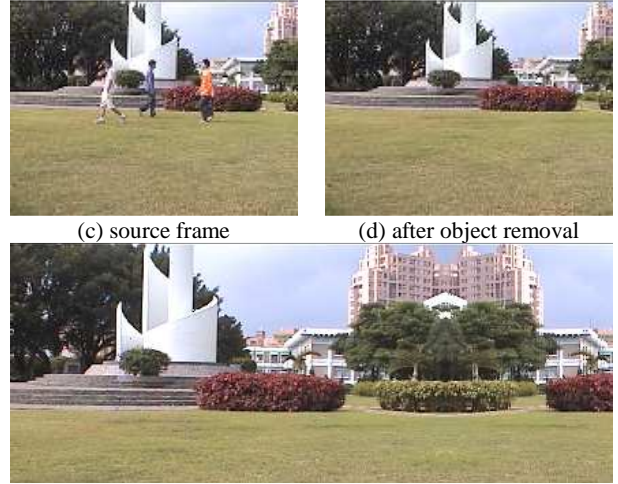
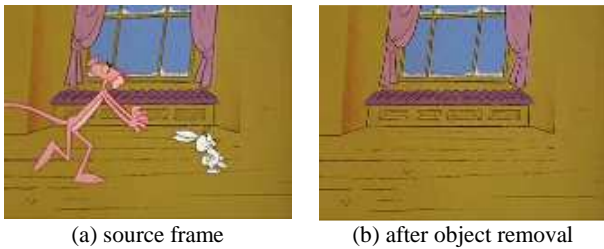
1. Use CDHS algorithm [4] to compute motion vectors.
2. Generate a new frame F_{t+1} by the motion vectors from step1.
3. Compare the differences between F_{t+1} and F_{t+1} .
4. Construct a motion map based on the motion vectors calculated from each block.

After the above algorithm is implemented, we can use the information of motion map to remove unnecessary objects. The method is similarity to one discussed in [13]. We use the video inpainting algorithm for object removal. Figure 1 shows two results.

Finally, we have to generate a panorama from the results in the above steps. The algorithm of multi-band blending [2] proposed by Burt and Adelson is commonly used in combining two or more images for generating a large image mosaic. In this section, we used this technique with a simple 3-band scheme to combine all frames in a video. Hence, a video scene F can be generated by using this technique and also can be described as follows:

$$F = \bigcup_{i=1 \text{ to } N} B(f_i) \quad (1)$$

where N is the frame number in a video, f_i is a frame after remove the foreground objects at time i and B is the function of frame blending. Figure 1 (e) is a panorama sample.



(e) Camera Motion: Panning Video
Figure 1. Results of patch referencing and frame blending

2.2 Dynamic Background Generating

The continuity of dynamic background is very hard to maintain. We can't use the above method or image inpainting algorithm to do the dynamic background generating. Traditional image inpainting tries to restore area covered by patches from other areas in the same frame. But the dynamic background can't be inpainted by nearby patches. It is necessary to consider the temporal continuity in all frames of videos. In this reason, we propose a motion inpainting algorithm to generate several extra parts from dynamic background. There are two important steps in this approach, the firsts is to extend the length of video and the second is patch blending.

2.2.1 Extension of Video Length

Before we extend the length of the input video, we have to choose an inserted patch and inpainted area. And, we will compute the similarity of the structure information in the edge of the insert patch and the inpainted area. To compare the similarity between the inserted patch and inpainted area, we use the following formula:

$$Structure_{SAD(i,j)} = \sum_{a=0}^{p-1} \sum_{b=0}^{q-1} |Struct_{check}(i+a, j+b) - Struct_n(i+a, j+b)| \quad (2)$$

where, $struct_{check}$ is the patch of the computing frame to make the estimation. The function $struct_n$ is the inpainted area in the first frame. After computing, we can get a frame number. That means the patch of this frame fits the best to the inpainted area in the first frame. Because the frame number for computing is not definite, the lengths of the two patch video are not the same. By this reason, we have to extend the length of the video to make the final video much better.

In order to extend the length of the selection patch video, we have to analyze the video. Video texture [14] was proposed to find the loop in a video. This algorithm is suitable for us to extend the patch video. The original algorithm computes whole size of the frame; it will cost too much time to analyze the video. Another problem is that every part of dynamic background is not the same. The analysis of original algorithm can't look for a best loop to

extend the video. So we modify the algorithm to compute a small patch video. The following is our video analysis algorithm:

1. Select the patch need to analyze
2. Computing the similarity between each frame. The following is the formula of this step.

$$D_{ij} = \sum_{i=1}^N \sum_{j=1}^N \sum_{k=-m}^m \|I_{i+k} - I_{j+k}\|_2 \quad (3)$$

,where N is the number of the frames. And, m is the computing frame number. We close m as 2 in our experiment. To compute the similarity between each frame in the video clip, we use L2 norm to finish this computing. Finally, we can get a matrix which records the similarity of the frames, as demonstrated in figure 2. In the next step, the loop will be according to this matrix to generate.

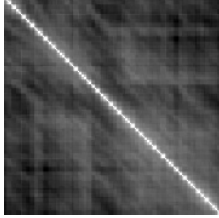


Figure 2. A example of similarity matrix from each frame in a video. The intensity in the matrix represents the similarity (the higher the intensity, the more similar of frames).

3. After step 2, we can generate the matrix from the frames of the video clips. In figure 4, we can see the similarities of the diagonal are the best cases in the matrix. The reason is that the values on the diagonal are computing from frame_i and frame_i. So, the results are the best. In order to extend the length of input video, we have to find the best loop from the matrix. At first, the loop begins on the last frame. We will accord the matrix to find the most similar frame to connect with the last frame.

4. The strategy in generating the infinite length video is distributed into two steps. The first is to find the connecting frame backward the target frame. In order to make the result smoother, the new connect frame must consider the similarity. The second step is to find the best similarity frame from current frame to the end frame.

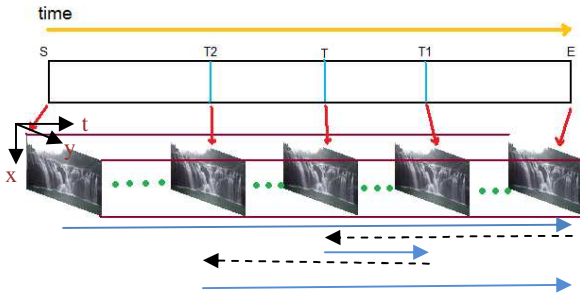


Figure 3. A example to extend the length of video. The new order is S->E->T->T1->T2->E. The dotted line in the figure means the connected order from the start point to the end point.

There is an example for step 4 showed in figure 3. In this figure, the symbol S is the first frame in the video, the symbol E is the

end of the video. The symbol T is the first connect frame searched, while the video plays to the frame E it will continue to frame T. Symbol T1 is the best choice from T to E, in the rule of getting the smoothest connecting frame. Finally, symbol T2 is the most similar to T1. So the order of the new extended video is S, E, T, T1, T2, and E.

After the above steps, we will get a new extended video. But it is just a video patch. We have to insert the new patch into the inpainting area.

2.2.2 Blending for Inserted Patches

In the last section, we know the fittest frame number to be inserted into the inpainting area. We also get an infinite patch video which can make the length of the new patch and source video. If we only insert the patch into the source frame, the final result will be very strange.

In order to make the merged result of inpainting area and the original image optimum, we use the multilayer blending concept to process the surroundings of the inpainting area. However, it's not only the surroundings of the inpainting area but also the middle of this area. Because there is a problem of the priority computing, it will make the middle part of this inpainting area not continuous. So we will focus on the surroundings and middle of the inpainting area to process the multilayer blending algorithm.

There are many algorithms proposed to do multilayer blending, such as graph cut [15,16] and Poisson algorithm [11]. The Poisson editor is an algorithm to merge two layers and make the seam of these two layers smoother. The goal of this paper is to inpaint an area which is not dynamic background but used as a dynamic background. So we can treat this inpainting area as a new layer and the original image as another layer. Then we can use the Poisson algorithm to merge the layers.

In general, we will compute all layers in the Poisson algorithm. The goal is to make the new layer blend in the original layer very naturally. We won't process all layers in this paper. If we do the Poisson algorithm for all layers, the features of the structure in dynamic background will be lost. And, the inpainting area will become fuzzy. So we just focus on the surroundings and middle of the inpainting area to process the Poisson algorithm.

Figure 4 illustrates the area that we have to process the Poisson algorithm [11]. The original image is Φ_1 , the inpainting area is Φ_s , because we just process the Poisson algorithm on the surroundings of this area, we use the three pixels wide area marked as Φ_{ar} . Then the Poisson algorithm will be used to merge two layers.

$$\min_{f_p} \sum_{(p,q) \in \Phi_s, \Phi_s \neq \emptyset} (f_p - f_q - v_{pq})^2, \text{ with } f_p = f_p^*, \forall p \in \Phi_{ar} \quad (4)$$

where v_{pq} is the difference between point p of gradient and point q of gradient. The point q represents the neighbors of point p. There is a condition to make this formula completed.

$$\text{for all } p \in \Phi_s, |N_p| f_p - \sum_{q \in N_p \cap \Phi_s} f_q = \sum_{q \in N_p \cap \Phi_{ar}} f_q^* + \sum_{q \in N_p} v_{pq} \quad (5)$$

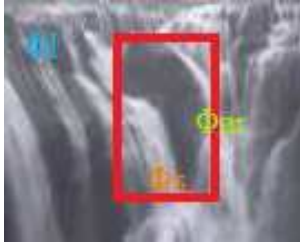


Figure 4. Φ_1 is the original image, Φ_s is the inpainting area and Φ_{ar} is the surroundings of Φ_s .

Equations form a classical, sparse (banded), symmetric, positive-definite system. Because of the arbitrary shape of boundary Φ_{ar} , we must use well-known iterative solvers. In figure 6, we will show more results.

3. VIDEO NARRATIVE GENERATION

Given a video schematic, the user has to choose avatars and use video scene as the base for timing and positioning. The user also needs to decide the type of clip (for extrapolation, interpolation, or down sampling). In order to produce a realistic result, several techniques such as motion clip processing and layer merging should be considered. The motion clip includes two methods: motion interpolation and extrapolation is used to produce a suitable video length of each motion clip. Layer merging is performed from the last layer to the first layer. In this section, detailed methods are discussed.

For generating a meaningful video narrative, an algorithm of motion interpolation/extrapolation of avatars is used in this paper for producing a suitable video length of each motion clip. In an earlier research result [12], authors proposed a motion analysis algorithm. This work can accord the information of the object motion to control the motion speed of object. By this algorithm, we can finish the motion interpolation and extrapolation.

When we get a new object with different speed, the motion clip needs to consider. It is a basic element to tell a story in a video narrative. A motion clip relies on the users to choose a motion track in the story. In our current research, we have three types of motion tracks:

1. **Regular Motion Track:** a motion clip can be placed in normal speed, fast speed, or slow speed. Reversed time line is also possible.
2. **Extrapolated Motion Track:** a cyclic motion clip can be repeated for a duration defined by the user.
3. **Spark Motion Track:** a motion clip is placed on specific time slots only. Actors are displayed on the motion track for a fixed number of frames before removal (similar to the effect in the movie "Jumper").

The above three types of motion tracks can be merged to create new motion tracks. The spatiotemporal placement algorithm takes several steps:

1. The user has to select a motion clip, decide its motion track starting location, and give extra parameters such as speed and length of extrapolation.
2. For each motion track
 - 2.1. Compute time slots of appearance
 - 2.2. Compute relative position of the object on frame
3. The user has to select a playback speed. The playback tool decides position of frames in the panorama and combines frames into a video.

After these three steps, we can get the information of background and foreground. The information includes speeds, motion direction and object's special effect etc. Computing time slot (i.e., frame number) for placing avatar is simply performed by slicing the panorama. That is, according to the playback seed, the panorama is segmented into background frames. However, spatial placement requires two extra steps. In the first step, assuming that F_0 and F_1 are two consecutive frames due to step 2.1, where the target object is placed. For the second step, the depth of avatar needs to be considered. We do not have an automatic solution for all types of cameras. However, for our experimental camera, we use a few distance markers to estimate the perspective effect.

Finally, we have to merge the foreground and background. Recently, Poisson image editing [11] is considered as a popular and robust technique for seamless image composition and can offer visually pleasant composition results. However, the results may not be acceptable in some cases due to object's excessive tone changes. In order to solve this problem, we propose an algorithm with a main purpose similar to [11] to preserve object color information and make the result of layer merging more realistic. This algorithm can be separated into two steps: (1) using Poisson equation to blend the boundary part of object with target region; and, (2) adjusting object's intensity and saturation by analyzing the target region. After the two steps, the foreground and background can merge much smoother. In reality, the method to merge two layers is very similar to equation 4 and equation 5. The area Φ_{ar} is assigned to the contour of the object. The width of the contour is four pixels in our experiment. Figure 5 shows an actor we used.

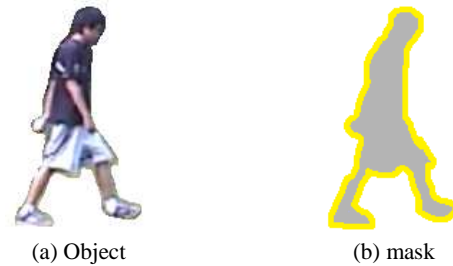


Figure 5. (a) is the object to be inserted. (b) is the mask as the reference for the Poisson algorithm. The yellow part is the contour to process the Poisson and the gray part will be adjusted according to the surrounding information.

4. EXPERIMENT

We demonstrate video narratives generated. Each example shows static background generation and dynamic background generation. We also show the results of storytelling in figure 6. N1 showed an example of waterfall. We insert a new waterfall to the right side. N2 is a view of volcano. The volcano is doubled. In N3, we use motion inpainting to add a new waterfall to the right, showing in a red box. In the narrative generation of N4, we insert an avatar in the background. Although it is hard to find a quantitative evaluation mechanism, our experiments illustrate excellent results.

Interested readers are welcome to visit our website at <http://member.mine.tku.edu.tw/www/workshop10/index.htm>.

5. CONCLUSION

We proposed a series of mechanisms to generate video narratives from existing video clips. We use patch referencing and frame blending to generate a video scene as a base for the user to plan for video tracks in video narrative generation. Object's tone and size adjustment are used to make the result more realistic. We allow video layers in different videos to be combined by adjusting the saturation, the intensity, and the spatiotemporal placement of layers. The motion inpainting is proposed to generate dynamic background. We demonstrate the feasibility of using our mechanisms for special effect production in digital movies. Applications of our mechanism include video forgery and special effect production. There still exist few limitations in our proposed algorithm.

6. ACKNOWLEDGEMENT

This research is sponsored by the National Science Council (NSC), Taiwan, under grant NSC96-2221-E-126-004-MY3. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSC.

7. REFERENCES

- [1] M. Ahmed, R. Ward, "A Rotation Invariant Rule-based Thinning Algorithm for Character Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 24, Issue 12, Dec. 2002 Page(s):1672 – 1678.
- [2] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Transactions on Graphics*, 2(4):217–236, 1983.
- [3] F. Charles, M. Cavazza, S. J. Mead, O. Martin, Alok Nandi, Xavier Marichal, "Compelling experiences in mixed reality interactive storytelling," *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, 2004.
- [4] C. -H. Cheung, L. -M. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *IEEE Trans. on Multimedia*, Volume 7, Issue 1, pp. 16-22, Feb. 2005
- [5] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, NO. 5, May 2002.
- [6] A. Criminisi, P. Perez and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting," *IEEE Transactions Image Processing*, 13, 2004, pp. 1200-1212.
- [7] K. Hariharakrishnan and D. Schonfeld, "Fast Object Tracking Using Adaptive Block Matching," *IEEE Transactions on Multimedia*, Vol. 7, No. 5, October 2005.
- [8] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, and D. Salesin, "Image Analogies," In *Proceedings of ACM SIGGRAPH 2002*, 341-346.
- [9] J. -F. Lalonde, D. Hoeim, A. A. Efros, C. Rother, J. Winn and A. Criminisi, "Photo Clip Art," *ACM Transactions on Graphics (SIGGRAPH 2007)*, August 2007, Vol 26. No. 3.
- [10] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video Inpainting Under Constrained Camera Motion," *IEEE Transactions on Image Processing*, Feb 2007.
- [11] P. Perez, M. Gangnet and A. Blake, "Poisson image editing," *ACM Transactions on Graphics (SIGGRAPH 2003)*, August 2007, Vol 2. Issue 3. pp. 313 – 318
- [12] T. K. Shih, N. C. Tang, J. C. Tsai and H. -Y Zhong, "Video Falsifying by Motion Interpolation and Inpainting," in the 2008 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, Alaska, June 24 – June 26, 2008.
- [13] T. K. Shih, N. C. Tang and J. -N. Hwang, "Exemplar-based Video Inpainting without Ghost Shadow Artifacts by Maintaining Temporal Continuity," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, Issue 3, March 2009 pp.:347 – 360.
- [14] A. Schödl, R. Szeliski, D. H. Salesin and Irfan Essa, "Video Textures," In *Proceedings of SIGGRAPH 2000*, 489–498, 2000.
- [15] J. Jia, J. Sun, C. Tang and H. Shum, "Drag-and-Drop Pasting." *ACM Trans. Graph. (TOG)* 25(3):631-637, 2006.
- [16] V. Kwatra, A. o Schödl, I. Essa, G. Turk and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts." *ACM Trans. Graph. (TOG)* 22(3):277-286 (2003).
- [17] A. Rav-Acha, Y. Pritch, D. Lischinski and S. Peleg, "Dynamosaics: video mosaics with non-chronological time," in *proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005. (CVPR 2005), San Diego, CA, United states, June 20-25,2005.










<p>N1: Motion inpainting result with a larger effect of waterfall</p>	
<p>N2: A source example showing fire in volcano</p>	
<p>N2: Motion inpainting is used to make the volcano spectacular</p>	
<p>N3: Another waterfall sources in a smaller scale with details</p>	
<p>N3: A new waterfall is added in the red box</p>	
<p>N4: A static background</p>	
<p>N4: A person is walking in a user-defined path</p>	

Figure 6. Narrative Generation (see demo website at <http://member.mine.tku.edu.tw/www/workshop10/index.htm>)