

Efficient Tracking of Regular Patterns on Non-rigid Geometry

Igor Guskov
University of Michigan
Ann Arbor, MI 48109
guskov@eecs.umich.edu

Abstract

We introduce a real-time robust tracking procedure for a regular pattern marked on a flexible moving surface such as cloth. Our system is capable of maintaining the tracked grid structure for long periods of time without quality deterioration, and requires minimal user interaction. It has been tested on videos of an actor dressed in a specially marked T-shirt and behaves favorably with the presence of self-occlusions, self-shadowing and folding of the cloth. The focus of this paper is on single camera video sequence processing, even though 3D shape reconstruction with multiple cameras is the motivating goal.

1. Introduction

Reliable acquisition of geometric data for non-rigidly moving surfaces is important for applications in character animation and ergonomic design. Examples of such non-trivially moving surfaces include face, hand, and clothed body motions. Often, the visible surface dynamics is the result of a complex physical interaction, and the collection of precise data for such surface motions is important for generation and validation of simplified motion models useful for animation and ergonomics applications.

Our goal is to build a general purpose geometry acquisition system able to acquire complex motion of non-rigid surfaces. Such system will employ several cameras tracking the motion of known regular patterns marked on the surfaces of interest. In this paper we focus on processing a single camera videofeed. The spatial coherence of the pattern allows us to reliably track the surface motion through long periods of time even when self-occlusions happens. This work is the extension of [7] with great improvement in tracking performance due mainly to incorporation of fast image alignment techniques of [1].

2. Related work

Motion capture data is widely used in computer graphics for animating articulated figures [4], as well as facial animation [13][6][5][3]. The setups used vary from video

tracking useful for the casual user to special marker-based motion capture products directed towards professional usage. A good review of computer vision techniques for capturing human motion is presented in [8]; many of these approaches have different objectives, using passively acquired video streams for surveillance and analysis. Being less invasive these techniques do not usually provide the degree of precision needed for production of high quality facial and cloth animation; although methods based on using low-rank constraints for surface reconstruction show great promise [11].

The capture of quality data for a wide range of flexible motions requires reliable handling of the effects of folding, self-occlusion and self-shadowing. We therefore choose to follow in the steps of [6] and introduce specially marked pattern on the tracked surface. The important feature of our method is that we track a coherent set of area primitives rather than individual point primitives as is customary in motion capture, or curve primitives of [2][12]. Area primitives have an advantage of establishing the connectivity of surface “grid” which can be very useful for further surface reconstruction.

Our method is also close to the “active blobs” work of [9] in its objective and reliance on pattern matching for non-rigid tracking. However, instead of considering one consistent textured mesh we treat our collection of tracked area primitives as a loose set of individuals. This may allow robust tracking of visible areas even in the situations where the majority of primitives is occluded.

3. Our approach

Our current system tracks the motion of a checkerboard pattern marked on a flexible surface. The black squares (*quads*) of such pattern are the individual entities tracked with our algorithm. For every frame of a video sequence, the parameters of each tracked quad are predicted based on the information from the previous few frames. Each active quad is then individually aligned with its counterpart in the image and assigned a confidence value. The quads with low confidence values are inactivated. The further processing

uses spatial coherence of the pattern to discard improbable quad arrangements and to revive the inactive parts of the tracked grid.

Notation We work in the image coordinates, and all the positions are pairs of real numbers. The image in a frame is denoted as $I(\mathbf{x})$ where $\mathbf{x} = (x, y)$ is the position in the image plane. The time dependence of variables is indicated with the integer parameter t (index of the frame).

The system tracks black squares as individual entities, and a homography h is associated with each black square ω_h so that ω_h matches the image of the unit square under the homography h : $\omega_h = W(\blacksquare; h)$, where $\blacksquare := [0, 1] \times [0, 1]$. The homography is represented with eight parameters [10]:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_0 & h_1 & h_2 \\ h_3 & h_4 & h_5 \\ h_6 & h_7 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix},$$

so that the transform $W(\cdot; h)$ is given as

$$W(x, y; h) = \left(\frac{h_0x + h_1y + h_2}{h_6x + h_7y + 1}, \frac{h_3x + h_4y + h_5}{h_6x + h_7y + 1} \right).$$

Pseudocode Below is our algorithm's structure:

```

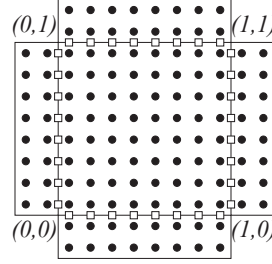
initialize
for every frame {
  for each active quad {
    predict homography in time
    match with the image
    if confidence is low -- deactivate
  }
  extend active region
  check grid's spatial consistency
}

```

There are two essential parts of the tracking cycle: first, all the active quads are tracked individually and then their spatial coherence is maintained via changes in active region. The following two sections give details on these two procedures.

4. Individual quad tracking

In order to track individual quads we first get the estimate of the current homography via temporal prediction, and then improve local match of the correspondingly distorted template with the image via a local optimization procedure. Then a separate calculation is performed to obtain a confidence measure that indicates how good is the local match between the distorted template and the corresponding image area.



An example of template sampling used in equations (1) and (2).

Temporal prediction Before the image alignment is run, we produce the initial estimate of every active quad's homography in the current frame from the previous two frames via $h(t) = 2h(t-1) - h(t-2)$ if it was active during both frames $t-1$ and $t-2$; otherwise the previous frame homography is copied $h(t) = h(t-1)$.

Fast template matching

We adapt the fast image alignment algorithm of [1] for the purpose of tracking individual quads. The iterative procedure is invoked that gradually improves local match between the template and the image. More precisely, the inverse compositional image alignment algorithm of [1] minimizes the error function

$$\int_{\Omega} [T(W(\mathbf{x}; \Delta h)) - I(W(\mathbf{x}; h))]^2 d\mathbf{x}$$

with respect to Δh . Here the region of integration is the cross-like area $\Omega := [-0.25, 1.25] \times [0, 1] \cup [0, 1] \times [-0.25, 1.25]$, and the template function T is defined on Ω as

$$T(\mathbf{x}) := \begin{cases} 0, & \text{if } \mathbf{x} \in \blacksquare \\ 1, & \text{otherwise.} \end{cases}$$

We discretize the integral and, using the fact that ∇T is zero away from the boundary of the unit square, we obtain the desired offset Δh as follows:

$$\Delta h = - \sum_{\mathbf{x} \in \beta_K} H^{-1} \left[\nabla T \frac{\partial W}{\partial h} \right] [T(\mathbf{x}) - I(W(\mathbf{x}; h))]. \quad (1)$$

Here the summation is over the sampled boundary of the unit square β_K (the figure on the left shows β_K as a set of small white squares). The Hessian matrix H is defined as

$$H = \sum_{\mathbf{x} \in \beta_K} \left[\nabla T \frac{\partial W}{\partial h} \right]^T \left[\nabla T \frac{\partial W}{\partial h} \right].$$

The Hessian matrix is constant and precomputed at the initialization stage of the algorithm. The resulting Δh is used to update current homography in the iterative procedure.

Confidence measure In [7], the L_0 -correlation was introduced to assign confidence level to a single tracked quad, that estimates the quality of the match of the template to the corresponding image region. In the current implementation that measure is replaced by a sampled version for the reasons of efficiency.

We define the confidence level as an L_0 norm correlation between the template function and the difference between the image and average image intensity within the template region:

$$c(h) := \frac{1}{|\Omega_K|} \sum_{\mathbf{x} \in \Omega_K} \text{sign}(I(W(\mathbf{x}; h)) - \bar{I}_h)(2T(\mathbf{x}) - 1). \quad (2)$$

Here Ω_K is the discretized version of the ‘‘cross’’ area (shown as small black circles in the figure above), and $|\Omega_K|$ is the number of samples in Ω_K .

$$\bar{I}_h := \frac{1}{|\Omega_K|} \sum_{\mathbf{x} \in \Omega_K} I(W(\mathbf{x}; h))$$

is the average image intensity within the template region. The defined confidence measure will increase when the relative distribution of image pixel values within the cross region $W(\Omega, h)$ is close to the template - most image pixels within the quad $W(\square, h)$ fall below the average intensity and the pixels in the ‘‘lapels’’ are above the average intensity. The confidence level $c(h)$ is guaranteed to be between -1 and 1 and the values close to 1 indicate a good match. Our experiments with other correlation functions show that the above L_0 correlation gives the most consistent results.

5. Managing the active region

There are two places during a tracking cycle where a quad can become inactive: first, if after a temporal prediction the local template matching fails to improve the confidence level above $c_{low} = 0.65$ then the quad becomes inactive; another routine that deactivates quads is the spatial consistency check. The spatial prediction, is on the other hand, the only procedure that can activate quads. In order to assure that the active region does not grow too fast we damp the confidence level of the quads just introduced by spatial extension by a factor of 0.9. Since only the quads with high confidence can predict their neighbors that restricts the growth of active region only to high confidence areas.

Checking spatial grid consistency The grid consistency check is needed to ensure that the tracked configuration corresponds to a valid grid structure. Every active quad is checked against its active neighbors whose current confidence level is above $c_{high} = 0.75$: denote the number of such confident quad-neighbors as N_c . A neighboring quad is consistent with the given quad if their common vertex position lies within a preset threshold. We are using a distance threshold equal to 40% of the maximal quad size in the current implementation. We then count the number N_i of inconsistent pairs for a given quad: note that N_i is always less or equal than N_c . If N_i is greater or equal than an empirically found value $N_i^{max}(N_c)$ that is given in Table 1 then the current quad is deactivated.

N_c	0	1	2	3	4
$N_i^{max}(N_c)$	0	1	1	2	2

Table 1. Spatial grid consistency threshold.

Spatial prediction for active region extension The currently inactive quads placement is predicted from its active neighbor by the translation that matches their common corner. Only active neighbors with confidence level above $c_{pred} = 0.7$ are used for such extension. After initial extension, the homography parameters are optimized via template matching described in the preceding section. If the match or spatial consistency criteria described above are not satisfied the quad is not activated.

Note This paper works with a single camera feed. For a pattern covering a general surface undergoing general motion there will always be parts of the surface not visible from a particular viewpoint. This problem will be addressed by combining streams of partial geometry (with confidence values for active quads) coming from several cameras.

6. Results

We have tested our algorithm on two video sequences acquired with B/W CCD camera at 30 frames/sec. The image resolution was 376 by 240 and the length of both video sequences was 20 seconds. An 8 x 8 checkerboard grid pattern with 32 black squares was painted with black markers on a white t-shirt. A person wearing the marked t-shirt performed various motions before the camera. The acquired video was then processed by our tracking system.

User interaction consisted of roughly delineating one square in the middle of the pattern (four mouse clicks). After this initialization, the grid structure was automatically extended to the entire connected visible region of the pattern. Consequently the grid pattern structure was tracked through the video. The tracker has successfully restored the grid structure after self-occlusions occurred. Figure 1 shows several tracked frames from the video sequence around occlusion time. There was no deterioration of the tracking quality through time.

The current procedure takes about 20 milliseconds per frame even for complicated frames with significant occlusions and self-shadowing. The system was tested on a PC workstation with 1GHz Pentium processor.

7. Conclusions and future work

We have considered direct tracking of marked grid patterns on flexible moving surfaces, and shown that we can track such grids at video frame rates using simple vision techniques. More future work is required with multiple cameras and full 3D surface descriptions.

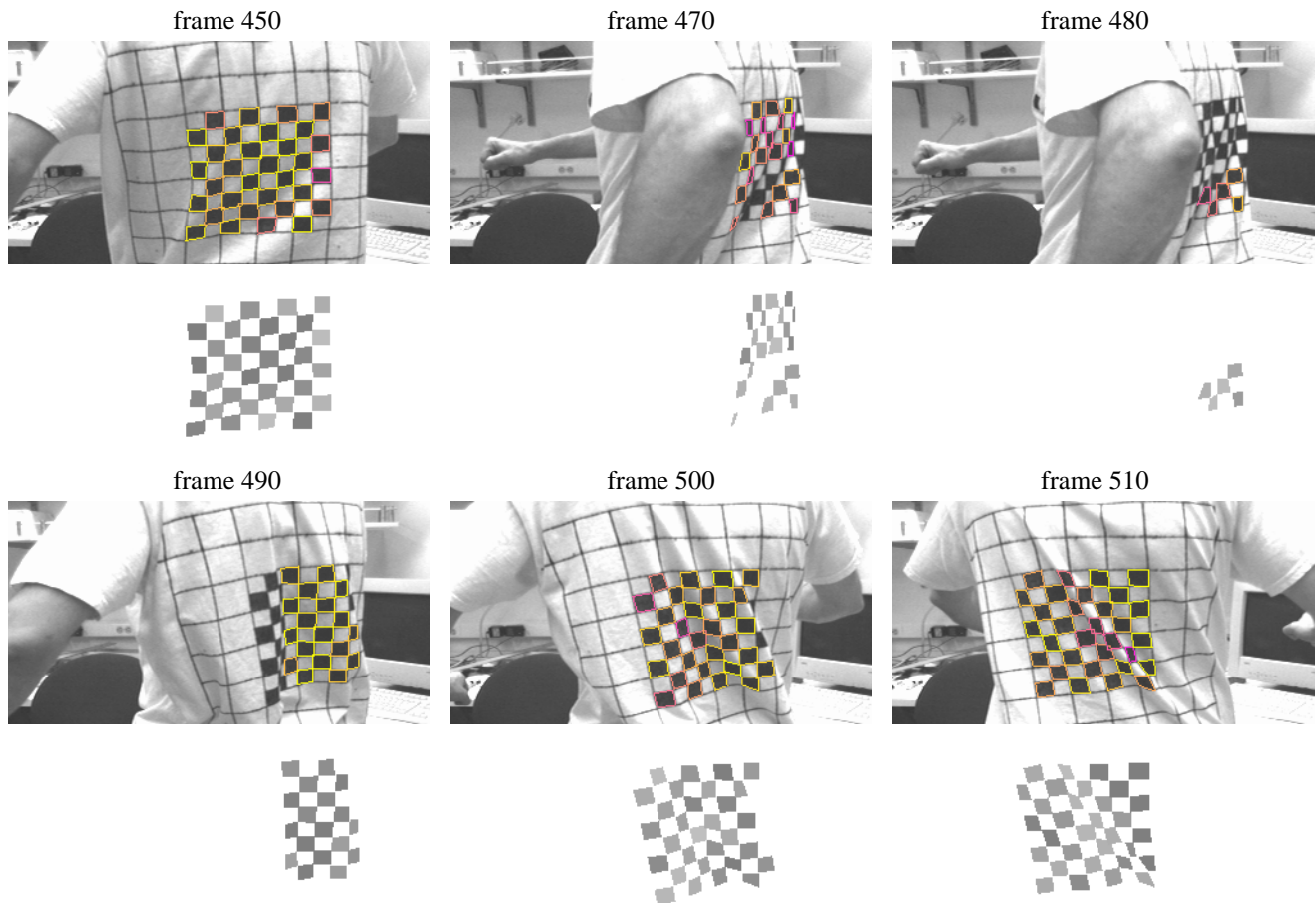


Figure 1. Six frames from the tracking sequence together with extracted grid of quads. Note that there are only five active quads in the frame 480. The structure of the grid is being correctly recovered in the following frames. Darker squares correspond to higher confidence values.

In this work we have not addressed the problem of acquiring fast abrupt surface motion that can result in tracker's confusion due to the invariance of the pattern to certain local translations. This is a fundamental problem that can be somewhat alleviated by careful design of color-coded patterns, and it will be addressed in our future work.

Acknowledgments This work was supported in part by NSF (CCR-0133554). The author would like to thank Leonid Zhukov, Pietro Perona and Xiaolin Feng for their help with acquiring the data, and Peter Schröder for encouragement and support.

References

- [1] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *Proc. of the CVPR*, 2001.
- [2] A. Blake and M. Isard. *Active Contours*. Springer, 1998.
- [3] I. Buck, A. Finkelstein, C. Jacobs, A. Klein, D. Salesin, J. Seims, R. Szeliski, and K. Toyama. Performance-driven hand-drawn animation. In *NPAR 2000*, Annecy, France, June 2000.
- [4] G. Cameron. Motion capture and CG character animation (panel). In *Proceedings of SIGGRAPH 1997*, 1997.
- [5] D. DeCarlo and D. Metaxas. Deformable model-based shape and motion analysis for images using motion residual error. In *Proceedings ICCV'98*, pages 113–119, 1998.
- [6] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *SIGGRAPH'98*, pages 55–66, 1998.
- [7] I. Guskov and L. Zhukov. Direct pattern tracking on flexible geometry. In *Proceedings of WSCG 2002*, 2002.
- [8] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81:231–268, 2001.
- [9] S. Sclaroff and J. Isidoro. Active blobs. In *Proceedings of ICCV 98*, pages 1146–1153, 1998.
- [10] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. *Computer Graphics*, 31:251–258, 1997.
- [11] L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and modelling non-rigid objects with rank constraints. In *Proc. IEEE CVPR 2001*, 2001.
- [12] L. Tsap, D. Goldgolf, and S. Sarkar. Nonrigid motion analysis based on dynamic refinement of finite element models. *IEEE Trans. on PAMI*, 22(5):526–543, May 2000.
- [13] L. Williams. Performance-driven facial animation. In *Proceedings of SIGGRAPH'90*, volume 24, pages 235–242, 1990.