Multiscale Inverse Compositional Alignment for Subdivision Surface Maps

Igor Guskov

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

Abstract. We propose an efficient alignment method for textured Doo-Sabin subdivision surface templates. A variation of the inverse compositional image alignment is derived by introducing smooth adjustments in the parametric space of the surface and relating them to the control point increments. The convergence properties of the proposed method are improved by a coarse-to-fine multiscale matching. The method is applied to real-time tracking of specially marked surfaces from a single camera view.

1 Introduction

Real-time tracking of textured surfaces in video is important for applications in user tracking and shape acquisition. In both of these applications active appearance models (AAMs) have been employed in a wide variety. A template matching procedure is often used as a basic component of AAMs. In order to accommodate the geometry of the tracked object the template is formulated as a textured shape. Both polygonal meshes and smooth splines were used as the underlying surface representations [1][2]. In this paper we develop a template tracking method for subdivision surfaces. Subdivision surfaces [3] offer a smooth and general representation of shape widely used in graphics and animation.

Inverse compositional template matching was recently proposed for the active appearance models based on triangular meshes [1][4]: it separately applies the current and incremental warps to the image and the template; the result is an efficient update procedure. For surface templates the separation of the incremental and current warps means that the incremental warp is performed in the parametric space of the surface. Thus, an additional difficulty arises on how to construct a space of such parametric warps. We propose a systematic approach to the construction of smooth atomic warps in the parameter space. Our derivations are first done for the ideal case of smooth warps, and then approximated in the space of subdivision surfaces.

The real-time operation of the template alignment procedure requires an efficient implementation. The natural multiresolution representation of subdivision surfaces serves as a suitable framework for implementing a coarse-to-fine matching algorithm that improves the convergence properties of our method. The multiscale approach also overcomes the deficiencies of the single resolution template matching for discontinuous textures.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3021, pp. 133–145, 2004.

Related work. Our work builds on the strengths of inverse compositional image alignment method developed by Baker and Matthews [4]. We extend their framework to handle templates specified on subdivision surfaces and propose a principled way of designing a set of smooth parametric adjustments (Section 3.3). The power of subdivision surfaces allow to represent big smooth patches of an arbitrary surface and control them with few control points. In this setting it is natural to consider coarse-to-fine version of the matching algorithm (Section 6). This improves the convergence properties of the matching without resorting to eigen-tracking approaches. Thus, it places less restrictions on the space of allowable surface deformations and does not require an apriori knowledge of the surface deformation model. The multiscale approach has been applied successfully for many related problems such as optical flow[5][6] and template search[7].

The goal of our work is similar to the Active Blobs effort [8]. While our method can handle general textured surfaces, it achieves its best tracking performance on specially quad-textured surfaces due to better control over conditioning of the involved matrix computations throughout all the levels of the hierarchy. In this paper, we only implement a simple appearance variation model and do not handle the detection of occluded regions and outlier pixels. Rather, the focus is on developing an efficient inverse compositional alignment procedure for general smooth surfaces described as subdivision models.

The paper is organized as follows: Section 2 introduces compositional template matching, Section 3 talks about surface and warp representations used in our work, Section 4 contains the detailed description of the proposed method. Sections 5 and 6 cover the partial template matching and the multiscale approach. Section 7 discusses the obtained results.

2 Compositional Template Matching

2.1 Forward Compositional Methods for Surfaces

The goal of template matching is to find the best warp of a template to match a given image. In the scenario of surface tracking, the template is better treated as a function on the parametric space of the tracked surface. Formally, let Ξ be the parametric space of the surface. Denote the projection of the surface in the image as $S(\xi)$, so that $S : \Xi \to \mathbf{R}^2$; we shall call the function S a *surface map*. The template function T represents surface color, so that $T : \Xi \to \mathcal{C}$, where $\mathcal{C} = \mathbf{R}$ for grayscale images and $\mathcal{C} = \mathbf{R}^3$ for color images.

Given an image $I : \mathbf{R}^2 \to \mathcal{C}$, the matching problem consists of finding the surface map S which minimizes the error functional

$$E(S) := \|I \circ S - T\|^2 = \int_{\Xi} (I(S(\xi)) - T(\xi))^2 d\xi.$$

At this point we assume that all of the surface is visible, the case of partial visibility is considered in Section 5. We would also like to delay specifying a particular representation for the surface map S and treat it as a general smooth function in this section.

The two approaches to the template matching problem are the *additive* and *compositional* methods. Additive approach performs update $S \leftarrow S + dS$, and finds the optimal surface adjustment dS by solving $\min_{dS} \|I \circ (S + dS) - T\|^2$, The compositional approach updates the surface map via $S \leftarrow S \circ W$. It looks for the optimal warping in the parametric space: $\min_W \|I \circ S \circ W - T\|^2$, or in more detail:

$$\min_W \int_{\Xi} (I(S(W(\xi))) - T(\xi))^2 d\xi. \tag{1}$$

The two approaches can be shown to be equivalent when the incremental warp $W : \Xi \to \Xi$ is close to the identity map so that $W(\xi) \approx \xi + dW(\xi)$ and dW is small. Then the corresponding surface adjustment will be close to

$$S \circ W - S \approx \frac{\partial S}{\partial \xi} dW. \tag{2}$$

Details of the proof can be found in [4]. When the Jacobian $\partial S/\partial \xi$ is not full rank the two approaches are not equivalent. This happens, for instance, in the silhouette region of the surface map where the 2D tangential space gets projected onto a 1D line in the image.

The compositional approach is possible in its pure form when one can find a set of planar warps which form a group. When representing a general evolving surface, one requires more flexibility than present in the classical groups of transformations such as translations, affine transforms or homographies. On the other hand, the very general group formed by composition of arbitrary smooth maps used above is not practical. Our approach will be therefore to derive compositional methods for the general smooth case, and then approximate the needed computations in a smooth basis.

2.2 Inverse Compositional Method

We shall now derive the *inverse* compositional method [4] in the general case of smooth surface maps and warps. The basic assumption for equivalence between the forward and inverse compositional methods is the closeness of the incremental warp map to being the identity map. In particular, it is assumed that $\det(\partial W/\partial \xi) \approx 1$. Then, the change of variable $\xi = W^{-1}(\eta)$ in (1) leads to the following minimization problem:

$$\min_W \int_{\Xi} (I(S(\eta)) - T(W^{-1}(\eta)))^2 d\eta.$$

The inverse of the incremental warp $V := W^{-1}$ can be sought directly:

$$\min_V \int_{\Xi} (I(S(\eta)) - T(V(\eta)))^2 d\eta.$$

This approach results in less per frame computation than in the corresponding forwards methods as was shown in [4]. Once the incremental warp is found, we can update the surface map via $S \leftarrow S \circ V^{-1}$.

3 Template and Warp Representation

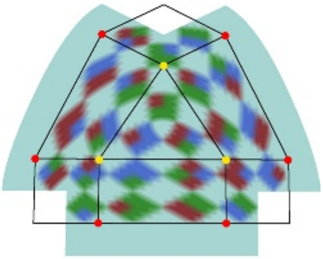
The preceding section introduced the inverse compositional method in a general form. In practice, we need to use a specific representation for both the subdivision surface S and the parametric warp W. This section describes surface and warp representations used in our approach.

3.1 Subdivision Surface Maps

We define our template surface maps using Doo-Sabin subdivision scheme [9]. In this section, we give short description of our implementation. For a detailed introduction to subdivision surface modeling the reader is referred to [10] [3].

A subdivision surface is controlled by a control polygonal mesh $\mathcal{M}$ that has a set of control vertices $\mathcal{V}$. For each control vertex $k \in \mathcal{V}$, a planar position $p_k \in \mathbf{R}^2$ is specified. The Doo-Sabin subdivision scheme is a *dual* scheme, so that its (dual) control vertices correspond to faces of a primal polygonal mesh $\mathcal{M}'$. We restrict the primal mesh to be a manifold quad mesh possibly with boundary.

The parametric space Ξ of a subdivision surface map is formed as the union of square patches Ξ_k $(k \in \mathcal{V})$ glued along the shared edges. Each patch is a square $[0,1] \times [0,1]$, and is associated to a particular primal face from $\mathcal{Q}'$ or the corresponding dual control vertex from $\mathcal{V}$. A point ξ in a parametric space is then fully described by the (dual) control vertex index k and a position in $[0,1] \times [0,1]$. Given a function $f : \Xi \to \mathbf{R}^d$, we shall use notation $\partial f / \partial \xi^i, i = 1,2$ for its derivatives. This is well defined within patches away from the patch boundaries, which is sufficient for the purposes of this paper.



We use primal-dual approach described in [11] to implement subdivision. The corner vertices are dependent on the boundary and inside vertices that share the same control face of the mesh; we also exclude the corner patches from the parametric region of the template. Thus, the corner control vertices only appear as an auxiliary dependent quantity; to simplify notation we redefine the set of control vertices $\mathcal{V}$ as the union of inside and boundary vertices in the remainder of this paper, with the parametric region Ξ defined correspondingly to exclude the corner patches.

Having defined parametric region Ξ and the set of control vertices $\mathcal{V}$, we define the C^1 subdivision surface map at any parametric position ξ via

$$S[p](\xi) := p_k \phi^k(\xi),$$

where the summation in k is assumed over every index in $\mathcal{V}$, and p_k's are two-dimensional points.

The following properties of $\phi^k(\xi)$ are important:

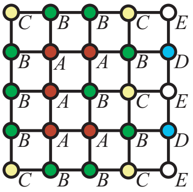
- $\sum_{k \in \mathcal{V}} p_k \phi^k(\xi) = 1$ for $\xi \in \Xi$, so that $\phi^k, k \in \mathcal{V}$ form the partition of unity.

- Each $\phi^k(\xi)$ has local support in Ξ that consists of the patch of k together with all the patches that share at least one patch corner with it. Thus, each parametric point ξ is only affected by few control points whose support includes ξ.

3.2 Control Mesh Extension

In this section we describe a procedure for extrapolating the control mesh of the subdivision map from a given subset of (dual) control vertices with known positions to a wider set of control vertices adjacent to this known region. We shall use this procedure in two situations: creation of a canonical control position arrangement for the atomic warps as described in Section 3.3, and extension of active region positions for partial template matching as described in Section 5.

Let $\mathcal{V}_A$ be the set of *active* control vertices, and define the set $\mathcal{V}_B$ of non-active control vertices immediately adjacent to the set $\mathcal{V}_A$. Suppose that the positions of control vertices in both sets $\mathcal{V}_A$ and $\mathcal{V}_B$ are known. We introduce three sets of control vertices, depending on their relative adjacency to the known vertex set, and define a procedure for extending positions in $\mathcal{V}_A \cup \mathcal{V}_B$ to these three sets. The figure on the right shows an example of vertex set assignment.



- $\mathcal{V}_C$ is the set of control vertices (not in $\mathcal{V}_A$ or $\mathcal{V}_B$) whose primal faces share at least one primal control vertex with the primal control faces corresponding to control vertices in $\mathcal{V}_A$. In other words, a control vertex from $\mathcal{V}_C$ will belong to at least one dual control face whose vertices include at least one vertex from $\mathcal{V}_A$. Note that the same dual face will also include at least two vertices from $\mathcal{V}_B$ (this will be important for the extrapolation method described below).
- $\mathcal{V}_D := \mathsf{bou}(\mathcal{V}_A \cup \mathcal{V}_B) \setminus (\mathcal{V}_A \cup \mathcal{V}_B \cup \mathcal{V}_C)$. [1] Each vertex in $\mathcal{V}_D$ has at least one adjacent vertex $v_b \in \mathcal{V}_B$. By construction, the vertex v_o on the opposite side of v_b will also be from $\mathcal{V}_A \cup \mathcal{V}_B$.
- $\mathcal{V}_E$ is the set of control vertices (not in $\mathcal{V}_A \cup \mathcal{V}_B \cup \mathcal{V}_C \cup \mathcal{V}_D$) whose primal faces share at least one primal control vertex with the primal control faces corresponding to control vertices in $\mathcal{V}_B$.

For a vertex in $\mathcal{V}_C$ take a dual face that has at least three known vertices. Let this dual control face have n vertices, and index its corners with integers $i = 0, \ldots, n-1$ in a counterclockwise order. We associate the vertex i with the parameter value $\alpha_i = 2\pi i/n$, and find the ellipse $p(\alpha) = C + D_1 \cos\alpha + D_2 \sin\alpha$ that best fit the known points in the least square sense. We then assign all the unknown vertex positions on the ellipse at the appropriate α locations. In the regular case of three known vertices and a single unknown vertex, we obtain a parallelogram rule. If there are several prediction for a vertex in $\mathcal{V}_C$ we compute their average.

[1] For a subset of vertices $\mathcal{U} \subset \mathcal{V}$ we define its boundary $\mathsf{bou}(\mathcal{U})$ as the set of vertices from $\mathcal{V} \setminus \mathcal{U}$ adjacent to at least one vertex in $\mathcal{U}$.

Now for each vertex in $\mathcal{V}_D$ we extrapolate its value linearly via $p(v_d) = 2p(v_b) - p(v_o)$ where v_b and v_o are as described above. After this step, all the vertices in $\mathcal{V}_A \cup \mathcal{V}_B \cup \mathcal{V}_C \cup \mathcal{V}_D$ are assigned a position. We can now replicate the extrapolation step of $\mathcal{V}_C$ for all the vertices in $\mathcal{V}_E$, which concludes our extrapolation procedure.

The described procedure guarantees the position assignment for all the control vertices that affect the subdivision map values within parametric patches associated with $\mathcal{V}_A \cup \mathcal{V}_B$. This will be especially important for the partial template matching of Section 5.

3.3 Warp Space

For a surface map represented as $S[p](\xi) = \phi^k(\xi)p_k$, the control point adjustments Δp result in the surface sample positions changed by $\phi^k(\xi)\Delta p_k$. For the compositional approach to work, this surface adjustment has to match the surface adjustment obtained via warping. Any smooth parameterized warp map $W(\xi; q)$ such that $W(\xi; 0) = \xi$ can be written:

$$W(\xi; q) = \xi + \frac{\partial W(\xi; 0)}{\partial q} q + O(q^2).$$

Define $\gamma^{ki}_{i'}(\xi) := \partial W^i(\xi; 0)/\partial q^{i'}_k$. For the purpose of template matching, these first derivatives are all that is necessary to define explicitly.

For a fixed index K, we choose γ^K in such a way that for some configuration of control points $\bar{p}^K$ the surface map update corresponding to the parametric control point q_K matches the surface update coming from the control point adjustment Δp_K. More precisely, we would like that for all $\xi \in \Xi$:

$$\phi^k(\xi)\Delta p^i_k = \frac{\partial S^i[\bar{p}^K](\xi)}{\partial \xi^j} \gamma^{kj}_{i'}(\xi) q^{i'}_k,$$

when $\Delta p^i_k = \delta^K_k Z^i$ and $q^{i'}_k = \delta^K_k Z^{i'}$ (here Z is an arbitrary two-dimensional vector). It follows that (no summation on the capital index K):

$$\phi^K(\xi)\delta^i_{i'} = \frac{\partial S^i[\bar{p}^K](\xi)}{\partial \xi^j} \gamma^{Kj}_{i'}(\xi).$$

The quantity on the left hand side is a scaled identity matrix, therefore we can conclude that $\gamma^{Kj}_{i'}(\xi)$ is the inverse of the 2×2 matrix $\partial S[\bar{p}^K](\xi)/\partial \xi$ times the scalar value $\phi^K(\xi)$ at each ξ. Hence we set

$$\gamma^K(\xi) := \left(\frac{\partial S[\bar{p}^K](\xi)}{\partial \xi} \right)^{-1} \phi^K(\xi). \tag{3}$$

Each $\gamma^{Kj}_{i'}(\xi)$ defined above is a local function, non-zero on the support of the basis function $\phi^k(\xi)$.

We now need to define $\bar{p}^K$ (for this canonic arrangement of control points around a vertex K the warped surface adjustment will have an exact representation in the basis of subdivision shapes). Note that only evaluation of $S[\bar{p}^K]$ within the support of a single basis function $\phi^k(\xi)$ is required. We use the control mesh extension process from Section 3.2 with the active set $\mathcal{V}_A$ consisting of a single vertex $K \in \mathcal{V}$. The four immediate neighbors of K form the set $\mathcal{V}_B$, and we assign the five control points values on the plane so that $\bar{p}^K_K$ is at the origin, and the points from $\mathcal{V}_B$ are positioned at $(1,0)$, $(0,1)$, $(-1,0)$, and $(0,-1)$. The extension procedure then defines all the other positions required to evaluate $S[\bar{p}^K]$ on the patches near K. In our experience, this results in non-degenerate assignments of $\bar{p}^K$.

4 Inverse Compositional Method for Subdivision Maps

4.1 Parametric Adjustment

We can now find the optimal parameters q of the parametric adjustment, by minimizing the fit functional with respect to q.

$$J(q) := \int_{\Xi} |I(S[p](\xi)) - T(W(\xi;q))|^2 \, d\xi$$

We introduce the pointwise error of the current fit as $E(\xi;p) := I(S[p](\xi)) - T(\xi)$, and obtain the following approximation to $J(q)$:

$$J(q) \approx \sum_{m=1}^{3} \int_{\Xi} \left[E_m(\xi;p) - \frac{\partial T_m}{\partial \xi^i}(\xi) \gamma^{ki}_{i'}(\xi) q^{i'}_k \right]^2 d\xi,$$

where the subscript $m = 1,2,3$ denotes the appropriate color channel.

Differentiating this expression with respect to $q^{i'}_k$ and introducing the notation $h^{ki'}_m(\xi) := \gamma^{ki}_{i'}(\xi)\partial T_m/\partial \xi^i(\xi)$ we get the following system of linear equations for the optimal parametric adjustment parameters q:

$$\int_{\Xi} h^{ki'}_m(\xi) h^{lj'}_m(\xi) d\xi \, q^{j'}_l = \int_{\Xi} h^{ki'}_m(\xi) E_m(\xi;p) d\xi, \quad k \in \mathcal{V}, i' = 1,2.$$

As is expected from an inverse compositional method, the matrix $A_{ki',lj'} = \int_{\Xi} h^{ki'}_m(\xi) h^{lj'}_m(\xi) d\xi$ on the left hand side does not depend on p and its inverse can be precomputed, while the right hand side $b_{ki'} = \int_{\Xi} h^{ki'}_m(\xi) E_m(\xi;p) d\xi$ depends on the current error of the fit, and has to be recomputed during optimization.

The linear system $Aq = b$ has $2|\mathcal{V}_I|$ unknowns, and in order to guarantee its proper solution we need to ensure that the surface template has enough edge features of different orientations (similar to [12]). In order to ensure more robust tracking we apply the multiscale template: in this case we need to ensure that the edge features exist at all the resolutions. Surfaces marked with quad patterns do provide such features as long as the surface patches controlled by a single vertex covers a few of the pattern quads (see Figure 1 for the comparison of condition numbers of the matrix $A = H^t H$).

4.2 Evaluation of Control Vertex Adjustments

Once the parametric adjustment is found, we compute the surface sample displacements that correspond to the approximate inverse of the parametric displacement, namely a surface sample $S[p](\xi)$ is moved to $S[p](W(\xi;-q))$ where q is the optimal parametric displacement parameters found in the previous section.

We find an approximation to the actual samples movement via

$$S[p](W(\xi;-q)) \approx S[p](\xi - q_k^{i'}\gamma_{i'}^{k\cdot}(\xi)) \approx S[p](\xi) - \frac{\partial S[p]}{\partial \xi^i}(\xi)\,\gamma_{i'}^{ki}(\xi)\,q_k^{i'},$$

so that the sample with parameter ξ undergoes the displacement

$$\sigma(\xi;q) := -\frac{\partial S[p]}{\partial \xi^i}(\xi)\,\gamma_{i'}^{ki}(\xi)\,q_k^{i'};$$

note that $\partial S[p]/\partial \xi$ depends on the current p which makes the update step nonlinear [1].

In order to find the approximation to the appropriate displacement of control vertices of the subdivision surface we sample the surface displacement on a fixed four-by-four grid of parameter samples within each quad patch, and solve for the corresponding control vertex displacements that are optimal in the least square sense.

Denote the discrete set of all the sampled surface displacement parameters as Ξ_S. We need to find Δp such that the following set of constraint is approximately satisfied, that is $S[\Delta p](\xi) = \sigma(\xi;q)$ for all $\xi \in \Xi_S$. Using the expression for the subdivision map we obtain $\Delta p_k \phi^k(\xi) = \sigma(\xi;q), \xi \in \Xi_S$. Introduce the matrix of basis function sample values $(\Phi_S)_{\xi v} := \phi^v(\xi), v \in \mathcal{V}_I, \xi \in \Xi_S$. The least-squares solution the above linear system then gives us the following expression for Δp:

$$\Delta p = (\Phi_S^t \Phi_S)^{-1} \Phi_S^t \sigma(q),$$

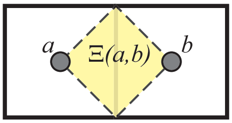
and the matrix inverse on the right-hand side can be precomputed during system initialization. Once the optimal control point adjustment is found, we update the control point positions using $p^{(n+1)} = p^{(n)} + \Delta p$ which concludes a single iteration of our template alignment procedure.

Appearance variation. A simple constant appearance variation model can be added as in [4] by changing the pointwise fit error to be $E(\xi;p) := (I(S[p](\xi)) - I_{average}) - T^*(\xi)$, where $I_{average}$ is an estimate of the average color value of image samples of the current surface, and $T^*(\xi) := T(\xi) - T_{average}$ is the original template adjusted so that its average is zero.

5 Partial Template Matching

When some part of the surface is occluded, it is no longer possible to track its motion, and the template matching should exclude the corresponding control vertices. At the same time, it would not be practical to create a completely separate surface template from scratch. Rather we would like to reuse a partially active template. In this section, we describe how the template matching algorithm described above can be modified when only a subset of control vertices are allowed to change independently.

We assume that a subset $\mathcal{V}_A$ of active vertices is given (one can think that the surface portion $S(\bigcup_{k\in\mathcal{V}_A} \Xi_k)$ is fully visible); and take $\mathcal{V}_B = \mathsf{bou}(\mathcal{V}_A)$. The union of $\mathcal{V}_A$ and $\mathcal{V}_B$ is therefore used as the set of independent control vertices found by template matching, we call it $\mathcal{V}_{AB} := \mathcal{V}_A \cup \mathcal{V}_B$. Thus, the surface is determined by vertices in $\mathcal{V}_{AB}$; we modify the set of warp parameters Δq in the same way, and all the linear systems solved in the algorithm will be of such reduced dimensions.



Another modification to the algorithm is the modification of the integration region Ξ. Denote by $\mathcal{E}_{AB}$ the set of all the edges with both ends in $\mathcal{V}_{AB}$. With each edge (a,b) we associate the union $\Xi(a,b)$ of two triangular sections of the corresponding patches Ξ_a and Ξ_b, as shown in the figure to the right. The integration region Ξ_{AB} used in the partial matching algorithm is defined to be $\Xi_{AB} = \bigcup_{(a,b)\in\mathcal{E}_{AB}} \Xi(a,b)$. Therefore, the surface map needs to be evaluated on samples within Ξ_{AB} at every step of matching. This is only possible when a wider set of control vertex positions is known; the extension procedure from Section 3.2 is used for extrapolating vertex positions from the set $\mathcal{V}_{AB}$ to the wider set $\mathcal{V}_{ext}$ required for the surface evaluation on Ξ_{AB}.

6 Tracking Quad Patterns with the Multiscale Matching

The derivation of the template alignment method of the previous section relied on the fact that we could take derivatives of the template function and apply basic first order approximations. We would like to apply this method to tracking colored quad patterns similar to the ones used in [13]. Those patterns considered as functions are not even continuous. In this section, we discuss the issues that arise from this complication and our approach to overcoming them. We start by analyzing a simple one-dimensional example, and then discuss the implications of this analysis for the original surface tracking case.

One-dimensional example. Assume that our template is the step function[2]: $T(x) = \chi(x)$, and consider the error functional $J(p) := \int |T(x+p) - I(x)|^2\, dx$. If the image function is a shifted step function, that is $I(x) = \chi(x+a)$ for some a, the error functional is not smooth: $J(p) = |p-a|$. It follows that the gradient based methods may not perform well on such an optimization problem. This can be noticed when we apply template matching on sharp images with discontinuities: the adjustment to the template parameters p coming from the gradient descent method will stop decreasing as it approaches the optimal value.

[2] Define $\chi(x) = 0$ for $x < 0$ and $\chi(x) = 1$ for $x \geq 0$

In practice we work with a discretized version of the template, so all the derivatives can be evaluated as divided differences. Consider the template at the grid step h:

$$T_h(x) = \begin{cases} 0, & x < -h/2 \\ 1/2 + x/h, & -h/2 \le x < h/2 \\ 1, & x \ge h/2 \end{cases}$$

The minimization of the functional $J_h(p) := \int |T_h(x+p) - I(x)|^2 \, dx$ leads to the following expression for the optimal translation of the template:

$$p_h^* = \int (I - T_h) T_h' dx \Big/ \int (T_h')^2 dx = \int_{-h/2}^{h/2} (I - T_h) dx.$$

Hence the upper bound on the parameter adjustment is proportional to the step h: $|p_h^*| \le h \|I - T_h\|_\infty \le 2h \max\{\|I\|_\infty, 1\}$. Thus, for large adjustments in p we need to employ coarse versions of the template, while small h are preferable for the precise positioning of the template. It therefore makes sense to proceed from coarse to fine discretizations. This is similar in spirit to multiscale optical flow and template matching algorithms [6] [7][14].

We apply the coarse-to-fine approach to our surface template matching algorithm. To illustrate its convergence properties at different resolutions we plot the mean-square error of a template fit with respect to the number of iterations for the *Quad sheet* model (see Figure 1). It is clear that the coarse template matching makes larger adjustments towards the minumum but the precision of the result is limited. At the same time a finer template matching is able to recover the minumum with high precision but requires more iterations, and is also more intensive computationally. The combined method shown in the plot

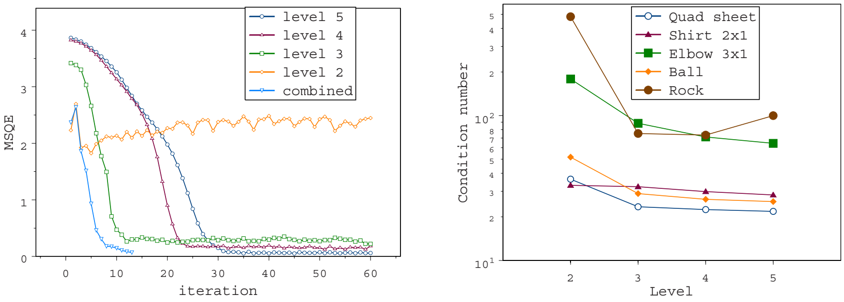


Fig. 1. Left: mean-square error during iterations of template matching procedures at different levels of template resolution. Right: comparison of condition numbers of $H^t H$ matrices at different levels of resolution. For quad patterns the condition number stays relatively low on all the levels, while for natural patterns it is less controlled.

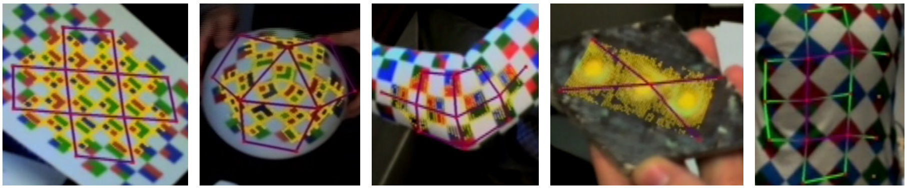


Fig. 2. Various surface templates used. See accompanying video for tracking examples

uses three iterations on each level starting from the coarsest, and achieves better convergence at lower cost. The moderate condition number of the matrix H^tH plotted in Figure 1 also contributes to the success of the multiscale method for quad-marked patterns.

In the case of a piecewise constant quad-marked pattern, the only template samples participating in template matching are located along the discontinuities. Thus, when the template subdivision level l is increased, the number of sampled points will only increase linearly with n rather than quadratically (as n^2 where $n = 2^l$) as is the case for a template with globally non-trivial gradient. This compression of the template improves the efficiency of the matching (see the left two templates in Figure 2, the yellow dots indicate the used template samples).

7 Results

We have implemented our template matching algorithm for Doo-Sabin subdivision surfaces within a real-time tracking application framework. After manual initialization, the template is tracked in video sequence. A linear temporal prediction scheme is employed to obtain the initial guess for the template positioning in every consecutive frame of video. The application was able to perform at 30 frames per second for all the full surface tracking examples presented below on a 2GHz Pentium laptop. The video was acquired with a digital camera at 640×480 resolution. We used the combined multiscale method that ran two iterations of matching on each level of resolution on levels two to four, and a single iteration on level five. The below table shows the number of inside and boundary control vertices in the models used for this paper. The accompanying video contains video sequences captured in real time.

Name	Number of CVs	Number of active inside CVs	Number of active boundary CVs	Texture type
Quad sheet	12	4	8	pure
Ball	9	3	6	pure
Shirt 2x1	8	2	6	acquired
Elbow 3x1	11	3	8	acquired
Rock	5	1	4	acquired
Partial shirt	21	5	10	pure

We have used both predefined *pure* quad template patterns and the textures *acquired* from the video frame during the initialization process. The pure patterns result in the sparse pattern for the participating samples along the quad edges; for the acquired textures we used thresholding on the $h_m^{li}(\xi)$ coefficients to determine which samples should be participating in the integral discretization.

For the partial tracking example, we used a five by five grid template for the t-shirt. The active region included five vertices as shown in Figure 2. This example runs at 15 frames per second.

8 Future Work

We presented a multiscale method for matching subdivision surface templates. The future work will need to address the maintenance of the active visible region for partial surface tracking as well as the automatic initialization procedure. The compositional methods work within the surface and cannot account for surface displacement near its silhouettes from a single view. A multi-view extension of the presented procedure can help alleviate this problem. A more complex appearance variation modeling is also left as a future work direction.

Acknowledgments. This work was partially supported by NSF(CCR-0133554) and University of Michigan AI Lab.

References

1. Matthews, I., Baker, S.: Active appearance models revisited. Technical Report CMU-RI-TR-03-02, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2003)
2. Cascia, M.L., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: An approach based on robust registration of texture-mapped 3d models. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) **22** (2000)
3. Zorin, D., Schröder, P., eds.: Subdivision for Modeling and Animation. Course Notes. ACM SIGGRAPH (1999)
4. Baker, S., Matthews, I.: Equivalence and efficiency of image alignment algorithms. In: Proc. of the CVPR. (2001)
5. Szeliski, R., Shum, H.Y.: Motion estimation with quadtree splines. IEEE Transactions on Pattern Analysis and Machine Intelligence **18** (1996) 1199–1210
6. Simoncelli, E.: Bayesian multi-scale differential optical flow. In: Handbook of Computer Vision and Applications. (1993) 128–129
7. Borgefors, G.: Hierarchical chamfer matching: A parametric edge matching algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence **10** (1988) 849–865
8. Sclaroff, S., Isidoro, J.: Active blobs. In: Proceedings of ICCV 98. (1998) 1146–1153
9. Doo, D., Sabin, M.: Behaviour of recursive division surfaces near extraordinary points. Computer-Aided Design **10** (1978) 356–360
10. Warren, J., Weimer, H.: Subdivision Methods For Geometric Design: A Constructive Approach. Morgan Kaufmann (2001)

11. Zorin, D., Schröder, P.: A unified framework for primal/dual quadrilateral subdivision schemes. CAGD **18** (2001) 429–454
12. Shi, J., Tomasi, C.: Good features to track. In: CVPR. (1994) 593–600
13. Guskov, I., Klibanov, S., Bryant, B.: Trackable surfaces. In: Proceedings of ACM/EG Symposium on Computer Animation. (2003) 251–257
14. Gleicher, M.: Projective registration with difference decomposition. In: Proc. of IEEE CVPR 1997. (1997) 331–337